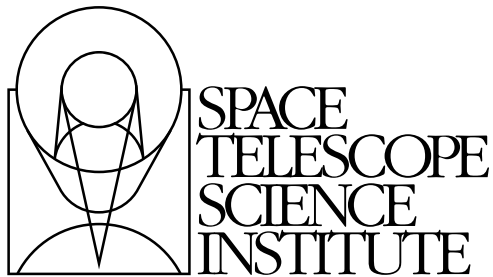

Version 3.0
July, 2004

HST Data Handbook for ACS



Space Telescope Science Institute
3700 San Martin Drive
Baltimore, Maryland 21218
help@stsci.edu

User Support

For prompt answers to any question, please contact the STScI Help Desk.

- **E-mail:** help@stsci.edu
- **Phone:** (410) 338-1082 or 1-800-544-8125

World Wide Web

Information and other resources are available on the ACS World Wide Web page:

- URL: <http://www.stsci.edu/hst/acs/>

ACS Revision History

Version	Date	Editor
3.0	July 2004	Cheryl Pavlovsky, Adam Riess, Jennifer Mack, and Ron Gilliland, Editors, ACS Data Handbook, Diane Karakla, Chief Editor and Susan Rose, Technical Editor, HST Data Handbook
2.0	November 2003	Jennifer Mack and Ron Gilliland
1.0	January 2002	Jennifer Mack, ACS Editor Bahram Mobasher, Chief Editor

Authorship

This document is written and maintained by the ACS+WFPC2 Branch in the Instruments Division of STScI which consists of the following individuals: Biretta, J., Boffi, F., Bohlin, R., Cox, C., Fruchter, A., Giavalisco, M., Gilliland, R., Gonzaga, S., Heyer, I., Hook, R., Joglee, S., Koekemoer, A., Mack, J., Mutchler, M., Pavlovsky, C., Platais, V., Riess, A., Sirianni, M., van der Marel, R., and van Orsow, D. Other scientists who contributed to the writing of this handbook include Clampin, M., de Marchi, G., Hack, W., Hartig, G., Krist, J., Sparks, W., and Welty, A. at STScI or formerly at STScI, Blakeslee, J. and Meurer, G., of the ACS Science Team, and Pasquali, A. and Walsh, J. at the ST-ECF.

In publications, refer to this document as:
Pavlovsky, C., et al. 2004, "ACS Data Handbook", Version 3.0, (Baltimore: STScI).

Send comments or corrections to:
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, Maryland 21218
E-mail:help@stsci.edu

Table of Contents

Preface	xi
----------------------	----

Part I: Introduction to Reducing the HST Data

Chapter 1: Getting HST Data	1-1
1.1 Archive Overview	1-2
1.1.1 Archive Registration.....	1-3
1.1.2 Archive Documentation and Help	1-3
1.2 Getting Data with StarView.....	1-4
1.2.1 Downloading and Setting Up StarView.....	1-4
1.2.2 Simple Use of StarView	1-4
1.2.3 Marking and Retrieving Data with StarView	1-9
1.2.4 Using StarView to Retrieve Calibration Files and Proposal Information.....	1-10
1.2.5 Advanced Features of StarView	1-11
1.2.6 StarView and the Visual Target Tuner.....	1-15
1.2.7 Quick Proprietary Data Retrieval with StarView	1-15
1.3 Getting Data with the World Wide Web	1-16
1.3.1 WFPC2 Associations	1-19
1.3.2 High Level Science Products.....	1-19
1.4 Reading HST Data Disks	1-21
Chapter 2: HST File Formats	2-1
2.1 FITS File Format	2-2
2.1.1 Working with FITS Image Extensions.....	2-4
2.1.2 Working with FITS Table Extensions.....	2-8

2.2 GEIS File Format.....	2-11
2.2.1 Converting FITS to GEIS.....	2-12
2.2.2 GEIS Data Groups.....	2-13
2.2.3 Working with GEIS Files.....	2-14
2.2.4 The “waiver” FITS Format.....	2-17
Chapter 3: Analyzing HST Data	3-1
3.1 Alternative Means of Accessing HST Data.....	3-2
3.1.1 Interactive Data Language (IDL).....	3-2
3.1.2 Python.....	3-2
3.1.3 Fortran and C.....	3-3
3.1.4 Java.....	3-3
3.1.5 PyRAF.....	3-3
3.2 Navigating STSDAS.....	3-4
3.2.1 STSDAS Structure.....	3-4
3.2.2 Packages of General Interest.....	3-4
3.3 Displaying HST Images.....	3-7
3.3.1 The Display Task.....	3-7
3.3.2 Working with Image Sections.....	3-11
3.4 Analyzing HST Images.....	3-11
3.4.1 Basic Astrometry.....	3-12
3.4.2 Examining and Manipulating Image Data.....	3-14
3.4.3 Working with STIS, ACS, and NICMOS Imsets.....	3-16
3.4.4 Photometry.....	3-18
3.4.5 Combining Dithered HST Datasets with MultiDrizzle.....	3-22
3.5 Displaying HST Spectra.....	3-23
3.5.1 STIS Spectra.....	3-23
3.5.2 FOS and GHRS Spectra.....	3-25
3.5.3 Producing Hardcopy.....	3-26
3.6 Analyzing HST Spectra.....	3-27
3.6.1 Preparing STIS Spectra for Analysis.....	3-28
3.6.2 Preparing FOS and GHRS Data.....	3-30
3.6.3 Photometry.....	3-33
3.6.4 General Tasks for Spectra.....	3-33
3.6.5 STSDAS Fitting Package.....	3-37
3.6.6 Specfit.....	3-40

3.7 References	3-40
3.7.1 Available from STScI	3-40
3.7.2 Available from NOAO	3-41
3.7.3 Other References Cited in This Chapter.....	3-41

Part II: ACS Data Handbook

ACS Introduction	3
<i>How to Use this Handbook</i>	3
Chapter 1: ACS Overview	1-1
1.1 Instrument Design and Capabilities	1-1
1.1.1 Detectors	1-2
1.1.2 ACS Optical Design	1-4
1.1.3 ACS Geometric Distortion.....	1-4
1.2 Basic Instrument Operations	1-7
1.2.1 Target Acquisitions	1-7
1.2.2 Typical ACS Observing Sequence	1-7
1.2.3 Data Storage and Transfer	1-8
1.2.4 Parallel Operations	1-8
Chapter 2: ACS Data Structure	2-1
2.1 Types of ACS Files	2-2
2.1.1 Data Files and Suffixes.....	2-2
2.1.2 Association Tables.....	2-2
2.1.3 Trailer Files	2-4
2.2 ACS File Structure	2-5
2.3 Data Storage Requirements.....	2-7
2.4 Headers and Keywords	2-9
Chapter 3: ACS Calibration Pipeline	3-1
3.1 On The Fly Reprocessing (OTFR)	3-2
3.2 Pipeline Overview	3-3
3.2.1 CALACS: Image Calibration	3-3
3.2.2 PyDrizzle: Distortion Correction and Dither Combination.....	3-5
3.2.3 When is OTFR not Appropriate?.....	3-5
3.3 Structure of CALACS	3-6

3.4 CALACS Processing Steps	3-9
3.4.1 ACSCCD.....	3-10
3.4.2 ACSREJ.....	3-15
3.4.3 ACS2D.....	3-18
3.4.4 ACSSUM	3-25
3.5 Manual Recalibration of ACS Data.....	3-25
3.5.1 Requirements for Manual Recalibration	3-25
3.5.2 CALACS Examples.....	3-28

Chapter 4: Distortion Correction and Drizzling of ACS Images..... 4-1

4.1 ACS Geometric Distortions.....	4-2
4.1.1 Distortion Model.....	4-2
4.1.2 On-orbit Calibration Programs	4-4
4.1.3 Results.....	4-5
4.1.4 Distortion Table (IDCTAB).....	4-12
4.2 Combination of Dithered Data.....	4-14
4.2.1 Introduction to Dithering.....	4-14
4.2.2 Introduction to Drizzling	4-15
4.2.3 Overview of Drizzling Software Tools.....	4-16
4.3 PyDrizzle	4-17
4.3.1 Application by the Pipeline.....	4-17
4.3.2 Association Tables.....	4-19
4.3.3 PyDrizzle Data Products.....	4-19
4.4 Reprocessing with PyDrizzle.....	4-20
4.4.1 Specifying the ‘Bits’ Parameter.....	4-21
4.4.2 Creating Custom Association Tables.....	4-22
4.4.3 Applying User-defined Shifts	4-24
4.4.4 PyDrizzle Examples.....	4-28
4.5 MultiDrizzle	4-34
4.5.1 Reprocessing with MultiDrizzle.....	4-35
4.5.2 Tasks Performed by MultiDrizzle	4-36
4.5.3 Specifying Shifts Between Images	4-37
4.6 MultiDrizzle Example.....	4-37
4.6.1 Detailed Processing Steps & Parameters.....	4-39

Chapter 5: ACS Error Sources	5-1
5.1 Readout Noise and A-to-D Conversion	5-1
5.1.1 Readout Noise	5-1
5.1.2 A-to-D Conversion	5-2
5.2 Bias Subtraction	5-2
5.2.1 Bias Calibration Issues	5-2
5.2.2 Bias Jump	5-5
5.2.3 Bias Subarrays for the WFC and HRC	5-6
5.3 Dark Current, Hot Pixels, and Cosmic Rays	5-6
5.3.1 Dark Current	5-6
5.3.2 Hot Pixels	5-9
5.3.3 Cosmic Rays	5-13
5.4 Flat Fields	5-17
5.4.1 Ground Flats (P-flats)	5-17
5.4.2 Inflight L-flat Correction	5-18
5.4.3 Pipeline Flats	5-21
5.4.4 Earth Flat Verification	5-24
5.4.5 Sky Flats	5-24
5.5 Image Anomalies	5-25
5.5.1 Dust Motes	5-25
5.5.2 Optical Ghosts and Scattered Light	5-26
5.5.3 Cross Talk	5-30
5.5.4 Scattered Earth Light	5-36
5.6 Generic Detector and Camera Properties	5-37
5.6.1 Gain Calibration	5-37
5.6.2 Full Well Depth	5-39
5.6.3 Linearity at Low to Moderate Intensity	5-39
5.6.4 Linearity Beyond Saturation	5-43
5.6.5 Shutter Stability	5-45
Chapter 6: ACS Data Analysis	6-1
6.1 Photometry	6-1
6.1.1 Photometric Systems, Units, and Zeropoints	6-1
6.1.2 Aperture and Color Corrections	6-4
6.1.3 Pixel Area Maps	6-5
6.1.4 PSF	6-11
6.1.5 CTE	6-14
6.1.6 Red Leak	6-21
6.1.7 UV Contamination	6-22

6.2 Astrometry.....	6-23
6.2.1 Coordinate Transformations	6-23
6.2.2 Absolute and Relative Astrometry	6-24
6.2.3 Impact of Guide Star Failure.....	6-25
6.3 Polarimetry.....	6-26
6.3.1 Absolute Calibration.....	6-26
6.3.2 Instrumental Issues.....	6-27
6.3.3 Flats.....	6-30
6.3.4 Polarization Calibration.....	6-30
6.4 Coronagraphy	6-32
6.4.1 Flat Fielding	6-33
6.4.2 PSF Subtraction.....	6-34
6.4.3 Photometry	6-34
6.4.4 Ghosting	6-35
6.5 Ramp Filters	6-35
6.5.1 Introduction	6-35
6.5.2 Existing Ground Calibrations	6-35
6.5.3 Future In-flight Calibrations.....	6-36
6.6 Spectroscopy with the ACS Grisms and Prisms.....	6-37
6.6.1 What to Expect from ACS Slitless Spectroscopy Data.....	6-38
6.6.2 Pipeline Calibration.....	6-41
6.6.3 Extracting and Calibrating Slitless Spectra.....	6-42
6.6.4 How to Handle Dithered Slitless Spectroscopy Data.....	6-47
6.6.5 Accuracy of Slitless Spectra Wavelength and Flux Calibration	6-48

Part III: Appendixes

Appendix A: IRAF Primer	A-1
A.1 Initiating IRAF	A-2
A.1.1 Setting Up IRAF in Unix/Linux	A-2
A.1.2 Starting and Stopping an IRAF Session	A-4
A.2 IRAF Basics	A-4
A.2.1 Loading Packages	A-5
A.2.2 Running Tasks	A-6
A.2.3 Getting Help	A-7
A.2.4 Setting Parameters	A-9
A.2.5 Setting Environment Variables	A-11
A.2.6 File Management	A-12
A.2.7 Troubleshooting	A-14
A.3 Getting IRAF and STSDAS	A-15
A.3.1 Retrieving the IRAF and STSDAS Software	A-15
A.3.2 Getting the Synphot Database	A-16
A.3.3 Extracting the Synphot Unix Tar Files	A-17
Appendix B: HST File Names	B-1
B.1 Rootnames	B-3
B.2 Suffixes of Files Common to all Instruments	B-4
B.3 Associations	B-5
Appendix C: Observation Logs	C-1
C.1 Observation Log Files	C-1
C.1.1 Observation Log File Contents (October 1994 version)	C-3
C.1.2 Observation Log File Contents (February 1997 version)	C-4
C.1.3 Jitter File Contents (February 2003 Version)	C-9
C.2 Retrieving Observation Logs	C-11
C.3 Using Observation Logs	C-12
C.3.1 Guiding Mode	C-12
C.3.2 Guide Star Acquisition Failure	C-14
C.3.3 Moving Targets and Spatial Scans	C-14
C.3.4 High Jitter	C-15
Index	1

Preface

The HST Data Handbook is comprised of three separate sections which are merged together to form a single document:

- Part I is a general introduction which describes the process of retrieving and reducing Hubble Space Telescope (HST) data.
- Part II is an instrument-specific document which describes the reduction procedures, calibrations, and sources of error specific to each active HST instrument.
- Part III is a general set of appendices which includes an IRAF primer, a description of HST file names, and a summary of the observation log files.

Use of HST data necessarily involves using software to retrieve, analyze, and view it. With regard to analysis and visualization, there are many different tools and packages available. It would be impractical for a handbook to show how to do this for all available software. Since much of the software developed by STScI for calibrating, analyzing, and visualizing HST data has been based on the IRAF system, the focus of this handbook will be on use of IRAF, STSDAS, and TABLES for doing so.

Chapter 3 will briefly mention other software tools for accessing HST data and where to get more information about these tools. PyRAF is a new command language (CL) for IRAF and as such, allows use of almost all IRAF tools. It is briefly discussed in Chapter 3, but for the most part, the IRAF examples shown will work exactly the same for PyRAF.

The specifics of the data produced by each of these five active instruments (FGS, WFPC2, STIS, NICMOS, and ACS) are described in separate versions of Part II, while the general information from Parts I and III, referred to as ‘the introductory chapters’, are tacked onto the beginning and end of each. We recommend a careful reading of the introductory chapters before proceeding to the instrument-specific data handbooks and before starting to work on your HST data.

The present introductory chapters are based on information available as of June 2004. Many changes in the HST Data Archive and HST data reduction software have occurred since the last revision of the Hubble Data Handbook introductory chapters.

Future changes in this handbook are anticipated as the Multimission Archive at STScI (MAST) expands to cover additional missions, and as StarView, PyRAF, and STSDAS software continue to evolve. The reader is advised to consult the STScI web site at <http://www.stsci.edu/hst> for the latest information.

(Editors, HST Introduction)



PART I:

Introduction to Reducing the HST Data

The chapters in this part provides an introduction to the process of retrieving and reducing Hubble Space Telescope (HST) data.

Getting HST Data

In this chapter. . .

1.1 Archive Overview / 1-2
1.2 Getting Data with StarView / 1-4
1.3 Getting Data with the World Wide Web / 1-16
1.4 Reading HST Data Disks / 1-21

This chapter describes how to obtain Hubble Space Telescope (HST) data files. All HST data files are stored in the Hubble Data Archive (HDA), which forms part of the Multimission Archive at STScI (MAST)¹. HST Guaranteed Time Observers (GTOs), Guest Observers (GOs) and Archival Researchers can retrieve data in either of two ways:

- Electronically over the Internet from the HDA, where data are stored immediately after they pass through HST pipeline processing.
- On data storage media written at STScI from the HDA. Data can be written to CDs or DVDs.

Non-proprietary data in the HDA can be retrieved electronically either by registered HDA users or via anonymous login. However most GO and GTO observations carry a proprietary period of up to one year after observation. Proprietary data may not be retrieved except by a registered HDA user who has the permission of the program's Principle Investigator (PI). Note that HST PIs are *not* automatically registered. PIs should register before their first observations have been taken. All calibration observations as well as observations made as part of the GO Parallel programs are

1. MAST currently includes data from HST, FUSE, GALEX, IUE, EUVE, ASTRO, HUT, UIT, WUPPE, ORFEUS, BEFS, IMAPS, TUES, Copernicus and ROSAT. Data from the FIRST radio survey, Digital Sky Survey (DSS) and Sloan Digital Sky Survey (SDSS) are also available.

immediately public. All observations made as part of the Treasury Programs begun in Cycle 11 will either be immediately public or have only a brief proprietary period. The High-Level Science Products section of MAST also contains several sets of fully reduced HST data, including the Hubble Deep Field, the Ultra Deep Field, and the GOODS Treasury program. These data are also public. As of September 2002, WFPC2 associations are also available through MAST. Read more about this in Section 1.3.1.

This chapter describes how to search the HDA, how to electronically retrieve files from it, and how to request and read disks containing HST data. As an aid to retrieving their data, PIs will automatically receive e-mail notification of the status of their observations two times: first, when the first datasets for their proposal are archived, and second, when all the datasets for their proposal and all necessary calibration files have been archived.

Note for Advanced Camera for Surveys (ACS) Users: Calibrated ACS images are approximately 168 MB in size, larger than those of any other HST instrument. Therefore, the preferred option for data retrieval is from the HDA staging disk via ftp/sftp. Users retrieving large numbers of ACS files should also consider requesting them on DVDs. In addition, the archive recommends to ask for compressed data, which distinctly shortens the retrieval times without any information loss.



1.1 Archive Overview

The HDA contains all HST observations ever made. It also contains a database that catalogs and describes these observations. The archive provides On-the-Fly Reprocessing (OTFR) of HST data for the following instruments: WFPC2, NICMOS, and STIS. The OTFR system reconstructs FITS data files from original telemetry and calibrates data at the time that a user's request for the data is processed. Through this system, users obtain data calibrated with up-to-date reference files, parameters, and software. There are currently two ways to search and retrieve data from the HDA. The first is a program called StarView, which acts as an interface to the HDA. StarView currently runs as Java-based, stand-alone application that can be downloaded from the web site <http://starview.stsci.edu/html/>. Previous versions of StarView, such as XStarView, are no longer available. The second search and retrieval method is through the HST section of the MAST web site, <http://archive.stsci.edu>. StarView is the more powerful of

the two methods, and in particular allows an examination of the calibration files applied to a given data file. StarView also provides an interface to the Visual Target Tuner (VTT) in the Astronomer's Proposal Tools (APT) suite of programs. The VTT interface can display archive observations on a Digital Sky Survey (DSS) image alongside planned observations. StarView is thus recommended for observation planning, duplication checking, calibration file review, investigation of On-The-Fly Reprocessing flags and proprietary status. It is also recommended for those needing to retrieve large numbers of datasets, and those needing to examine calibration files. The MAST web site interface to the HDA has the same basic capabilities as StarView, and may be preferable for those requiring simple retrievals of datasets. Both StarView and the MAST web site allow cross-qualified searches of the other MAST mission archives for all HDA searches. They also offer simple preview of HST datasets when available, as well as links to references citing a given dataset using the Astrophysics Data System (ADS). In later sections, StarView and the MAST web site are discussed in more detail.

1.1.1 Archive Registration

The simplest way to register and retrieve HST data is to complete the form on the Web page at: <http://archive.stsci.edu/registration.html>. If problems occur, registration requests may also be sent to the HDA hotseat, at: archive@stsci.edu.

The PI of each HST proposal must request access to their proprietary data for themselves, and for anyone else whom the PI wants to have access to it. PI retrieval permission is not granted automatically, for security reasons. PIs wishing to allow access to their proprietary data should make that request to archive@stsci.edu.

When registration is granted, your account will be activated automatically, and you will receive your username and password via e-mail.

1.1.2 Archive Documentation and Help

The MAST web site provides a wealth of useful information, including an online version of the HST Archive Manual available at <http://archive.stsci.edu/hst/pub.html>. Investigators expecting to work regularly with HST and other datasets supported by MAST should also subscribe to the MAST electronic newsletter by sending an e-mail to archive_news-request@stsci.edu and putting the single word *subscribe* in the body of the message. Questions about the HDA can be directed to archive@stsci.edu, or by phone to (410) 338-4547.

1.2 Getting Data with StarView

1.2.1 Downloading and Setting Up StarView

The latest version of StarView runs under versions 1.3 and later of Java and may be downloaded from <http://starview.stsci.edu>. Alternatively, a version of Starview is bundled with the Astronomer's Proposal Tools and that version enables more graphical interface between StarView and the VTT. This version may be downloaded with APT, see <http://apt.stsci.edu> for more information.

This StarView site also includes a FAQ page and news on releases and updates. StarView will automatically update itself to the latest version, so users do not have to worry about additional installations. Following its installation on computers running Unix and Linux, begin StarView by typing

```
> StarView
```

at the system prompt. Under Windows and Mac systems, StarView will appear as an icon. The StarView session then begins, first with an Information window explaining navigation within StarView, and a request for the user to specify an object name resolver (SIMBAD or NED) for use in HDA searches. First-time users are asked to supply their e-mail information in order to allow StarView to communicate the results of its attempts to retrieve the files requested from the HDA. This e-mail information includes the user's SMTP host, or the computer from which e-mail messages are routed. If unsure of your SMTP host, ask your system administrator. These queries can be turned off for future sessions once this information has been supplied.

1.2.2 Simple Use of StarView

We now proceed to an introduction to the use of StarView. A more detailed description of its capabilities is provided at the web site above, which should also be consulted for more advanced topics such as its Table Exportation and Cross-Qualification functions.

The basic function of StarView is to enable the user to first search the HDA (and the other mission archives in MAST) for data files matching criteria such as object name, position, or proposal number, then allow the user to navigate through the set of files matching those criteria, and finally to let the user select files for retrieval. Several options for the type of search that can be performed (e.g. by a particular instrument) will be discussed later.

The design of StarView is similar to that of a Web browser. At its top are pull-down menu bars including File, Searches, and Help. The Help menu offers links to documents including the StarView FAQ page. Beneath these menu bars is a row of buttons that run StarView's basic functions, such as searching, marking files for retrieval, and previewing images. A Help button allows users to display pop-up windows describing the function of the different StarView buttons and windows, by first clicking the Help button, then the item of interest. Beneath the row of buttons is the Qualifications panel, which is displayed when a search is begun. It consists of several cells corresponding to the search parameters the user wishes to use, e.g., object name, proposal ID, or instrument. Below this window will appear the Results panel, displaying the datasets found to match a given set of search parameters entered into the Qualifications panel. For the purpose of introduction, we will describe the use of the most basic search option, called "Quick Search," which can be started by clicking the "Quick" button at the top left of StarView.

As an example of the use of the Quick Search option, we will request all available WFPC2 data for the galaxy M87. This is done by typing "WFPC2" and "M87" in the Instrument and Target Name cells of the Qualifications section, then clicking the "Search" button at the top left of the StarView window. The results of the search will then be displayed in the bottom panel of StarView, as shown in Figure 1.1. These results include the dataset name, instrument name, R.A. and Dec. of the target, and the instrument aperture used. Note that these parameters could also have been specified in the Qualifications section, as can other parameters including proposal ID number, proposal PI name, and image central wavelength (corresponding to particular instrument filters or gratings).

An additional useful example would be to search by coordinates. Click "coords" button, select SIMBAD or NED, and click "resolve". This pushes coordinates into the corresponding qualifiers of the Quick Search panel.

Figure 1.1: Results of StarView Quick Search for WFPC2 files of M87

The screenshot displays the StarView Quick Search interface. At the top, there is a toolbar with icons for Quick, Search, Scan, Prev, Stop, Next, Scan, Mark, All, Unmark, All, and Help. Below the toolbar is a section titled "Enter qualifications for: Quick Search" with a table for inputting search criteria. The search results are displayed in a window titled "Results for: Quick Search", which includes fields for Proposal ID, PI (last name), Target Name, Target Description, Instrument, Config, Start Time, and Flag. A table of results is shown below these fields, listing Dataset Name, RA, Dec, Instrument, Flag, Apertures, and Central Wavelength. The status bar at the bottom indicates "All records retrieved." and shows "Static", "Snap", "Table", "No Update" buttons, along with "30 of 30" records.

Enter qualifications for: Quick Search

Label	Qualification (click cell to edit)	Get Field
Dataset Name:		Info.
RA:		Info.
Dec:		Info.
Radius (degrees):	0:10	Info.
PI (last name):		Info.
Proposal ID:		Info.
Flag:		Info.
Target Name:	M87	Info.
Instrument:	WFPC2	Info.

Results for: Quick Search

Proposal ID: 8725 **Release Date:** 2002-02-24 04:32:27 **Proposal Abstract**
PI (last name): ZEPF
Target Name: M87
Target Description: GALAXY;ELLIPTICAL
Instrument: WFPC2 **Config:** WFPC2 **Instrument Info**
Start Time: 2001-02-23 18:28:13.3 **Flag:** NORMAL **Calibration Info**

Dataset Name:	RA:	Dec:	Instrument:	Flag:	Apertures:	Central Wavelength:
U2900101T	+12 30 49.43	+12 23 27.98	WFPC2	NORMAL	PC1	6590.0
U2900102T	+12 30 49.43	+12 23 27.98	WFPC2	NORMAL	PC1	6590.0
U2900103T	+12 30 49.43	+12 23 27.98	WFPC2	NORMAL	PC1	5479.0
U2900104T	+12 30 49.43	+12 23 27.98	WFPC2	NORMAL	PC1	5479.0
U4TU0101R	+12 30 39.19	+12 27 59.94	WFPC2	NORMAL	PC1-FIX	5957.0
U4TU0102R	+12 30 39.19	+12 27 59.94	WFPC2	NORMAL	PC1-FIX	5957.0

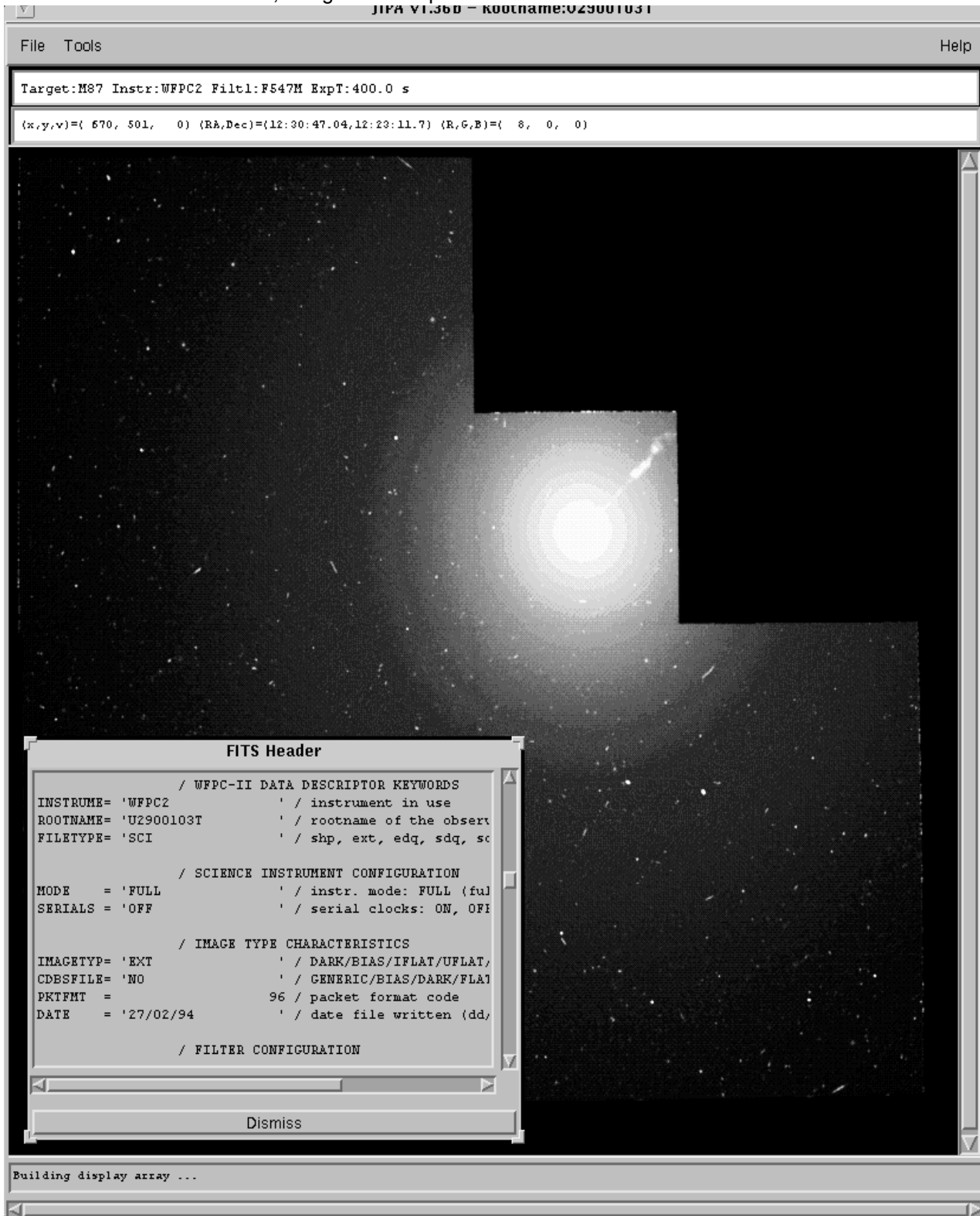
All records retrieved. Static Snap Table No Update 30 of 30

Clicking on a given dataset in the Results panel will display the information shown in the cells above it (Proposal ID, Release Date, PI, etc.). You may browse through the retrieved datasets either by using the mouse and scroll bar, or by using the navigation buttons (Scan, Previous, Next) in the top row of mouse buttons. The Scan option will automatically step through all of the files retrieved in the search, provided that the right most button at the bottom of the Results window is toggled to “Update.” If this button is toggled to “No Update,” the Scan option will go straight to the end of the list of files.

The ability to obtain a preview is available for many, but not all, of the datasets in the HDA (e.g., previews are not available for many FOC datasets). This is done with the “Preview” button, if it is enabled. For images, this will display a re-sampled version of the image using the Java Image Preview Application (JIPA) tool developed by ESO that is part of StarView. For spectra, simple GIF image of the calibrated spectrum will be displayed. JIPA and VTT can also display an image's FITS header. When viewing with JIPA, use the “Tools” menu. The JIPA preview of the WFPC2 image U2900103T retrieved in the previous search for WFPC2 images of M87 is shown in Figure 1.2, along with the window displaying part of the FITS header file of this image.

Other display options with StarView include “DSS,” which will display a 20' x 20' Digital Sky Survey image at the target coordinates, while the “Overlay” button will display the same DSS image with outlines of the HST instrument apertures at the target coordinates superimposed on it, at the orientation of the selected observation. The “References” button provides a link to any known published papers citing the dataset, as listed in ADS. Note that the HST images displayed by Preview are of reduced quality compared to the actual data files, and cannot be downloaded. They are only meant to provide a quick check that the datasets found by the search met the search criteria, i.e., contained the object(s) of interest, and are of the desired quality.

Figure 1.2: JIPA preview of WFPC2 image U2900103T, along with image header file, using Preview option



1.2.3 Marking and Retrieving Data with StarView

Datasets are marked for retrieval by first clicking on them, then using the “Mark” button at the top of StarView. An alternate method is to right-click on the dataset and choose “Mark Dataset(s)/fetch entry” from the menu (or simply Ctrl-M on the dataset). There is also the “All” button, which will mark all the datasets retrieved in the search. Marked datasets will be displayed in the Retrieval window. Datasets still within their proprietary period will be displayed in yellow. Only the proposal PI and those authorized by the PI will be able to retrieve them. The release date of files still within their proprietary period will also be indicated on the retrieval panel.

If satisfied with the marked datasets, choose “Submit” in the Retrieval window to retrieve them. You will then be queried for the type of data files associated with the dataset(s) to be retrieved, the method of delivery of these files, and your archive username and password.

Proprietary data retrieval requires an archive username and password. Public data retrieval can be requested using an archive username and password or by checking the “anonymous” box. For help with your username and password, contact archive@stsci.edu. The options for file type include files calibrated with the On-The-Fly-Recalibration (OTFR) pipeline for the WFPC2, NICMOS, STIS and ACS instruments. OTFR applies the best available calibration files (e.g., dark current and flat field images taken closest in time to the observations) to the uncalibrated data files. You may also request the uncalibrated (raw) files and calibration files separately, or even specify a three letter file extension (e.g. FLT). For some of the earlier instruments, e.g., WFPC and FOS, even though OTFR is not available you may request both the calibration files actually applied to the images, as well as those that should provide the best calibration of them, if recalibration is desired. You may also request Data Quality and Observation Log files.

Options for data delivery include ftp transfer by the user from the HDA staging disk, automatic transfer from the HDA via the ftp and sftp to a host and directory specified by user, and the mailing of CDs and DVDs. If ftp/sftp delivery is specified, you will be queried for the name of the computer and directory in which the files are to be placed, as well as your user name and password on that computer. These requests are encrypted, so there is no danger of your login information being stolen. Upon final submission of the request, you will receive an e-mail message acknowledging its receipt, and another message after all the requested files have been transferred. The status of the request, (i.e., how many files have been transferred and any errors that have occurred), can be checked on a Web page at the address given in the acknowledgment message.

1.2.4 Using StarView to Retrieve Calibration Files and Proposal Information

StarView allows several additional types of searches of the HDA besides the Quick Search option described above. These can be selected from the Searches menu bar at the top of the StarView screen. One such search option is by instrument. This is the option necessary for identifying calibration reference files. As an example, selecting the option “WFPC2 OTFR” under the Instrument and WFPC2 sub-menus of the Searches menu, and then entering “M87” under Target Name in the qualifications box, brings up the screen shown in Figure 1.3. This screen shows all the calibration images and files applied by OTFR to the first file in the set of WFPC2 images of M87, as well as whether the application of these files was performed or omitted in the calibration pipeline. This is the same set of images found by the Quick Search query described above, and the same information for the other datasets from this search can be found using the Previous, Next and Scan buttons. Once these calibration images have been identified, further information on them can be obtained. For example, taking the name of the flat field file found in the above search and entering it into the “WFPC2 Calibration Data” Searches option will retrieve information on and where this file was taken, and the date after which its use is recommended. This will help users decide if they would prefer to recalibrate their data using different files.

StarView can also be used to search for and view the abstracts of accepted HST proposals. Like the Preview capability of StarView, this provides additional information about a given dataset and whether it may be useful for your science goals. Viewing proposal abstracts is an option under the Searches menu, and an example is shown in Figure 1.4. The Qualifications window again offers several parameters by which this search can be constrained, including proposal ID number, HST cycle, PI name, and combinations thereof. In the example shown only the proposal ID number was used.

Finally, StarView can be used during the Phase I proposal process to see whether or not HST observations of a given object or object class have already been made, or else are scheduled for execution. Specifically, the Duplications option under the Searches menu allows users to check a database containing both HDA files and a list of queued observations in order to see if a given object has been or will be observed. Similarly, under Duplications the user may also query the database of proposal abstracts for a given object or object class, to check for archived or scheduled observations. It may be necessary to obtain the target name from NED or SIMBAD first, or to use wildcards in the target name to find all occurrences of a target.

1.2.5 Advanced Features of StarView

In addition to its basic search and retrieval functions, StarView allows users to cross-qualify (“push”) results from separate searches of the HDA or external lists, and to export the results of searches to disk as ASCII files. These operations are performed with the “XQual” and “Export” functions, respectively.

As an example of cross-qualification, a user might want to take a list of targets observed in proposal 8721 and find out what WFPC2 observations were made of these targets. This could be accomplished with the Cross-Qualification search in several ways.

The most straightforward approach would be to first open 2 Quick Search panes. In one, do a Quick search, specifying the proposal ID, e.g. 8721. After searching, make sure that you scan through all the results using the Scan button at the top of StarView. You will find the number of results at the bottom right of the search panel. Next, click on the “XQual” button on the left side of StarView. This brings up a new Cross Qualification window, such as the one shown in Figure 1.5. On the left of this new window, select the search form just run and on the right the blank Quick search form. For each, select the RA and Dec fields by first marking them with the left mouse button and then importing them to “Selected Fields” with the arrow button. You can also simply double click on the respective fields. Make sure they are in the same order, then click the “X-Qualify” button in the middle. This will push the RA and Dec results from the first search into the qualifiers of the new search. On the new search form, add WFPC2 to the Instrument qualifier cell and execute the search. This will return all the datasets from observations using WFPC2 for the targets observed in proposal 8721. Be aware that when pushing multiple results from one search to another, very long lists may cause StarView to run out of memory. So, construct your queries to retrieve only the most relevant data.

Figure 1.3: Results of StarView search for WFPC2 OTFR calibration files for M87

The screenshot displays the StarView software interface. At the top is a toolbar with icons for Quick, Search, Scan, Prev, Stop, Next, Scan, Mark, All, Unmark, All, and Help. Below the toolbar is a search qualification entry form titled "Enter qualifications for: WFPC2 OTFR". This form contains fields for Label, PI (last name), Proposal ID, Target Name (M87), Release Date, Radius (degrees) (0:10), Dec, RA, Dataset Name, and Filter 1. Each field has a "Get Field" button with an information icon.

Below the search form is a detailed results window titled "Results for: WFPC2 OTFR". This window contains the following information:

- PI (last name): ZEPF
- Proposal ID: 8726
- Target Name: M87
- Release Date: 2002-02-24 04:32:27
- RA: +00 49 34.86
- Dec: +187 42 48.99
- Dataset Name: U6600106R
- Filter 1: F170W
- Serials: OFF
- Mode: FULL
- Shutter: A
- A-D Gain: 7.0
- Filter 2:
- Exptime: 1300.0
- Date of Last Software Change (calwp2): 1997-01-03 00:00:00.0
- Date of Last On The Fly Calibration Action Update: 2003-12-10 22:34:23.6

At the bottom of the results window is a table of software switches:

SOFTWARE SWTCH	REFERENCE FILE	OTFR FILE/TABLE	OTFR ACTION
ATODCORR	AtoD Correction	DBU1405IU.R1H	PERFORM
BLEVCORR	Engineering File	U6600106R.X0H	PERFORM
BIASCORR	Bias Correction	L9I12088U.R2H	PERFORM
DARKCORR	Dark Current	L2Q1603NU.R3H	PERFORM
FLATCORR	Flat Field	I5515215U.R4H	PERFORM
MASKCORR	Static Pixel Mask	F8213081U.R0H	PERFORM
SHADCORR	Shutter Shading	E371355EU.R5H	OMIT
DOPHOTOM	Graph Table	N9I1408HM_TM	
	Components Table	NC809508M_TM	

Figure 1.4: Results of the StarView search for the abstract of Proposal 8725

File Edit View Searches Tools Window Help

Quick Search Scan Prev Stop Next Scan Mark All Unmark All Help

Enter qualifications for: Proposal Abstract

Label	Qualification (click cell to edit)	Get Field
Proposal Type:		Info.
Cycle:		Info.
TAC Panel:		Info.
Title:		Info.
PI (Last name):		Info.
Abstract:		Info.
Proposal ID:	8725	Info.

Science Only
 OR Logic

Results for: Proposal Abstract

Proposal ID: 8725

Proposal Type: GO Cycle: 9

TAC Panel: GALAXIES

PI (Last name): Zepf

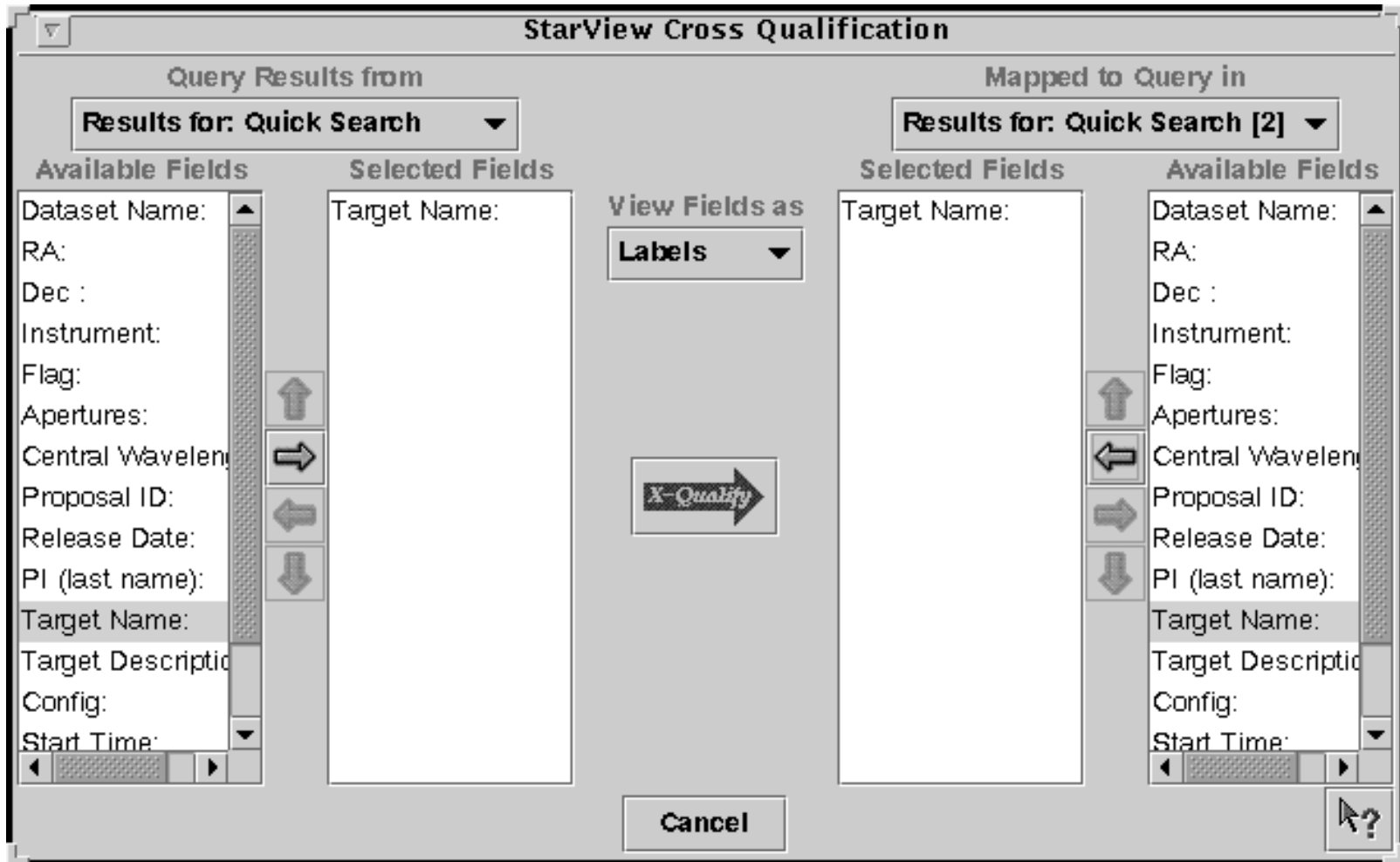
Title: The Formation Epoch(s) of Globular Clusters Around Ellipticals f

Abstract: We propose to study the formation history of elliptical galaxies by obtaining far-UV photometry of their globular clusters. The far-UV photometry will be used in conjunction with existing optical data to constrain the ages of these clusters, and thus the formation epochs of their host galaxies. Optical data show that globular cluster systems of ellipticals often have bimodal metallicity distributions.

One record matched search criteria Static Snap Table No Update 1 of 1

XQual Export Preview Overplot DSS Overlay Refs.

Figure 1.5: Example of Cross Correlation Feature, in which Target Name has been chosen as the common feature to search for in two Quick Search result lists.



1.2.6 StarView and the Visual Target Tuner

The Visual Target Tuner is part of the Astronomer's Proposal Tools (APT) package, which has been created to aid astronomers in planning their HST observations during the Phase I and Phase II proposal stages (see <http://apt.stsci.edu>). VTT is an image display tool which allows the user to display DSS images or local FITS images with proper World Coordinate System keywords in the headers. It offers more features than JIPA, which is the default StarView display tool. However, for a limited number of operating systems, the VTT can be used with StarView. VTT offers the particular advantage that it can overlay the instrument apertures of multiple observations on a single DSS image. Clicking on these apertures will also highlight the associated datasets in StarView.

Currently, to combine StarView and VTT requires downloading and installing APT from the above Web site. APT is only available for those operating systems with the Java Virtual Machine 1.4.1 (JVM 1.4.1). You can download the StarView/VTT package with JVM 1.4.1 included (a large download). Note, for Mac users, STScI provides a smaller APT/VTT package without JAVA. This option is only available for Mac users. To make StarView use VTT, you must change your Viewer options from JIPA to VTT. Go to the preferences sub-menu of Edit in StarView, and change JIPA to VTT in the Viewers section. If VTT is not listed here, you should reinstall the two programs. Following this change, the Preview, DSS and Overlay buttons of StarView should all bring up VTT.

Alternatively, once VTT has been installed, you can also bring up StarView from it. Enter into StarView mode by clicking on the StarView button in the lower left hand menu of VTT. Clicking on a DSS image will then spawn a Quick StarView screen with the R.A. and Dec of the position you clicked loaded into the search fields. You can enter other constraints into these fields as usual. Search results can be displayed on the VTT screen by selecting the results in StarView, and pressing the Overlay button.

1.2.7 Quick Proprietary Data Retrieval with StarView

The following steps summarize the basic process that users need to go through to retrieve their data with StarView. These steps follow registration as a MAST user, notification from STScI that the observations for a given proposal are complete, and providing StarView with your e-mail information. They are intended as a quick reference for this process.

1. Start StarView.
2. Click the "Quick" button.
3. Enter your PI name and/or proposal ID number in the appropriate cell.

4. Click the “Search” button.
5. Use the “Scan” button to step through the retrieved files, after toggling the right most button at the bottom of the Results window to “Update” to verify that all datasets have been retrieved.
6. Preview some or all of the datasets if desired, to verify data quality and target acquisition.
7. Click “All” to mark all datasets for retrieval, or “Mark” to mark individual datasets for retrieval.
8. Click “Submit” in the window that will be spawned by marking the files.
9. Enter your MAST username and password and specify the means of data delivery. StarView remembers your name and password from past searches so it does not have to be entered each time.
10. Click “Done”, and your data are on their way. You will receive an e-mail message when your retrieval has been queued, and another when the transfer is complete.

1.3 Getting Data with the World Wide Web

HDA datasets can be searched for, previewed and retrieved via the World Wide Web in very much the same way as with StarView. As noted in Section 1.1, StarView offers more capabilities for this process, including cross-qualification, the use of VTT, and more information about instrument calibration files. However, Web retrievals may be preferable in some cases, particularly when information on calibration files is not needed, and the hypertext on the Results pages makes it easy to access all the information they contain. The starting point for Web-based searches of the HDA is the MAST web site at: <http://archive.stsci.edu>²

This web page is shown in Figure 1.6. A powerful feature of MAST is that all of its mission archives, including the HDA, can be searched simultaneously. This is done with the Quick Target Search option shown on the MAST home page. This search will return all datasets for all missions available for a given object or coordinates, according to the search constraints specified by the user (based on the wavelength region of interest), and will provide hypertext links to these datasets. If only HST datasets are desired, they can be accessed separately by clicking “HST” on

2. European archive users should generally use the ST-ECF Archive at <http://archive.eso.org>. Canadian users should request public archival data through the CADAC web site at <http://cadcwww.dao.nrc.ca>. Proprietary data are only available through STScI.

the MAST home page. Searches of the HDA by object class can also be made with the VizieR Catalog Search tool at:

<http://archive.stsci.edu/vizier.php>

Figure 1.6: MAST Home Page

The screenshot shows a Netscape browser window displaying the MAST Multimission Archive website. The browser's address bar shows the URL <http://archive.stsci.edu/>. The website header includes navigation links for "Data Search / Missions / Contacts / STScI / MAST" and a "Site Search" box. The main heading is "MAST Multimission Archive at Space Telescope".

On the left side, there is a vertical navigation menu with the following items: "About MAST", "MAST Search Toolbox", "VizieR/MAST Cross Correlation Search", "MAST Scrapbook", "MAST Coplotter", "What's New", "FAQ", "High-Level Science Products", "Software", "FITTS", "Related Sites" (listing ADS, HEASARC, IRSA, LAMBDA, NED, NSSDC), "MAST and the VO", and "Acknowledgments".

The main content area features a section titled "Missions" and "Catalogs & Surveys". Under "Missions", there are links for HST, ASTRO, ORFEUS, Copernicus, EUVE, HUT, BEFS, ROSAT, IUE, UIT, IMAPS, and WUPPE. Under "Catalogs & Surveys", there are links for GALEX, SDSS, GSC, DSS, and VLA-FIRST.

Below this, there is a section for "NSSDC Legacy Missions" with two news items:

- April 8, 2004 High-Level Science Products for the Hubble Ultra Deep Field and the ACS-WFC raw data and associated calibration files are available via ftp from STScI, ST-ECF, and the CADC.
- March 9, 2004 NASA announces the GALEX Investigator Program 2004 (proposals due April 16th). See <http://galex.stsci.nasa.gov>.

The central feature is a "Quick Target Search and/or Mission Search" form. It includes:

- A text input field for "Enter Target name (or Coordinates):".
- A "Resolver:" section with radio buttons for "SIMBAD" (selected) and "NED".
- A section for "and/or Band/Data Type(s):" with a grid of checkboxes:

	X-Ray	Extreme UV	Far UV	Near UV	Optical	Near IR	Radio
Imager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spectra	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
- "Search", "Reset", and "Help" buttons at the bottom.

At the bottom of the page, there are links for "Top of Page", "Copyright Notice", "printer-friendly page", and "http://archive.stsci.edu/index.html". The footer also includes the email "archive@stsci.edu" and the date "Modified: Apr 08, 2004 16:42". The browser status bar at the very bottom shows "Document: Done (0.815 secs)".

The HST section of MAST offers tutorials about the HDA as well as a FAQ page and HDA news. It also provides links to HST “Prepared” datasets such as the Ultra Deep Field and the Hubble Deep Field images. Clicking on the “Main Search Form” option of the HST section brings up the page shown in Figure 1.7. Here the user is queried for the same search parameters as requested by StarView, e.g., Object Name, Instrument and Proposal ID. Once these are entered, clicking the Search button returns a page listing the datasets found, which can then be selectively marked for retrieval. The data type and retrieval options remain the same as those for StarView. Previews of GIF files of most datasets are also available by clicking on the image name. Datasets marked for retrieval can be retrieved via ftp, sftp, or can be placed on the Archive staging disk. The preferred format for ftp is compressed as gzip. Data placed on the staging disk (the preferred method) can be retrieved using ftp with the archive user name and password.

Figure 1.7: HST Archive Web search Form

The screenshot shows a Netscape browser window displaying the HST Archive Web search Form. The browser's address bar contains the URL `http://archive.stsci.edu/hst/search.php`. The page title is "HST Search Form". The form is organized into several sections:

- Standard Form:** Contains a "Search" button, a "Reset" button, and a "Clear Form" button.
- File Upload Form:** Contains a "Clear Form" button.
- Search Parameters:** Includes fields for "Target Name", "Right Ascension", "Declination", "Resolver" (set to "SIMBAD"), "Radius (arcmin)" (set to "3.0"), and "Equinox" (set to "J2000").
- Instrument Selection:** Divided into "Imagers", "Spectrographs", and "Other" categories, each with "ALL" and "NONE" radio buttons. Under "Imagers", there are checkboxes for STIS, NICMOS, WFPC2, WF/PC, FOC, and ACS. Under "Spectrographs", there are checkboxes for STIS, NICMOS, GHRS, and FOS. Under "Other", there are checkboxes for PGS and HSP.
- Advanced Search:** Includes fields for "Start Time", "Exp Time", "Proposal ID", "Release Date", "Dataset", "Filters/Gratings", "Obset ID", "Archive Date", "Target Descrip", "Apertures", "Observations" (with a checked "Science" checkbox and an unchecked "Calibration" checkbox), and "PI Last Name".
- User-specified fields:** Two sections for "User-specified field 1" and "User-specified field 2", each with a "Field Descriptions" dropdown menu.
- Output and Sorting:** Includes an "Output Columns" list (with "Mark" selected), "up", "down", "remove", and "reset" buttons, and an "add" button. The "Sort By" section has dropdowns for "ang_sep (?)", "Target Name", and "Dataset", each with a "Reverse" checkbox. The "Output Format" is set to "HTML_Table". There are checkboxes for "Show Query" and "Make Rows Distinct", and a "Maximum Records" field set to "100".
- Navigation:** "Search", "Reset", and "Clear Form" buttons are repeated at the bottom of the form.

1.3.1 WFPC2 Associations

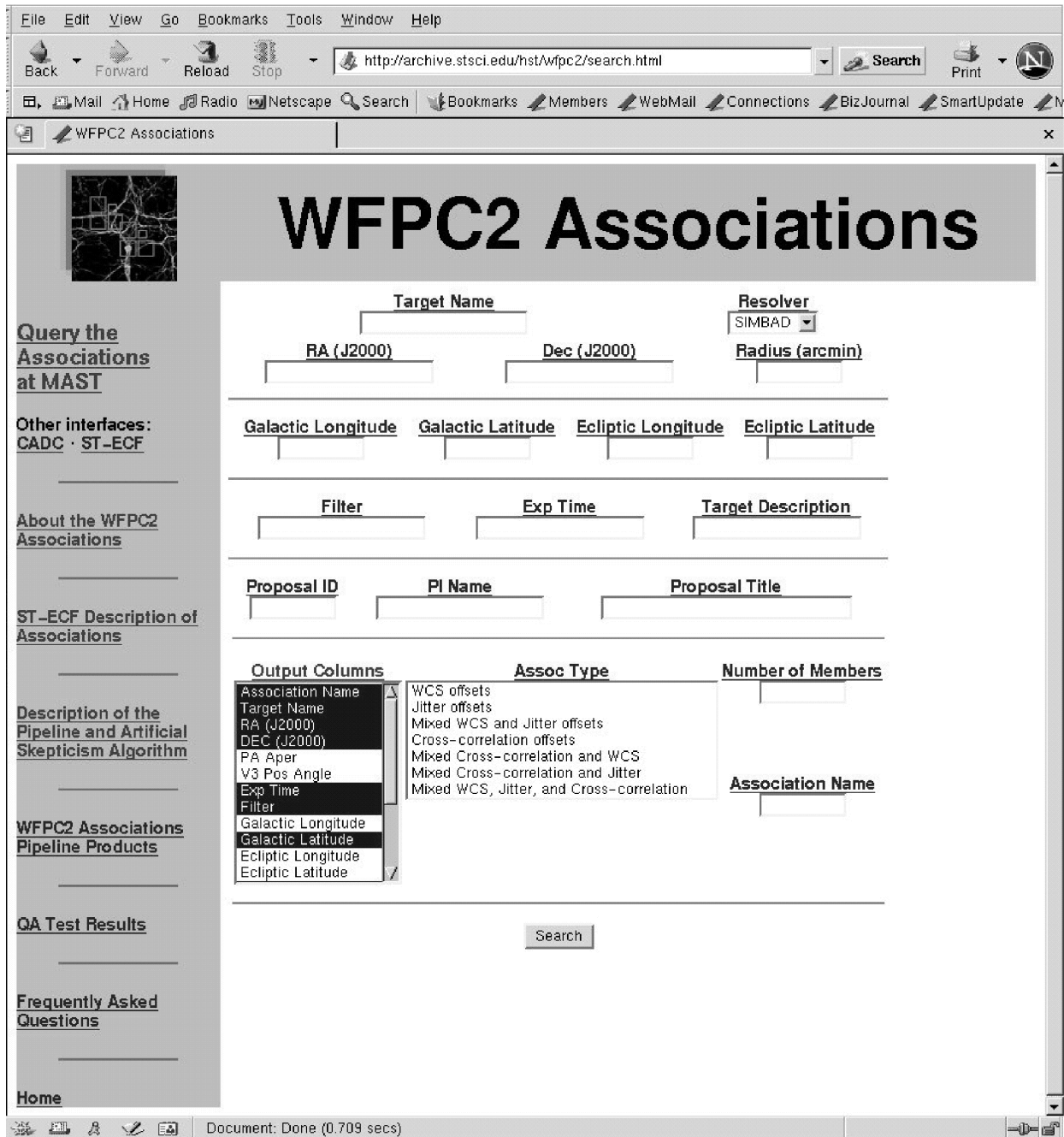
Since November 2002, MAST contains WFPC2 associations, more than 15000 combined WFPC2 images stacked by CADC and ST-ECF (<http://archive.stsci.edu/hst/wfpc2/about.html>). These combined images are the products of the basic registration and averaging of related sets of WFPC2 images, referred to as *associations*, that is usually performed by archival researchers after the retrieval of individual images. The WFPC2 constituent images all have the same orientation on the sky and hence no rotation procedure is used to stack them. They form an important new set of high-quality astronomical data that should be useful for a wide range of investigations. In addition, they provide better previews of a given field than the individual WFPC2 images. The web page also provides a project description and the pipeline procedure applied.

Figure 1.8 shows the query form that can be used to search for WFPC2 associations in MAST. Entering a target like M87 and clicking on the search button will return you to a screen with the respective datasets in FITS format. The data can be downloaded directly from this page.

1.3.2 High Level Science Products

MAST also contains a number of High Level Science Products, which are accessible at http://archive.stsci.edu/prep_ds.html. High-Level Science Products (HLSP) are fully processed (reduced, co-added, cosmic-ray cleaned, etc.) images and spectra that are ready for scientific analysis. HLSP also include files such as object catalogs, spectral atlases, and README files describing a given set of data. The data originate from the Treasury, Archival Legacy and Large Programs (TALL) of cycle 11 and 12, but contain contributions from smaller HST programs and other MAST missions. Users who are interested in contributing to the HLSP, are referred to the Guidelines for Contributing High-Level Science Products to MAST (http://archive.stsci.edu/hlsp/hlsp_guidelines.html, please make sure you got the latest version). Furthermore, they are asked to contact the archive scientist involved as soon as they start working on the data.

Figure 1.8: HST Archive Web search FormWFPC2 Associations Screen



1.4 Reading HST Data Disks

If you request HDA files on CDs or DVDs, you will receive them within a few days of your request. The datasets will all be in FITS (Flexible Image Transport System) format³. On disk, the data is placed in subdirectories based on the rootname.

Currently, datasets obtained with HST's original instruments (FGS, FOC, FOS, GHRS, HSP and WFPC) as well as one of the current instruments WFPC2, must have their FITS files converted to GEIS (Generic Edited Information Set) format in order to work on them with IRAF/STSDAS. Further information on HST file formats is presented in Chapter 2. STSDAS is the package analysis software for HST data, and is discussed further in Chapter 3. Datasets obtained with the other current HST instruments (ACS, NICMOS and STIS) should be reduced in FITS format without conversion to GEIS. (STSDAS support for the analysis of WFPC2 data in FITS format is currently planned.)

The steps for reading and converting FITS files to GEIS files are as follows:

First bring up IRAF/STSDAS in your IRAF home directory by typing

```
> cl
```

If this does not start up an IRAF session, you will need to type “mkiraf” first and select an appropriate terminal window. See Appendix A.3

This will start an IRAF session. IRAF and STSDAS are organized into *packages*. To load a package, type its name. To begin with, you must load the **stdas** and **fitsio** (FITS Input/Output) packages:

```
cl> stdas
st> fitsio
```

The IRAF prompt (such as st>) shows the first two letters of the most recently loaded package. The **fitsio** package contains the STSDAS programs (called *tasks* in the IRAF/STSDAS environment) required to read and write FITS files to and from tapes and disks. The two principle tasks are **strfits** for reading files, and **stwfits** for writing them.

Next, set the IRAF environment variable *imtype* to specify that your data files are to be written in GEIS format. This is done by typing

```
fi> set imtype="hhh"
```

3. A description of FITS format and various supporting documents can be found at the Web site http://fits.gsfc.nasa.gov/fits_home.html

You should then move to the directory containing the FITS files.

The last step is to use **strfits** to read the data. Like most IRAF/STSDAS tasks, **strfits** has several parameters that control its function. You can either edit these tasks using the IRAF “epar” command, or specify them on the command line. For the purpose converting FITS files to GEIS files, the important parameter is *oldirafname*, which needs to be set to “yes” in order to keep the file rootname the same. To convert all the FITS files in a directory to GEIS files, type

```
fi> strfits *.fits " " oldirafname=yes
```

This command will make GEIS format copies (having extension “.hhh”) of all the FITS files in the directory with the same rootname. Following reduction and analysis of the GEIS files with the IRAF/STSDAS tasks, they may be written back into FITS format, on hard disk or to a tape or other storage media, with the **stwfits** task.

HST File Formats

In this chapter. . .

2.1 FITS File Format / 2-2
2.2 GEIS File Format / 2-11

STScI automatically processes and calibrates all the data received from HST. The suite of software programs that performs this processing—part of a system known as OPUS—is frequently called the *pipeline*, and its purpose is to provide data to observers and to the HST Data Archive in a form suitable for most scientific analyses. For the older instruments (FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2) pipeline processing assembles data received from HST into *datasets*, calibrates the data according to standard procedures described in the instrument sections (Part II) of this handbook, and stores both calibrated and uncalibrated datasets in the Archive. For newer instruments, calibrated data is no longer stored in the archive. The OTFR System generates calibrated data as it is requested from the archive.

Pipelines of the older instruments generate files in GEIS (which stands for Generic Edited Information Set) format. Since GEIS is a machine-dependent format, these files are converted to a specific kind of FITS file format, sometimes referred to as “waiver” FITS, before being archived. We'll explain the structure of this “waiver” FITS format later in this chapter. Since the “waiver” FITS format is only designed for archival purpose, it is necessary to convert it back to the GEIS format before further data processing and analysis using IRAF/STSDAS tasks.

Instruments installed after the 1997 servicing mission (STIS, NICMOS, and ACS) have pipelines which generate FITS files directly. They are ready to be used by relevant IRAF/STSDAS tasks and, unlike the “waiver” FITS files, do NOT need to (and indeed, should not) be converted to GEIS format. Sometimes FITS files for the newer instruments are referred to as

“FITS with extension” or “extended” FITS files. But this can be misleading, since a “waiver” FITS file also has one (ASCII table) extension.

Much confusion has occurred about the two kinds of FITS files archived at STScI. So we would like to repeat this warning one more time:



Older instruments (FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2) generate files in GEIS formats, but are stored and delivered as “waiver” FITS format in the archive, and need to be converted back to GEIS format before processing. Newer instruments (STIS, NICMOS, ACS) generate and store files in FITS format and should not be converted to GEIS.

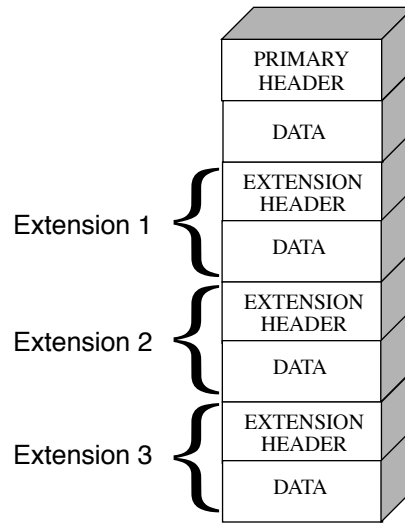
This chapter describes these two HST file formats in more detail. STIS, ACS, and NICMOS observers should pay particular attention to the section on FITS files, which shows how to identify and access the contents of these files and covers some important conventions regarding header keywords. Veteran observers with the other instruments will find little new in the section on GEIS files, but newcomers to the older HST instruments should consult the material on data groups and conversion from FITS to GEIS before proceeding to Chapter 3.

2.1 FITS File Format

Flexible Image Transport System (FITS) is a standard format for exchanging astronomical data between institutions, independent of the hardware platform and software environment. A data file in FITS format consists of a series of Header Data Units (HDUs), each containing two components: an ASCII text header and the binary data. The header contains a series of *header keywords* that describe the data in a particular HDU and the data component immediately follows the header.

The first header in a FITS file is known as the *primary header*, and any number of *extensions* can follow the primary HDU. The data unit following the primary header must contain either an image or no data at all, but each extension HDU can contain one of several different data types, including images, binary tables, and ASCII text tables. The value of the XTENSION keyword in the extension’s header identifies the type of data the extension contains. Figure 2.1 schematically illustrates the structure of a FITS file and its extensions.

Figure 2.1: FITS File Structure



The three-letter identifier (e.g., d0h) that follows the rootname of a GEIS format HST data file (see Appendix B for more on HST file names) has often been called an “extension” in the past. However, because of the potential for confusion with FITS extensions, this handbook will refer to these three-letter identifiers as “suffixes.”



Each FITS extension header has the required keyword XTENSION which specifies the extension type and has one of the following values: IMAGE, BINTABLE, and TABLE, corresponding to an image, binary table, and ASCII table, respectively.

A set of FITS extension images which are logically related to one another is called an *imset*. For example, the error image and the data quality image are in the same *imset* as the science image itself. The keyword EXTNAME is used to specify the extension names of different images in the same *imset*.

For the newer HST instruments (ACS, STIS, and NICMOS), EXTNAME can have the following values: SCI, ERR, DQ, corresponding to science, error, and data quality images. NICMOS files have two additional extension names: SAMP and TIME, for number of samples and exposure time, respectively. Thus, ACS and STIS *imsets* contain three images while NICMOS has five. Images of the same *imset* will have the same integer value in their headers' keyword EXTVER.

Here, the word “image” is used in a general sense. Often, an image has the same value at all pixels (e.g. data quality value). In this case, the extension has no data part and the constant pixel value is stored in the header keyword PIXVALUE. Except for SCI, any of the extensions may use this convention. Typically, SAMP and TIME extensions are single-valued and follow this rule.

2.1.1 Working with FITS Image Extensions

The FITS image kernel included in IRAF version 2.12.2 is designed to read and write the images in FITS extensions and their associated headers. Once IRAF has ingested a FITS image and its header, it treats the header-data pair like any other IRAF image. The following discussion describes how to specify the image extensions in FITS files that you would like to process with IRAF/STSDAS tasks and presumes that you are using IRAF 2.12.2 or higher. It covers how to:

- List a FITS file’s extensions,
- Access data in particular FITS extension,
- Inherit keywords from the primary header,
- Append new extensions to existing FITS files,



Retaining the .fits suffix at the end of every FITS file name in your file specifications will ensure that IRAF both reads and writes these images in FITS format.



If you want to work with NICMOS, STIS, or ACS data, you will need to upgrade to IRAF 2.12.2 or higher and STSDAS 3.2 or higher.

Generating a FITS File Listing

Once you have downloaded STIS, ACS, or NICMOS FITS files from the Archive, you may want an inventory of their contents. To generate a listing of a FITS file’s extensions, you can use the **catfits** task in the **tables** package. The following example, in Table 2.1, illustrates the first 11 lines generated by **catfits** from a NICMOS multiaccum FITS file containing only images.

The first column of a **catfits** listing gives the extension numbers. Note that the primary HDU is labeled extension number zero. The second column lists the extension type whose value is specified in the keyword XTENSION. The third column lists the extension name, given by the keyword EXTNAME.

All images belonging to the same imset share the same integer value of the EXTVER keyword, given in the fourth column of a **catfits** listing. Several STSDAS tasks can work with entire imsets (see Section 3.4.3), but most operate on individual images. See the Data Structure chapters of STIS, ACS, and NICMOS Data Handbooks for more information on the contents of a particular instruments' imsets.

Table 2.1: NICMOS MULTIACCUM Listing from catfits

```
tt> catfits n3t501c2r_raw.fits
```

EXT#	FITSNAME	FILENAME	EXTVE	DIMENS	BITPI	OBJECT
0	n3t501c2r_raw	n3t501c2r_raw.fits			16	n3t501c2r_raw.f
1	IMAGE	SCI	1	256x256	16	n3t501c2r_raw.f
2	IMAGE	ERR	1		-32	
3	IMAGE	DQ	1		16	
4	IMAGE	SAMP	1		16	
5	IMAGE	TIME	1		-32	
6	IMAGE	SCI	2	256x256	16	
7	IMAGE	ERR	2		-32	
8	IMAGE	DQ	2		16	
9	IMAGE	SAMP	2		16	
10	IMAGE	TIME	2		-32	

Accessing FITS Images

After you have identified which FITS image extension you wish to process, you can direct an IRAF/STSDAS task to access that extension using the following syntax:

```
fitsfile.fits[extension number][keyword options][image section]
```

Note that all the bracketed information is optional. However, the only time it is *valid* to provide only a file name without further specification is when the file is a simple FITS file that contains a single image or data block in the primary HDU (e.g. for WFPC2 or for the FGS).

Specifying the extension number is the most basic method of access, but it is not necessarily the most useful. Referring to an extension's EXTNAME and EXTVER in the [keyword options] is often more convenient. If a number follows an EXTNAME, IRAF interprets the number as an EXTVER. For example, if extension number 6 holds the

science image belonging to the imset with EXTVER = 2, as in the **catfits** listing above, it can be specified in two equivalent ways:

```
fitsfile.fits[6]
fitsfile.fits[sci,2]
```

Designating an EXTNAME without an EXTVER refers to the first extension in the file with the specified value of EXTNAME. Thus, `fitsfile.fits[sci]` is the same as `fitsfile.fits[sci,1]`.

The syntax for designating image sections follows the IRAF standard. So, in the current example the specifications

```
fitsfile.fits[6][100:199,100:299]
fitsfile.fits[sci,2][100:199,100:299]
```

both extract a 100 by 200 pixel subsection of the same science image in `fitsfile.fits`.

Header Keywords and Inheritance

STIS, ACS, and NICMOS data files use an IRAF image kernel convention regarding the relationship of the primary header keywords to image extensions in the same file. In particular, IRAF allows image extensions to *inherit* keywords from the primary header under certain circumstances. When this inheritance takes place, the primary header keywords are practically indistinguishable from the extension header keywords. This feature circumvents the large scale duplication of keywords that share the same value for all extensions. The primary header keywords effectively become global keywords for all image extensions. The FITS standard does not include keyword inheritance, and while the idea itself is simple, its consequences are often complex and sometimes surprising to users.

In general, keyword inheritance is the default, and IRAF/STSDAS applications will join the primary and extension headers and treat them as one. For example, using **imheader** as follows on a FITS file will print both primary and extension header keywords to the screen:

```
cl> imheader fitsfile.fits[sci,2] long+ | page
```

Using **imcopy** on such an extension will combine the primary and extension headers in the output HDU, even if the output is going to an extension of another FITS file. Once IRAF has performed the act of inheriting the primary header keywords, it will normally turn the

inheritance feature off in any output file it creates unless specifically told to do otherwise.



If you need to change the value of one of the global keywords inherited from the primary header, you must edit the primary header itself (i.e., “extension” [0]).

Keyword inheritance is not always desirable. For example, if you use **imcopy** to copy all the extensions of a FITS file to a separate output file, IRAF will write primary header keywords redundantly into each extension header. You can suppress keyword inheritance by using the **NOINHERIT** keyword in the file specification. For example:

```
im> imcopy fitsfile.fits[6][noinherit] outfile.fits
im> imcopy fitsfile.fits[sci,2,noinherit] outfile.fits
```

Both of the preceding commands will create an output file whose header contains only those keywords that were present in the original extension header. Note that in the second command, the **noinherit** specification is bracketed with the **EXTNAME** and **EXTVER** keywords and not in a separate bracket of its own, as in the first command where an absolute extension number is used. For a complete explanation of FITS file name specifications, see:

<http://iraf.noao.edu/iraf/web/docs/fitsuserguide.html>.

Appending Image Extensions to FITS Files

IRAF/STSDAS tasks that produce FITS images as output can either create new FITS files or append new image extensions to existing FITS files. You may find the following examples useful if you plan to write scripts to reduce STIS, ACS, or NICMOS data:

If the specified output file does not yet exist, a new output file is created containing only a primary HDU if no specification is appended to the output file name. For example, to copy the contents of the primary header of `fitsfile.fits` into the primary HDU of the FITS file `outfile.fits`, type the command:

```
cl> imcopy fitsfile.fits[0] outfile.fits
```

If the specified output file already exists and you want to append a new extension to it, you need to include the **APPEND** option in the output file specification. The following command appends extension `[sci,2]` of `fitsfile.fits` onto the existing file `outfile.fits`, while retaining

the original EXTNAME and EXTVER of the extension—the `noinherit` specification inhibits the copying of the primary header keywords from the input file into the output extension header:

```
cl> imcopy fitsfile.fits[sci,2,noinherit] \  
>>> outfile.fits[append]
```

In the previous example, and others that follow, note that the backslash is added to indicate that the remainder of the command follows on the next line, after the “>>>” prompt.

If you want to change the EXTNAME or EXTVER of the appended extension, you can specify the new values of these keywords in the output extension, like this:

```
cl> imcopy fitsfile.fits[sci,2,noinherit] \  
>>> outfile.fits[sci,3,append]
```

For obvious reasons, it is not generally advisable for two file extensions in the same FITS file to share the same EXTNAME and EXTVER values. However, if you must append an extension to an output file already containing an extension with the same EXTNAME/EXTVER pair you can do so with the DUPNAME option:

```
cl> imcopy fitsfile.fits[7] \  
>>> outfile.fits[append,dupname]
```

If you need to replace an existing extension with a new output extension, you can use the OVERWRITE option as follows. Overwriting can cause a lengthy rewrite of the whole file to insert the new extension, if its size is not the same as the extension it replaces.

```
cl> imcopy fitsfile.fits[sci,2,noinherit] \  
>>> outfile.fits[sci,2,overwrite]
```

2.1.2 Working with FITS Table Extensions

NICMOS, STIS, and ACS use FITS tables in two basic ways. Each instrument produces association tables (see Appendix B.3) listing the exposures that go into constructing a given association product. In addition, STIS provides certain spectra, calibration reference files, and time-tagged data in tabular form. Here we describe:

- How to access and read FITS table extensions.
- How to specify data arrays in FITS table cells.

This discussion assumes you are using STSDAS 3.2 or later. (The IRAF FITS kernel deals only with FITS images. The **tables** package installed with STSDAS handles FITS table extensions.)

Accessing FITS Tables

You can access data in FITS table extensions using the same tasks appropriate for any other STSDAS table, and the syntax for accessing a specific FITS table is similar to the syntax for accessing FITS images (see Section 2.1.1), with the following exceptions:

- The FITS table interface does not support header keyword inheritance.
- FITS tables cannot reside in the primary HDU of a FITS file. They must reside instead in a FITS table extension, in either ASCII form (XTENSION=TABLE) or binary form (XTENSION=BINTABLE).
- If the first extension in a FITS file is a TABLE or a BINTABLE, you can access it by typing the file name with no extension specified. It is not sufficient for the table to be just the first BINTABLE or TABLE; *it must actually be the first extension.*

For example, running **catfits** on the NICMOS association table `n3tc01010_asn.fits` provides the following output:

```
fi> catfits n3tc01010_asn.fits

EXT#  FITSNAME          FILENAME          EXTVE ...
0      n3tc01010_asn  N3TC01010_ASN.FITS ...
1      BINTABLE      ASN                1 ...
```

Extension number 1 holds the association table, which has EXTNAME=ASN and EXTVER=1. You can use the **tprint** task in the STSDAS **tables** package to print the contents of this table, and the following commands are all equivalent:

```
tt> tprint n3tc01010_asn.fits
tt> tprint n3tc01010_asn.fits[1]
tt> tprint n3tc01010_asn.fits[asn,1]
```

STSDAS **tables** tasks can read both FITS TABLE and BINTABLE extensions, but they can write tabular results only as BINTABLE extensions. Tasks that write to a table in-place (i.e., **tedit**) can modify an existing FITS extension, and tasks that create a new table (i.e., **tcopy**) will create a new extension when writing to an existing FITS file. If the designated output file does not already exist, the task will create a new

FITS file with the output table in the first extension. If the output file already exists, your task will append the new table to the end of the existing file; the APPEND option necessary for appending FITS image extensions is not required. As with FITS images, you can specify the EXTNAME and EXTVER of the output extension explicitly, if you want to assign them values different from those in the input HDU. You can also specify the OVERWRITE option if you want the output table to supplant an existing FITS extension. For example, you could type:

```
tt> tcopy n3tc01010_asn.fits out.fits[3][asn,2,overwrite]
```

This command would copy the table in the first extension of `n3tc01010_asn.fits` into the third extension of `out.fits`, while reassigning it the EXTNAME/EXTVER pair `[asn,2]` and overwriting the previous contents of the extension. Note that overwriting is the only time when it is valid to specify an extension, EXTNAME, and an EXTVER in the output specification.

Specifying Arrays in FITS Table Cells

A standard FITS table consists of columns and rows forming a two-dimensional grid of cells; however, each of these cells can contain a data array, effectively creating a table of higher dimensionality. Tables containing extracted STIS spectra take advantage of this feature. Each column of a STIS spectral table holds data values corresponding to a particular physical attribute, such as wavelength, net flux, or background flux. Each row contains data corresponding to one spectral order, and tables holding echelle spectra can contain many rows. Each cell of such a spectral table can contain a one-dimensional data array corresponding to that cell's physical attribute and spectral order.

In order to analyze tabular spectral data with STSDAS tasks other than the **sgraph** task and the **igi** package, which have been appropriately modified, you will need to extract the desired arrays from the three-dimensional table. Two new IRAF tasks, named **tximage** and **txtable**, can be used to extract the table-cell arrays. Complementary tasks, named **tiimage** and **titable**, will insert arrays back into table cells. To specify the arrays which should be extracted from or inserted into the table cells, you will need to use the *selectors* syntax to specify the desired row and column. The general syntax for selecting a particular cell is:

```
intable.fits[extension number][c:column_selector][r:row_selector]  
or  
intable.fits[keyword options][c:column_selector][r:row_selector]
```

A *column selector* is a list of column patterns separated by commas. The column pattern is either a column name, a file name containing a list of column names, or a pattern using the IRAF pattern matching syntax (type `help system.match` for a description of the IRAF pattern matching syntax). If you need a list of the column names, you can run the `tlcol` task (type `tlcol infile.fits`).

Rows are selected according to a *filter*. The filter is evaluated at each table row, and the row is selected if the filter is true. For example, if you specify:

```
infile.fits[3][c:WAVELENGTH,FLUX][r:SPORDER=(68:70)]
```

IRAF will extract data from the table stored in the third extension of the FITS file, `infile.fits`, specifically the data from the columns labelled `WAVELENGTH` and `FLUX`, and will restrict the extraction to the rows where the spectral order (`SPORDER`) is within the range 68–70, inclusive. Alternatively, if you specify:

```
infile.fits[sci,2][c:FLUX][r:row=(20:30)]
```

IRAF will obtain data from the table stored in the FITS file extension with an `EXTNAME` of `SCI` and `EXTVER` of 2. The data will come from the column `FLUX` and be restricted to the row numbers 20–30, inclusive. All `STSDAS` and `TABLES` tasks are now able to use row and column selection. For a complete explanation of the table selector syntax, type `help selectors`.

2.2 GEIS File Format

The HST-specific Generic Edited Information Set (GEIS) format¹ is the standard format for reducing data from FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2. All HST images in GEIS format consist of two components: a *header file* and a separate *binary data file*, both of which should reside in the same directory. GEIS header files, whose suffixes end in “h” (e.g., `w01o0105t.c1h`), consist entirely of ASCII text in fixed-length records of 80 bytes. These records contain header keywords that specify the properties of the image itself and the parameters used in executing the observation and processing the data. GEIS binary data files, whose suffixes end in “d” (e.g., `w01o0105t.c1d`), contain one or more *groups* of binary data. Each group comprises a data array followed by an

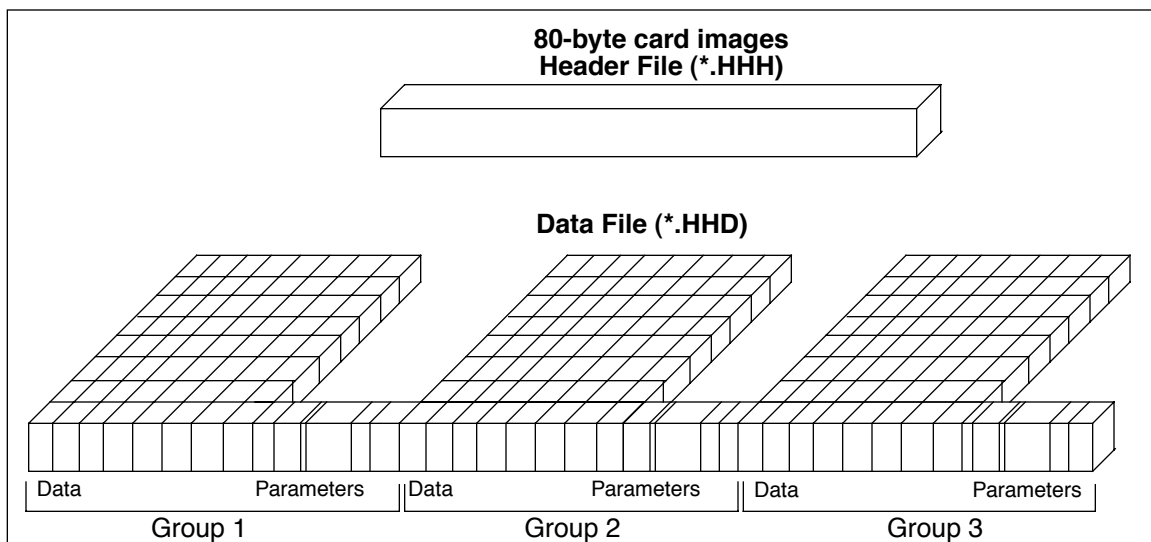
1. GEIS files are also commonly referred to as STSDAS images.

associated block of binary parameters called the Group Parameter Block (GPB). The sizes and datatypes of the data arrays and group parameters in each group of a GEIS file are identical. Figure 2.2 depicts the structure of a GEIS data file graphically.



The binary content of GEIS files is machine dependent. Copying GEIS files directly from one platform to another (e.g., from a VAX to a Sun) may result in unreadable data.

Figure 2.2: GEIS File Structure



2.2.1 Converting FITS to GEIS

The STScI archive stores and distributes datasets from FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2 in a special archival FITS format.



We highly recommend that users convert these datasets back into their native GEIS format before working with them.

Your data must be in GEIS format for you to use many of the STSDAS software tools developed specifically for analysis of these data. It is important to use the **strfits** task found in **stsdas.fitsio** or in **tables.fitsio** to perform the conversion from archival FITS format to the GEIS format because the data-processing pipeline employs a special convention for

mapping GEIS files to FITS format. While other FITS readers may be able to read portions of the data correctly, they are unlikely to reconstruct the entire data file properly.

To recreate the original multigroup GEIS file using **strfits**, you must first type:

```
cl> set imtype=hhh
```

This command tells IRAF to write output files in GEIS format. You then need to set the **strfits** parameters `xdimtogf` and `oldirafname` both to “yes”. For example, after you have set `imtype = hhh`, you can convert the FITS file `*_hhf.fits` into the GEIS format files `*.hhh` and `*.hhd` by typing:

```
cl> strfits *_hhf.fits "" xdim=yes oldiraf=yes
```

In particular, for WFPC2 dataset `u6n20101m_clf.fits`, **strfits** produces `u6n20101m.clh` and `.cld`.

2.2.2 GEIS Data Groups

One of the original advantages of GEIS format was that it could accommodate multiple images within a single file. This feature is useful because a single HST observation often produces multiple images or spectra. For example, a single WF/PC-1 or WFPC2 exposure generates four simultaneous images, one for each CCD chip. Likewise, the FOS and GHRS obtain data in a time-resolved fashion so that a single FOS or GHRS dataset comprises many spectra—one corresponding to each readout. The data corresponding to each sub-image (for the WF/PC-1 or WFPC2) or each sub-integration (for the FOS or GHRS) are stored sequentially in the groups of a single GEIS binary data file. The header file corresponding to this data file contains the information that applies to the observation as a whole (i.e., to all the groups in the image), and the group-specific keyword information is stored in the group parameter block of each data group in the binary data file.

The *number* of groups produced by a given observation depends upon the instrument configuration, the observing mode, and the observing parameters. Table 2.2 lists the *contents* and the number of groups in the final calibrated image for the most commonly-used modes of each instrument which uses the GEIS data format.

Table 2.2: Groups in Calibrated Images, by Instrument and Mode

Instrument	Mode	Number of Groups	Description
FGS	All	7	FGS data are not reduced with IRAF and STSDAS. Therefore, FGS groups have different meaning than for the other instruments.
FOC	All	1	All FOC images have only a single group.
FOS	ACCUM	n	Group n contains accumulated counts from groups (subintegrations) 1, 2, ... n . The last group is the full exposure.
	RAPID	n	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE.
HSP	All	1	HSP datasets always have only a single group that represents either digital star (.d0h, .c0h), digital sky (.d1h, .c1h), analog star (.d2h, .c2h), or analog sky (.d3h, .c3h).
GHRS	ACCUM	n	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE. If FP-SPLIT mode was used, the groups will be shifted in wavelength space. The independent subintegrations should be coadded prior to analysis.
	RAPID	n	Each group is a separate subintegration with exposure time given by group parameter EXPOSURE.
WF/PC-1	WF	4	Group n represents CCD chip n , e.g., group 1 is chip 1 (unless not all chips were used). Group parameter DETECTOR always gives chip used.
	PC	4	Group n is chip $n + 4$, e.g., group 1 is chip 5. If not all chips were used, see the DETECTOR parameter which always gives the chip used.
WFPC2	All	4	Planetary chip is group 1, detector 1. Wide Field chips are groups 2–4 for detectors 2–4. If not all chips were used, see the DETECTOR keyword.

2.2.3 Working with GEIS Files

This section briefly explains how to work with information in GEIS header and data files.

GEIS Headers

Header keyword information relevant to each group of a GEIS file resides in two places, the header file itself and the parameter block associated with the group. Because GEIS header files are composed solely of ASCII text, they are easy to print using standard Unix or VMS text-handling facilities. However, the group parameters are stored in the

binary data file. To access them you need to use a task such as **imheader**, as shown in "Printing Header Information" (this section, below).

You can use the IRAF **hedit** task to edit the keywords in GEIS headers. While it is possible to edit GEIS header files using standard Unix and VMS text editors, you must maintain their standard 80-character line length. The **hedit** task automatically preserves this line length. If you need to add or delete group parameters, you can use the STSDAS **groupmod** task in the **stsdas.hst_calib.ctools** package. The STSDAS **chcalpar** task, described in more detail in the Calibration chapters for each instrument's data handbook, is useful for updating header keywords containing calibration switches and calibration reference files.



*Always edit headers using tasks like **hedit**, **eheader**, and **chcalpar**. Editing headers with a standard text editor may corrupt the files by creating incorrect line lengths.*

GEIS Data Files

Numerous IRAF/STSDAS tasks exist for working with GEIS images (see Chapter 3 of the HST Introduction). Most of these tasks operate on only one image at a time, so you usually need to specify which group of a GEIS file is to be processed. If you do not specify a group, your task will choose the first group by default.

Specifying a Group

To specify a particular group in a GEIS file, append the desired group number in square brackets to the file name (e.g., `z2bd010ft.d0h[10]`). For example, to apply the **imarith** task to group 10 of a GEIS image, type the following (always refer to a GEIS file by its header file name, i.e. `*.??h`, even though mathematically you are operating on the data portion):

```
cl> imarith indata.hhh[10] + 77.0 outdata.hhh
```

This command will add 77.0 to the data in group 10 of the file `indata.hhh`, and will write the output to a new single-group file called `outdata.hhh`. Any operation performed on a single group of a multigroup GEIS file results in an output file containing a single group.

Specifying an Image Section

If you wish to process only a portion of an image, you can specify the image section after the group specification in the following manner:

```
cl> imarith indata.hhh[2][100:199,200:399] * 32.0 outdata.hhh
```

This command extracts a 100 by 200 pixel subsection of the image in the second group of the file `indata.hhh`, multiplies this data by a factor of 32.0, and stores the result in a new output file, `outdata.hhh`, which is a 100 by 200 pixel single group GEIS file.

An image section of one group of a GEIS image may be overwritten or operated upon, leaving the rest of the image intact. For example, the following 2 lines will first create `outdata.hhh` and then overwrite a section of it:

```
cl> imarith indata.hhh * 0.0 outdata.hhh
```

```
cl> imarith indata.hhh[2][100:199,200:399] * 32.0 \  
>>> outdata.hhh[100:199,200:399]
```

Printing Header Information

As discussed in the previous section, the task **imheader** extracts and prints information about the GEIS image. This task reports the image name, dimensions (including the number of groups), pixel type, and title of the image when it is run in default mode. For example:

```
cl> imhead indata.hhh  
indata.hhh[1/64][500][real]: INDATA[1/64]
```

The output line indicates that `indata.hhh` is a multigroup GEIS file which contains 64 groups of images, each consisting of an array 500 pixels in length. The data type of the values is real (floating point). Note that since no group designation was provided, the task defaulted to the first group. To reveal more information regarding group 10, you can type:

```
cl> imhead indata.hhh[10] long+ | page
```

which will generate a long listing of both the ASCII header parameters in the `*.hhh` file and the specific group parameters for group 10 from the `*.hhd` file.

Other Group-Related Tasks

Currently, IRAF and STSDAS tasks cannot process all the groups in an input image and write the results to corresponding groups in an output image. However, there are several STSDAS tasks, particularly in the **toolbox.imgtools** and **hst_calib.ctools** packages, that simplify working with group format data. Please refer to Chapter 3 and the *STSDAS User's Guide* for more details about working with GEIS images.

2.2.4 The “waiver” FITS Format

Although “waiver” is not an accurate word for the intended purpose, for historical reasons it has stuck and will be reluctantly adopted. However, in the past, a grammatically incorrect word “waivered” was used.

The “waiver” FITS format was developed when the HST archive needed a format to store and distribute the data products in a machine-independent medium for the community, at a time before FITS image extension was standardized. As a result, the “waiver” FITS format was adopted as a compromise.

Since, at the time, FITS could only have a single image while the HST data (in GEIS format) could have several images as multiple groups in one file, the idea of stacking the images of different groups together as a new dimension in the FITS image was conceived. As for group parameters, they are put in an ASCII table and the table becomes the first (and only) extension of the FITS file.

For example, the WFPC2 pipeline generates the science data as a GEIS file of 4 groups, each is an 800x800 image corresponding to one of the 4 detectors. When this GEIS file is converted to the “waiver” FITS file, the FITS file has an image of 800x800x4 (a three-dimensional image!) at its primary HDU. Similarly, an FOS GEIS file may have 40 groups, each group is a 1-D image (spectrum) of the size 2064. The waiver FITS file then will have one 2-D image of the size 2064x40, at its primary HDU. In the case of WFPC2, the first extension of the waiver FITS file will be an ASCII table containing 4 rows; each row corresponds to a group. The value of each group parameter is under a column named after the group parameter, e.g. the value of the group parameter CRVAL1 of the 2nd group will be at the 2nd row, under the column named “CRVAL1”. In other words, the ASCII table has as many rows as there are groups in the original GEIS file, and as many columns as group parameters.

Although, *in theory, certain* IRAF/STSDAS tasks can directly access the data in the “waiver” FITS file, e.g. to display the 2nd “group” of a WFPC2 image:

```
st> display u67m0206r_c0f.fits[0][*,*,2]
```

will work, *most tasks, especially those specific to HST instruments, will not.*



It is therefore STRONGLY recommended that all waiver FITS files be converted back to the GEIS format, by using the task `strfits`, before further processing and analysis with IRAF/STSDAS tasks.

Analyzing HST Data

In this chapter. . .

3.1 Alternative Means of Accessing HST Data / 3-2

3.2 Navigating STSDAS / 3-4

3.3 Displaying HST Images / 3-7

3.4 Analyzing HST Images / 3-11

3.5 Displaying HST Spectra / 3-23

3.6 Analyzing HST Spectra / 3-27

3.7 References / 3-40

The Space Telescope Science Data Analysis System (STSDAS) is the software system for calibrating and analyzing data from the Hubble Space Telescope. The package contains programs—called *tasks*—that perform a wide range of functions supporting the entire data analysis process, from reading tapes, through reduction and analysis, to producing final plots and images. This chapter introduces the basics of STSDAS, showing you how to display your data, leading you through some simple data manipulations, and pointing you towards more sophisticated tasks, some of which are described in the instrument sections of the data handbooks.

STSDAS is layered on top of the [Image Reduction and Analysis Facility \(IRAF\)](#) software developed at the National Optical Astronomy Observatory (NOAO). Any task in IRAF can be used in STSDAS, and the software is portable across a number of platforms and operating systems. To exploit the power of STSDAS, you need to know the basics of IRAF. If you are not already familiar with IRAF, consult the IRAF Primer in Appendix A before reading further.

3.1 Alternative Means of Accessing HST Data

There is much software available to access HST data. This section will mention a few of the more popular alternatives to IRAF.

3.1.1 Interactive Data Language (IDL)

IDL is very popular in the astronomical community. In short it is an array-based, interactive programming language that is strong in numerical analysis and visualization tools. It is typically much easier to develop new analysis and visualization applications and utilities in IDL than in Fortran or C. As a result many astronomers use it for doing analysis of HST data. It is moderately expensive, however, and can be obtained from Research Systems Inc. (www.rsinc.com/idl/). Libraries for reading HST data are part of the freely available ASTRON library (idlastro.gsfc.nasa.gov) which also has links to other astronomy IDL libraries.

3.1.2 Python

Python is rapidly becoming a popular language for astronomical data reduction applications. It is a freely available, general-purpose, dynamically-typed interactive language that provides modules for scientific programming. These include:

- numarray: provides IDL-style array manipulation facilities
- PyFITS: ability to read and write FITS files to and from arrays
- matplotlib: plotting and image display package
- numdisplay: package to display arrays to saomage, ds9 and ximtool
- PyRAF: package to run IRAF tasks from Python

Python is a very powerful language that is well suited to writing programs to solve many needs beside scientific analysis. It is generally much more productive than Fortran, C, or Java. Tools are available to read (but currently not write) GEIS files. PyRAF allows easy use of IRAF tasks with code written in Python.

STScI is developing most of its new calibration and data analysis software in Python. More information on the use of Python to analyze HST data can be obtained from:

http://www.stsci.edu/resources/software_hardware/pyraf

http://www.stsci.edu/resources/software_hardware/numarray

http://www.stsci.edu/resources/software_hardware/pyfits

3.1.3 Fortran and C

For those that wish to write their own Fortran or C applications, we recommend use of the FITSIO library for reading FITS files (<http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html>; note that the C library is called CFITSIO).

This library does not support the GEIS format directly so users will need to use the waiver FITS form obtained from the archive and manually extract the needed information.

3.1.4 Java

The most widely used FITS libraries for Java are the Java FITS Utilities (<http://heasarc.gsfc.nasa.gov/docs/heasarc/fits/java/v0.9/>) and the Java FITS Class Library (http://www.eso.org/~pgrosbol/fits_java/jfits.html). Like FITSIO, neither can read GEIS files, however; one must read the waiver FITS file of these data sets.

3.1.5 PyRAF

PyRAF is a new command language for IRAF that is based on Python. It has a number of advantages over the IRAF CL. Most importantly, with few exceptions, it allows use of exactly the same syntax that the IRAF CL accepts. Some of the advantages that it provides are:

- true command line recall (with arrow key editing)
- command and filename completion
- GUI-based graphics windows, previous plot recall, multiple graphics windows
- GUI epar editor with display of help in separate window
- IDL-like capabilities
- true error handling for scripts (shows which line the script fails at when errors occur)
- can script IRAF tasks in Python language
- exception handling capability

Since PyRAF is so highly compatible with the IRAF CL, virtually all of the examples shown in this handbook will work the same for PyRAF. Minor differences include the user prompt and the graphics windows appearance.

More information on PyRAF can be found at:

http://www.stsci.edu/resources/software_hardware/pyraf

3.2 Navigating STSDAS

The tasks in STSDAS are far too numerous and complicated to describe comprehensively in this volume. Instead, we will show you where to find the STSDAS tasks appropriate for handling certain jobs. You can refer to online help or the *STSDAS User's Guide* for details on how to use these tasks. Some useful online help commands are:

- `apropos word` - searches the online help database for tasks relating to the specified word (see Figure A.4).
- `help task` - provides detailed descriptions and examples of each task.
- `help package` - lists the tasks in a given package and their functions.
- `describe task` - provides a detailed description of each task.
- `example task` - provides examples of each task.

3.2.1 STSDAS Structure

STSDAS is structured so that related tasks are grouped together as packages. For example, tasks used in the calibration process can be found in the **hst_calib** package, and tasks used for image display and plotting can be found in the **graphics** package. Table 3.1 shows the STSDAS package structure. Note that IRAF version 2.11 must be installed on your system in order for you to use STSDAS 2.0 and TABLES version 2.0 or higher.

3.2.2 Packages of General Interest

Images

Both IRAF and STSDAS contain a large number of tasks that work with HST images. Some of the packages you should investigate are:

- **images**: This package includes general tasks for copying (**imcopy**), moving (**imrename**), and deleting (**imdelete**) image files. These tasks operate on both the header and data portions of the image. The package also contains a number of general purpose tasks for operations such as rotating and magnifying images.
- **stdas.toolbox.imgtools**: This package contains general tools for working with multigroup GEIS images, including tasks for working with masks, and general purpose tasks for working with the pixel data, such as an interactive pixel editor (**pixedit**), and **gcombine** for coadding GEIS images.

Table 3.1: STSDAS Version 3.2 Package Structure

analysis	Data analysis package.
dither	Dithered image combination.
fitting	Curve fitting tools.
fourier	Fourier analysis.
gasp	Access the HST Guide Star Catalog on CD-ROM.
isophote	Elliptical isophote image analysis.
nebular	Tasks for analyzing nebular emission lines
restore	Deconvolve or filter 1- or 2-dimensional images.
statistics	Statistical analysis software.
contrib	User-contributed software.
redshift	Tasks for determining redshifts and dispersions.
spfitpkg	Fitting spectra with non-linear chi-square minimization.
vla	Spectral image reduction for VLA data.
fitsio	FITS input/output for Space Telescope data (images and tables).
graphics	Graphics and image display package.
sdisplay	Image display package for SAOImage display device.
stplot	General plotting utilities.
hst_calib	HST Science Instrument calibration package.
acs	Tasks for calibrating ACS data.
ctools	General calibration tools.
foc	Tasks for calibrating FOC data.
focprism	FOC prism package.
fos	Tasks for calibrating FOS data.
spec_polar	Tasks for reducing and analyzing FOS polarimetry.
hrs	Tasks for calibrating HRS data.
nicmos	Tasks for calibrating NICMOS data.
paperprod	Tasks for generating paper products.
stis	Tasks for calibrating STIS data.
synphot	Synthetic photometry and modelling instrument response.
simulators	Synthetic photometry simulation package.
wfpc	Tasks for calibrating WFPC and WFPC-2 data.
w_calib	Tasks for deriving the WF/PC instrument calibration.
playpen	Miscellaneous experimental tasks.
sobsolete	Package of tasks that have been retired.
foccs	FOC calibration software package.
focgeom	FOC geometry package.
focphot	FOC photometry package.
focutility	Obsolete FOC utility package.
hsp	Tasks for calibrating HSP data.

olddither	Older version (1.2) of dither.
registration	Compute registration parameters and resample unaligned data files.
testdata	Tools for creating artificial images.
timeseries	Time series photometry data reduction and analysis.
y_calib	Tasks supporting the FOS calibration process.
z_calib	Tasks supporting the HRS calibration process.
toolbox	General tools package.
convfile	Reformat images between VAX and Sun.
headers	Tools for modifying image headers.
imgtools	Tools for manipulating & examining images and bad pixel lists.
mstools	Tasks to handle STIS/NICMOS IMSETs.
tools	Generic data handling and utility tools.
ttools	Table manipulation tools.

- **stdas.toolbox.imgtools.mstools**: This package contains tools for working with FITS image extensions, in particular NICMOS, STIS, and ACS image sets (imsets). **Msstatistics**, for example, will print statistics on these images and **msarith** can do image arithmetic.
- **stdas.analysis**: This package contains general tasks for image analysis, such as Fourier analysis (**fourier**) and dithering (**dither**).

Tables

Several of the analysis packages in STSDAS, including calibration pipeline tasks, create output files in STSDAS table format (which is a binary row-column format) or in FITS binary table format. (ASCII-format tables are supported, but for input only.) The *STSDAS User's Guide* describes the STSDAS table format in detail. Tasks in the **ttools** package or in the external **tables** package can be used to read, edit, create, and manipulate tables. For example:

- **tread** displays a table, allowing you to move through it with the arrow keys.
- **tprint** displays a table.
- **tcopy** copies tables.
- **tedit** allows you to edit a table.

Many other tasks in **ttools** perform a variety of other functions. See the online help for details.

3.3 Displaying HST Images

This section will be of interest primarily to observers whose datasets contain two-dimensional images, as it explains:

- How to display images in IRAF using the **display** task.
- How to display subsections of images.

Observers viewing WF/PC-1 and WFPC2 data may wish to remove cosmic rays before displaying their data. The FOC photon-counting hardware does not detect cosmic rays as easily as CCDs; the NICMOS pipeline automatically removes cosmic rays from MULTIACCUM observations; and the STIS and ACS pipelines automatically remove cosmic rays from CR-SPLIT association products.

3.3.1 The Display Task

The most general IRAF task for displaying image data is the **display** task, the best choice for a first look at HST imaging data. To display an image, you need to:

1. Start an image display server, such as SAOimage, in a separate window from your IRAF session, either from a different xterm window or as a background job before starting IRAF. To start SAOimage, type the following in a Unix window:

```
> saimage &
```

2. Set the size of the window to 1024x1024 by typing:

```
cl> set stdimage = imt1024
```

3. Load the **images.tv** package from the window where you're running IRAF:

```
cl> images
im> tv
```



Several different display servers, including SAOimage, ds9 (the next generation of SAOimage), and Ximtool, can be used with IRAF. ds9 may be retrieved from <http://hea-www.harvard.edu/RD/ds9/>. Ximtool may be retrieved from <ftp://iraf.noao.edu/iraf/x11iraf/>.

4. Display the image with the IRAF **display** task, using the syntax appropriate for the file format (Chapter 2 explains how to specify GEIS groups and FITS extensions):

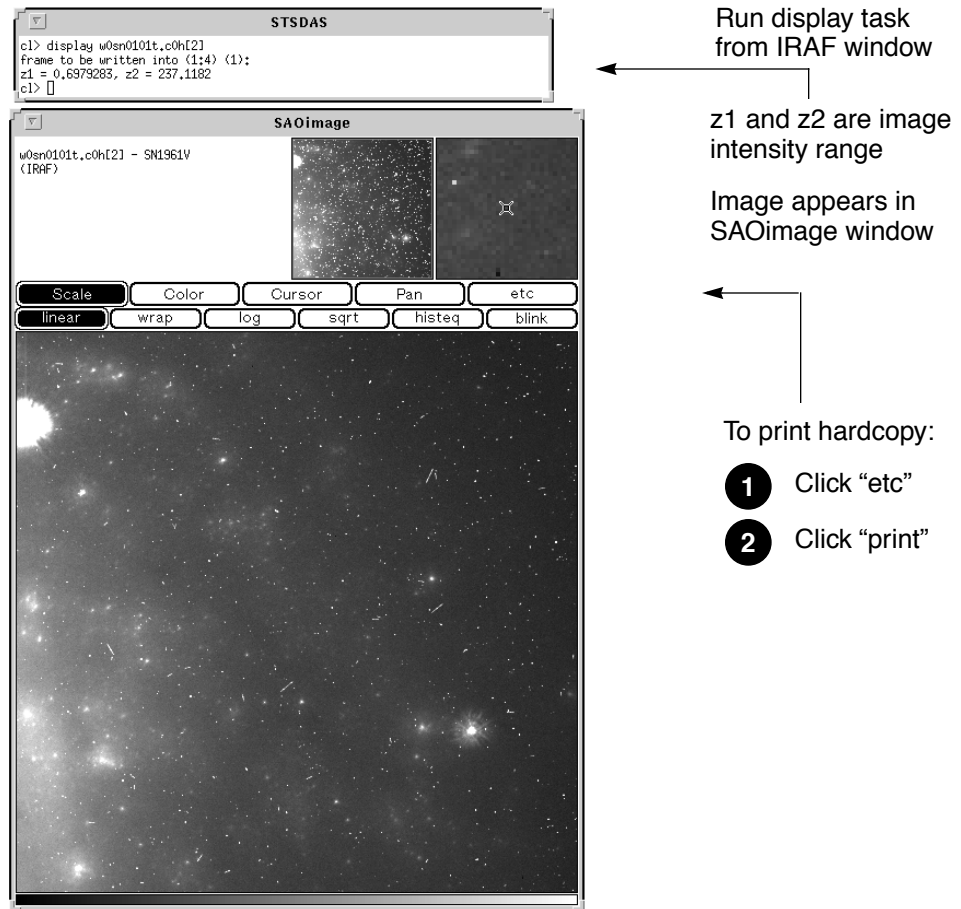
```
tv> display fname.c0h[2] 1      (GEIS group 2)
tv> display fname.fits[11] 1    (FITS extension 11)
tv> display fname.fits[sci,3] 1 (FITS extension sci,3)
```

Note that when using **display** or any other task on GEIS images, you do not need to specify a group; the first group is the default. However, when working with FITS files you must specify an extension, unless the FITS file contains only a single image in the primary data unit and has no extensions. Figure 3.1 shows how to display group two of a WF/PC-1 image.



*If you want to display all four chips of a WF/PC-1 or WFPC2 image simultaneously, you can create a mosaic with the STSDAS **wmosaic** task in the `hst_calib.wfpc` package. Type `help wmosaic` for details.*

Figure 3.1: Displaying an Image



Modifying the Display

There are two ways to adjust how your image is displayed:

- Use the SAOimage command buttons that control zooming, panning, etc.
- Reset the **display** task parameters.

Once an image appears in your SAOimage window, you can use the SAOimage commands displayed near the top of the image window to manipulate or print your image. The *SAOimage Users Guide* describes these commands, although most are fairly intuitive. Just click on the buttons to scale, pan, or print the image, or to perform other commonly-used functions. On-line help is also available at the system level: type `man saimage` in Unix or `help saimage` in VMS.

The example in Figure 3.1 shows how you should display an image for a first look. By default, **display** automatically scales the image intensity using a sampling of pixels throughout the image. During your first look, you may want to experiment with the scaling using the `zscale`, `zrange`, `z1` and `z2` parameters. The `zscale` parameter toggles the autoscaling.

Setting `zscale-` and `zrange+` tells the task to use minimum and maximum values from the image as the minimum and maximum intensity values. To customize your minimum and maximum intensity display values, set `zscale-`, `zrange-`, `z1` to the minimum value and `z2` to the maximum value that you want displayed. For example:

```
im> disp w0mw0507v.c0h 1 zrange- zscale- z1=2.78 z2=15.27
```

Notice in Figure 3.1 that when you run **display**, the task shows you the `z1` and `z2` values that it calculates. You can use these starting points in estimating reasonable values for the minimum and maximum intensity display parameters.¹

If you want to display an image with greater dynamic range, you may prefer to use logarithmic scaling. However, the log scaling function in `SAOimage` divides the selected intensity range into 200 linearly spaced levels before taking the log. The resulting intensity levels are rendered in a linear rather than logarithmic sense. You can often obtain better results if you create a separate logarithmic image to display. One way to create a logarithmic image is with the **imcalc** task:

```
im> imcalc x2ce0502t.c1h x2ce0502t.hhh "log10(im1+1.0)"
```

If the peak pixel in your original image contained 2000 counts, for example, you would then display the logarithmic image with `z1=0` and `z2=3.3`.

Otherwise, the user can simply do:

```
im> display x2ce0502t.c1h ztrans=log
```

The image display buffer can also be adjusted in IRAF by setting the `stdimage` parameter. For example,

```
im> set stdimage = imt 2048
```

will allow a larger image to be displayed without losing the borders.

1. Type `help display` within IRAF to obtain more information about these parameters.

3.3.2 Working with Image Sections

Sometimes you may want to display only a portion of an image, using the syntax for specifying image sections discussed in Chapter 2. Your specified pixel range should give the starting point and ending point, with a colon separating the two. List the horizontal (x -axis) range first, followed by the vertical (y -axis) range. For example, to specify a pixel range from 101 to 200 in the x -direction and all pixels in the y -direction from group three of a GEIS format image:

```
tv> display image.hhh[3][101:200,*] 1
```

To specify the same pixel range in the second SCI extension of a NICMOS FITS image:

```
tv> display image.fits[sci,2][101:200,*] 1
```



If you specify both a group and an image section of a GEIS file, the group number must come first. When displaying sections of FITS image extensions, you must specify the extension, which also comes before the image section.

Figure 3.2 shows examples of displaying an image and an image section.

3.4 Analyzing HST Images

This section describes methods for using STSDAS and IRAF to work with two-dimensional image data from HST. Subjects include:

- Relating your image to sky coordinates.
- Examining and manipulating your image.
- Working with NICMOS, STIS, and ACS imsets.
- Converting counts to fluxes.

Figure 3.2: Displaying Sections and Groups of an Image

1 Display group 2 of entire image

```

STSDAS
st> display w0mw0502t.c0h[2]
frame to be written into (1:4) (1):
z1 = -13,14832, z2 = 13,61855
st>

```

2 Display only a section of group 2 of the image

```

STSDAS
st> display w0mw0502t.c0h[2][50:450,50:450]
frame to be written into (1:4) (1):
z1 = -5,630896, z2 = 17,13277
st>

```

SAOImage

```

w0mw0502t.c0h[2] - W0M0502T[2/4]
(IRAF)

```

270.0 649.0 -0.3062

Scale Color Cursor Pan etc
linear wrap log sqrt histeq blink

SAOImage

```

w0mw0502t.c0h[2] - W0M0502T[2/4]
(IRAF)

```

-7.0 474.0 x

Scale Color Cursor Pan etc
linear wrap log sqrt histeq blink

3.4.1 Basic Astrometry

This section describes how to determine the orientation of an HST image and the RA and Dec of any pixel or source within it, including:

- Tasks that supply positional information about HST images.
- Methods for improving your absolute astrometric accuracy.

Positional Information

The header of every calibrated HST two-dimensional image contains a linear astrometric plate solution, written in terms of the standard FITS astrometry header keywords: reference pixel values (CRPIX1, CRPIX2, CRVAL1, CRVAL2), and the CD matrix (CD1_1, CD1_2, CD2_1, and CD2_2). IRAF/STSDAS tasks can use this information to convert between

pixel coordinates and RA and Dec. Two simple tasks that draw on these keywords to relate your image to sky coordinates are:

- **disconlab**: Displays your image with a superimposed RA and Dec grid. Simply open an SAOimage window and type, for example:

```
sd> disconlab n3tc01a5r_cal.fits[1]
```

- **xy2rd**: Translates x and y pixel coordinates to RA and Dec. (The task **rd2xy** inverts this operation.) SAOimage displays the current x,y pixel location of the cursor in the upper-left corner of the window. To find the RA and Dec of the current pixel, you supply these coordinates to **xy2rd** by typing

```
sd> xy2rd n3tc01a5r_cal.fits[1] hms+ x y
```

Note, the `hms` option formats the results in hours, min, and seconds.

Observers should be aware that these tasks do not correct for geometric distortion. Only FOC, STIS, and ACS images currently undergo geometric correction during standard pipeline processing. (For WFPC2, a pixel area map correction image (f1k1552bu.r9h) may be obtained from the archive.) If precise relative astrometry is needed, an instrument-specific task that accounts for image distortion, such as the **metric** task for WF/PC-1 and WFPC2 images, should be used.



Do not use tasks like `rimcursor` or `xy2rd` directly on WF/PC-1 or WFPC2 images if you require accurate relative positions. WF/PC-1 and WFPC2 pipelines do not correct for geometric distortions which will affect the accuracy of relative positions. Both `wmosaic` and `metric`, found in the `stdas.hst_calib.wfpc` package, correct for this distortion.

Table 3.2 lists some additional tasks that draw on the standard astrometry keywords.

Table 3.2: Additional IRAF and STSDAS Astrometry Tasks

Task	Purpose
compass	Plot north and east arrows on an image.
north	Display the orientation of an image based on keywords.
rimcursor	Determine RA and Dec of a pixel in an image.
wcscoords	Use WCS ¹ to convert between IRAF coordinate systems.
weslab	Produce sky projection grids for images.

1. World Coordinate System (WCS). Type `help specwcs` at the IRAF prompt for details.

Improving Astrometric Accuracy

Differential astrometry (measuring a position of one object relative to another in an image) is easy and relatively accurate for HST images. Absolute astrometry, on the other hand, is more difficult, owing to uncertainties in the locations of the instrument apertures relative to the Optical Telescope Assembly (OTA or V1) axis and the inherent uncertainty in Guide Star positions. However, if you can determine an accurate position for any single star in your HST image, then your absolute astrometric accuracy will be limited only by the accuracy with which you know that star's location and the image orientation.

If there is a star on your image suitable for astrometry, you may wish to extract an image of the sky around this star from the Digitized Sky Survey and measure the position of that star using, for example, the GASP software (described in the *STSDAS User's Guide*). These tools provide an absolute positional accuracy of approximately 0.7 arcsec. Contact the Help Desk for assistance (send E-mail to help@stsci.edu).

3.4.2 Examining and Manipulating Image Data

This section describes **implot** and **imexamine**, two basic IRAF tools for studying the characteristics of an image, and Table 3.4 lists some useful IRAF/STSDAS tasks for manipulating images.

implot

The IRAF **implot** task (in the **plot** package) allows you to examine an image interactively by plotting data along a given *line* (*x*-axis) or *column* (*y*-axis). When you run the task, a large number of commands are available in addition to the usual cursor mode commands common to most IRAF plotting tasks. A complete listing of commands is found in the on-line help, but the most commonly used are listed in Table 3.3. Figure 3.3 shows an example of how to use the **implot** task.

Table 3.3: Basic implot Commands








Keystroke	Command
	Display on-line help.
	Plot a line.
	Plot a column.
	Move down.
	Move up.
	Display coordinates and pixel values.
	Quit implot.

Figure 3.3: Plotting Image Data with implot

STSDAS

```

pl> ?
  calcomp      gkdir      indkern      phistogram   sgidecode    surface
  contour      gkiextract  implot       pradprof     sgikern      velvect
  crtpict      gkimosaic   nspkern     prow         showcap      stdgraph
  gdevices     graph       pcol        prows        stdgraph
  gkidecode    hafton     pcols       pvector

```


```

pl> dir w*
w0mw0502t.c0d  w0mw0502t.c0h  w0mw0502t.d0h
pl> implot w0mw0502t.c0h 200

```

Plot line 200 of a WF/PC-1 image

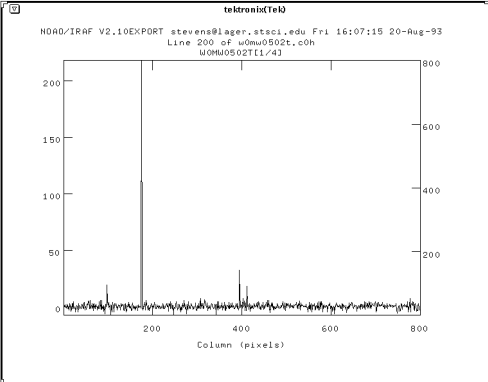
To Print This Plot:

1 Press 

2 Type `:.gflush` to flush the buffer

teleornb(Tel)

NOR0/IRAF V2.10EXPORT stevens@liger.stsci.edu Fri 16:07:15 20-Aug-93
Line 200 of w0mw0502t.c0h
W0MW0502T11/43



imexamine

The IRAF **imexamine** task (in the **images.tv** package) is a powerful task that integrates image display with various types of plotting capabilities. Commands can be passed to the task using the image display cursor and the graphics cursor. A complete description of the task and its usage are provided in the online help, available from within the IRAF environment by typing `help imexamine`.

3.4.3 Working with STIS, ACS, and NICMOS Imsets

NICMOS, STIS, and ACS data files contain groups of images, called *imsets*, associated with each individual exposure. A STIS or ACS imset comprises SCI, ERR, and DQ images, which hold science, error, and data quality information. A NICMOS imset, in addition to SCI, ERR, and DQ images, also contains TIME and SAMP images recording the integration time and number of samples corresponding to each pixel of the SCI image. See the NICMOS, STIS, and ACS Data Structures chapters for more details on imsets.

Table 3.4: Image Manipulation Tasks

Task	Package	Purpose
boxcar	images.imfilter	Boxcar smooth a list of images
blkavg	images.imgeom	Block average or sum an image
gcombine	stdas.toolbox.imgtools	Combine GEIS images using various algorithms and rejection schemes ¹
gcopy	stdas.toolbox.imgtools	Copy multigroup GEIS images ¹
geomap	images.immatch	Compute a coordinate transformation
geotran	images.immatch	Resample an image based on geomap output
grlist	stdas.graphics.stplot	List of file names of all groups of a GEIS image (to make @lists) ¹
gstatistics	stdas.toolbox.imgtools	Compute image statistics ¹
imcalc	stdas.toolbox.imgtools	Perform general arithmetic on images
imedit	images.tv	Fill in regions of an image by background estimation or copy and paste
imexamine	images.tv	Examine images using display, plots, and text (see page 3-15)
implot	plot	Plot lines and columns of images (see page 3-14)
magnify	images.imgeom	Magnify an image
msarith	stdas.toolbox.mstools	Performs basic arithmetic on NICMOS and STIS imsets ²
mscombine	stdas.toolbox.mstools	Extension of gcombine for NICMOS, STIS, and ACS imsets ¹
msstatistics	stdas.toolbox.mstools	Extension of gstatistics for NICMOS and STIS imsets ^{1, 2}
newcont	stdas.graphics.stplot	Draw contours of two-dimensional data
pixcoord	stdas.hst_calib.wfpc	Compute pixel coordinates of stars in a GEIS image ¹
plcreate	xray.ximages	Create a pixel list from a region file (e.g., from SAOimage regions)
rotate	images.imgeom	Rotate an image
saodump	stdas.graphics.sdisplay	Make image and colormap files from SAOimage display
siaper	stdas.graphics.stplot	Plot science instrument apertures of HST

1. Will process all groups of a multigroup GEIS file. An OIF (imh/pix) file is like a GEIS file with one group.

2. Task is not yet functional with ACS data.

Here we describe several STSDAS tasks, located in the `stsdas.toolbox.imgtools.mstools` package, that have been designed to work with imsets as units and to deconstruct and rebuild them.

msarith

This tool is an extension of the IRAF task **imarith** to include error and data quality propagation. The **msarith** task supports the four basic arithmetic operations (+, -, *, /) and can operate on individual or multiple imsets. The input operands can be either files or numerical constants; the latter can have associated errors, which will propagate into the error array(s) of the output file.

mscombine

This task runs the STSDAS task **gcombine** on NICMOS, STIS, and ACS data files. It divides each imset into its basic components (SCI, ERR, and DQ, plus SAMP and TIME for NICMOS). The SCI extensions then become the inputs for the underlying **gcombine** task, and the ERR extensions become the error maps. The DQ extensions are first combined with a user-specified Boolean mask allowing selective pixel masking and are then combined into the data quality maps. If scaling by exposure time is requested, the exposure times of each imset are read from the header keyword PIXVALUE in the TIME extensions.

Once **gcombine** has finished, **mscombine** then reassembles the individual output images into imsets and outputs them as one NICMOS, STIS, or ACS data file. The output images and error maps from **gcombine** form the SCI and ERR extensions of the output imset. The DQ extension will be a combination of the masking operations and the rejection algorithms executed in **gcombine**. For NICMOS, the TIME extension will be the sum of the TIME values from the input files minus the rejected values, divided on a pixel-by-pixel basis by the number of valid pixels in the output image. The final TIME array will be consistent with the output SCI image (average or median of the science data). The SAMP extension for NICMOS is built from all the input SAMP values, minus the values discarded by masking or rejection.

msstatistics

This tool is an extension of **gstatistics** in the STSDAS package, which is in turn an extension of **imstatistics**. The main novelty is the inclusion of the error and data quality information included with NICMOS and STIS images in computing statistical quantities. This task is not yet functional with ACS data.

In addition to the standard statistical quantities (min, max, sum, mean, standard deviation, median, mode, skewness, kurtosis), two additional quantities have been added to take advantage of the error information: the weighted mean and the weighted variance of the pixel distribution. If x_i is

the value at the i -th pixel, with associated error σ_i , the weighted mean and variance used in the task are:

$$\langle x \rangle_w = \frac{\sum_i \frac{x_i}{\sigma_i \times \sigma_i}}{\sum_i \frac{1}{\sigma_i \times \sigma_i}}$$

and:

$$\langle \sigma \rangle_w^2 = \frac{1}{\sum_i \frac{1}{\sigma_i \times \sigma_i}}$$

The data quality information carried by the NICMOS or STIS file is used to reject pixels in the statistical computation. Users can supply additional masks to reject objects or regions from the science arrays.

mssplit and msjoin

The **mssplit** task extracts user-specified imsets from a NICMOS, STIS, or ACS data file and copies them into separate files. Each output file contains a single imset along with the primary header of the original file. You might find this task useful for reducing the size of a NICMOS, STIS, or ACS file containing many imsets or for performing analysis on a specific imset. The **msjoin** task does the opposite of **mssplit**: it assembles separate imsets into a single data file.

There are additional tasks in this package for deleting and sorting imsets, as well as tasks for addressing a specific image class within an imset.

3.4.4 Photometry

Included in this section are:

- A list of IRAF/STSDAS tasks useful for determining source counts.
- Instructions on how to use header keyword information to convert HST counts to fluxes or magnitudes.
- A brief description of **synphot**, the STSDAS synthetic photometry package.

IRAF and STSDAS Photometry Tasks

The following are some useful IRAF/STSDAS packages and tasks for performing photometry on HST images:

- **apphot**: aperture photometry package.
- **daophot**: stellar photometry package useful for crowded fields.
- **isophote**: package for fitting elliptical isophotes.
- **imexamine**: performs simple photometry measurements.
- **imstat**: computes image pixel statistics.
- **imcnts**: sums counts over a specified region, subtracting background.
- **plcreate**: creates pixel masks.

Consult the online help for more details on these tasks and packages. The document “Photometry using IRAF” by Lisa A. Wells, provides a general guide to performing photometry with IRAF; it is available through the IRAF web page:

<http://iraf.noao.edu/docs/photom.html>



The apphot package allows you to measure fluxes within a series of concentric apertures. This technique can be used to determine the flux in the wings of the PSF, which is useful if you wish to estimate the flux of a saturated star by scaling the flux in the wings of the PSF to an unsaturated PSF.

Converting Counts to Flux or Magnitude

Calibrated HST images record signal in various units: WFPC2 and STIS imaging data are in DN (Data Numbers or counts); NICMOS data are in DN per second; ACS calibrated data are in electrons; and ACS drizzled data are in electrons per second. The pipeline calibration tasks do not alter these units in the images when performing the photometric correction step. Instead they calculate and write the sensitivity conversion factor (PHOTFLAM) and the ST magnitude scale zero point (PHOTZPT) into header keywords in the calibrated data. WF/PC-1 and WFPC2 observers should note that the four chips are calibrated individually, so these photometry keywords belong to the group parameters for each chip. For ACS observers, the PHOTFLAM values for the two WFC chips are defined to be the same.

PHOTFLAM is defined as the *mean* flux density F_λ in units of $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$ that produces 1 count per second in the HST observing mode (PHOTMODE) used for the observation. Note that the word ‘count’ may

refer to DNs or electrons, depending on the instrument used. For example, calibrated ACS images are already corrected for the instrumental gain, and the PHOTFLAM values are computed accordingly. The PHOTFLAM values for WFPC2, on the other hand, are dependent on the gain.

Calibrated images in units of counts may be converted to flux in units of $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$ by multiplying the image by the value of the PHOTFLAM header keyword and dividing by the value of the EXPTIME keyword (exposure time). Calibrated images in units of counts per second (i.e. NICMOS data in DN s^{-1} and drizzled ACS data in electrons s^{-1}), may simply be multiplied by the PHOTFLAM value to obtain the flux in units of $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$. NICMOS headers also contain the keyword PHOTFNU in units of Jy s. Multiplying NICMOS image by the PHOTFNU value will therefore yield fluxes in Janskys.

The STSDAS task **imcalc** may be used to convert an image from counts to flux units. For example, to create a flux-calibrated output image `outimg.fits` from an input WFPC2 image `inimg.fits[1]` with header keywords `PHOTFLAM = 2.5E-18` and `EXPTIME = 1000.0`, type:

```
st> imcalc inimg.fits[1] outimg.fits "im1*2.5E-18/1000.0"
```

If the F_λ spectrum of your source is significantly sloped across the bandpass or contains prominent features, such as strong emission lines, you may wish to recalculate the inverse sensitivity PHOTFLAM using **synphot**, described below. WF/PC-1 and WFPC2 observers should note that the PHOTFLAM values calculated during pipeline processing do not include a correction for temporal variations in throughput owing to contamination buildup, or Charge Transfer Efficiency (CTE) effects. Likewise, FOC observers should note that PHOTFLAM values determined by the pipeline before May 18, 1994 do not account for sensitivity differences in formats other than 512 x 512. Consult the appropriate instrument section of the Data Handbook for more information.



If your HST image contains a source whose flux you know from ground based measurements, you may choose to calibrate the final photometry of your HST image from the counts observed for this source.

To convert a measured flux F , in units of $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$, to an ST magnitude, the following equation may be used:

$$m = -2.5 \times \log_{10}(F) + \text{PHOTZPT}$$

where the value of the PHOTZPT keyword is the zero point of the ST magnitude or STMAG scale. The STMAG system is designed so that the

spectrum of Vega has constant flux per unit wavelength. The zero point of the STMAG system is equal to -21.10 , a value chosen so that Vega has an ST magnitude of zero for the Johnson V passband (see Koornneef et al., 1986; Horne, 1988; and the *Synphot Users Guide*). Further zeropoint corrections are necessary for converting from STMAG to other systems like Johnson/Cousins, and depend on the color of your sources. See specific photometry examples in the instrument part of the Data Handbook (Part II).

Synphot

The STSDAS synthetic photometry package, called **synphot**, can simulate HST observations of astronomical targets with known spectra. It contains throughput curves of all HST optical components, such as mirrors, filters, gratings, apertures, and detectors, and can generate passband shapes for any combination of these elements. It can also generate synthetic spectra of many different types, including stellar, blackbody, power-law and H II region spectra, and can convolve these spectra with the throughputs of HST's instruments. You can therefore use it to compare results in many different bands, to cross-calibrate one instrument with another, or to relate your observations to theoretical models.

One useful application of **synphot** is to recalculate the value of PHOTFLAM for a given observation using the latest HST sensitivity tables. The **bandpar** task may be used to compute the photometric parameters of a passband using the combined throughputs of the individual HST components. For example, to recalculate PHOTFLAM for an ACS observation, type:

```
sy> bandpar acs,wfc1,f555w
```

where the observation mode string is a comma separated list consisting of the instrument and its configuration, in this case the ACS detector with the WFC chip 1 and the F555W filter. (See the **obsmode** task in **synphot** and the *Synphot User's Guide* for help with these observation-mode keywords.) To see a list of observation mode keywords for the ACS, type:

```
sy> obsmode acs
```

Using the default parameters, the **bandpar** command shown above will print to the screen a table of photometric parameters. The URESP parameter contains the flux (in F_λ) of a source that produces a response of one count per second in this passband and is therefore identical to PHOTFLAM.

Please see the *Synphot User's Guide* for more details on this package, and see appendix A for information on getting the **synphot** dataset, which is not included with STSDAS.

3.4.5 Combining Dithered HST Datasets with MultiDrizzle

Many HST observations make use of the technique of dithering, or offsetting the telescope to different locations in order to move the target around the detector. This can be for several reasons, including sub-pixel offsets to improve PSF sampling, offsets to move bad pixels around to different locations on the sky, or large shifts comparable to the detector size, to create large mosaics of the target field.

The recommended software to combine dithered HST datasets is MultiDrizzle (Koekemoer et al. 2002, 2004), which is a Pyraf script designed to provide fully automated image registration, cosmic ray cleaning and final image combination using the drizzle software (Fruchter & Hook 2002) and PyDrizzle. MultiDrizzle is currently available within STSDAS 3.2 as a prototype release and has been tested on a representative set of commonly-used ACS, WFPC2 and STIS observing modes.

The basic required input for MultiDrizzle is simply a set of calibrated, flat-fielded files, with additional optional inputs including a user-defined bad pixel mask, or a user-defined shift file if the header shifts in the images are not sufficiently accurate. The script carries out the following steps:

- Calculate and subtract a background sky value for each exposure
- Search for additional bad pixels that are strongly negative, that may not have been flagged in the data quality arrays.
- Determine shifts from the coordinates in the image headers, which are then applied in drizzling all the input images onto a series of separate output images that are all registered with respect to one another.
- Use the drizzled exposures to create a median (optionally choosing the minimum instead of the median if there is a large enough difference)
- Transform the median back to the frame of the original input exposures and calculate the derivative of this clean median image
- Compare the median and its derivative against the original input exposure to create a cosmic ray mask for each exposure.
- Use the cosmic ray masks in a final step that drizzles all the original input exposures together onto a single output image.

The various steps can each be turned on or off by the user, since there may be cases where not all the steps need to be run or some of them may have been run already in an earlier iteration of the script. In addition, parameters controlling the behavior of each step can be adjusted by the user. The default parameter values are set such that the script should produce a scientifically useful combined, drizzled image in a single

one-touch operation. However, this may not be the optimal scientific image for a given set of exposures, therefore access is provided to parameters for drizzle and other steps for fine-tuning the results.

Please see the HST Dither Handbook (Koekemoer et al. 2004), as well as the on-line help documentation for MultiDrizzle within Pyraf, for further information about the various parameters to the script, as well as examples showing how to use the script to combine various types of HST datasets. Please note that since the software is still actively being improved, there may be some datasets for which problems might occur, in which case users are encouraged to send email to help@stsci.edu to obtain assistance. In general, however, the code has been well tested on a wide variety of the more commonly used observing modes and should produce useful results even with default parameters.

Finally, work is currently on-going to incorporate a more robust, fully redesigned version of MultiDrizzle into the operational HST Pipeline. The initial implementation of this version will be focussed on ACS associations, thus it will produce fully CR-cleaned, drizzle-combined images for all ACS associations. It will also be released within Pyraf as an off-line version capable of running WFPC2 and STIS data, in the same way as the current version. We expect the Pipeline release to occur in mid to late Summer 2004, followed shortly by the updated release within Pyraf.

3.5 Displaying HST Spectra

This section shows how to plot your HST spectra for a quick first look and how to generate hardcopies of your plots. Because the STIS data format differs from that of FOS and GHRS, we will discuss STIS data separately.

3.5.1 STIS Spectra

STIS data files retrieved from the [MAST Archive](#) can contain spectra in two different forms: as long-slit spectral images in FITS IMAGE extensions or as extracted spectra in FITS BINTABLE extensions.

Plotting STIS Imaging Spectra

You can use **sgraph** in the **graphics.stplot** package of STSDAS to plot STIS long-slit spectral images by specifying the image section that contains the spectrum. For example, to plot the entire x range of the calibrated two-dimensional spectrum in the first extension of the file

o43ba1bnm_x2d.fits, averaging rows 100 through 1000, you would type

```
st> sgraph o43ba1bnm_x2d.fits[1][*,100:1000]
```

Displaying the long-slit spectral image using the **display** task (see Section 3.3.1) allows you to see the range of your spectrum in x and y pixel space, so you can choose a suitable image section for plotting.

Plotting STIS Tabular Spectra

To plot STIS spectra in BINTABLE extensions, you first need to understand how STIS spectra are stored as binary arrays in FITS table cells. Section 2.1.2 discusses this format and describes the *selectors* syntax used to specify these data arrays. Each row of a STIS tabular spectrum contains a separate spectral order (first-order spectra will have one row, while echelle spectra will have many rows), and each column contains data of a certain type, such as wavelength or flux. To specify a particular array, you must first type the file name, then the extension containing the BINTABLE followed by the column selector, and finally the row selector. For example, to select the WAVELENGTH array corresponding to spectral order 80 of the echelle spectrum in extension 4 (EXTNAME=SCI, EXTVER=2) of `stis.fits`, you would specify the file as either:

```
stis.fits[4][c:WAVELENGTH][r:sporder=80]
or
stis.fits[sci,2][c:WAVELENGTH][r:sporder=80]
```

The **sgraph** task and the **igi** plotting package, to be discussed below, both understand the *selectors* syntax. In particular, if you wanted to plot flux vs. wavelength in STIS echelle order 80, you could type

```
st> sgraph "stis.fits[4][r:sporder=80] WAVELENGTH FLUX"
```

Remember to include the quotation marks, otherwise, **sgraph** will complain about too many positional arguments. Note also that **sgraph** understands only row selector syntax; columns are chosen by name.

The STIS-specific **echplot** task is particularly useful for browsing STIS echelle spectra. It can plot single spectral orders, overplot multiple orders on a single plot, or plot up to four orders in separate panels on the same

page. For example, to overplot the orders contained in rows two through four and row six on a single page:

```
cl> echplot "stis_x1d.fits[1][r:row=(2:4,6)]" output.igi \
>>> plot_style=m
```

Note that the `plot_style` parameter governs how the spectral orders are plotted. The `plot_style` values `s`, `m`, and `p` plot one order per page, several orders on a single plot, and one order per panel, respectively. The default brightness unit is calibrated FLUX, although you can specify other quantities (e.g., NET counts) using the `flux_col` parameter. See the online help for details.

3.5.2 FOS and GHRS Spectra

Before you work with FOS and GHRS data within STSDAS, you will want to convert the FITS files you received from the Archive into GEIS format (see Section 2.2.1 for instructions). After conversion, the `.c1h` file will hold the calibrated flux values for each pixel, the `.c0h` file will hold the corresponding wavelengths, and the `.c2h` file will hold the propagated statistical errors.

Each group of an FOS or GHRS GEIS file contains the results of a separate subintegration. FOS readouts taken in ACCUM mode are cumulative, so the last group contains the results of the entire integration. In contrast, GHRS readouts and FOS readouts in RAPID mode are independent. If you want to see the results of an entire GHRS FP-SPLIT integration, you will need to align and coadd the spectra in the groups of the GHRS file. You can also combine all the groups in an FOS or GHRS data file, without wavelength alignment, using the `rcombine` task in the `hst_calib.ctools` package. See online help for details.

Sgraph can plot the contents of a single GEIS group. For example, if you want to see group 19 of the calibrated FOS spectrum with rootname `y3b10104t` you can type

```
st> sgraph y3b10104t.c1h[19]
```

Given an input flux image (`.c1h`), the task **fwplot** (in the `hst_calib.ctools` package) will look for the corresponding wavelength (`.c0h`) file and plot flux versus wavelength. If requested, it will also look for the error (`.c2h`) file and plot the error bars. To see a plot of the same spectrum as above, but with a wavelength scale and error bars, type

```
st> fwplot y3b10104t.c1h[19] plterr+
```

If you ever need to plot the contents of multiple groups offset from one another on the same graph, you can use the **grspec** task in the **graphics.stplot** package. For example, to plot groups 1, 10, and 19 of a given flux file, you can type

```
st> grspec y3b10104t.c1h 1,10,19
```


Note that **grspec** expects group numbers to be listed as separate parameters, rather than enclosed in the standard square brackets.

3.5.3 Producing Hardcopy

This section shows how to generate hardcopies of plots directly and describes **igi**, the Interactive Graphics Interpreter available in STSDAS.

Direct Hardcopies

To print a quick copy of the displayed plot:

1. Type =gcur in the CL command window.
2. Move the cursor to any location in the graphics window.
3. Press  to write the plot to the graphics buffer.
4. Type q to exit graphics mode.
5. At the CL prompt, type gflush.

Plots will be printed on the printer defined by the IRAF environment variable `stdplot`. Type `show stdplot` to see the current default printer; use `set stdplot = printer_name` to set the default printer.

The PostScript kernel **psikern** allows you to create PostScript files of your IRAF/STSDAS plots. For example, setting the `device` parameter in a plotting task equal to `psi_port` or `psi_land` invokes **psikern** and directs your plot to either a portrait-mode or a landscape mode PostScript file. For example:

```
st> sgraph o43balbnm_x2d.fits[1][*,100:1000] device=psi_land
st> gflush
/tmp/pskxxx
```

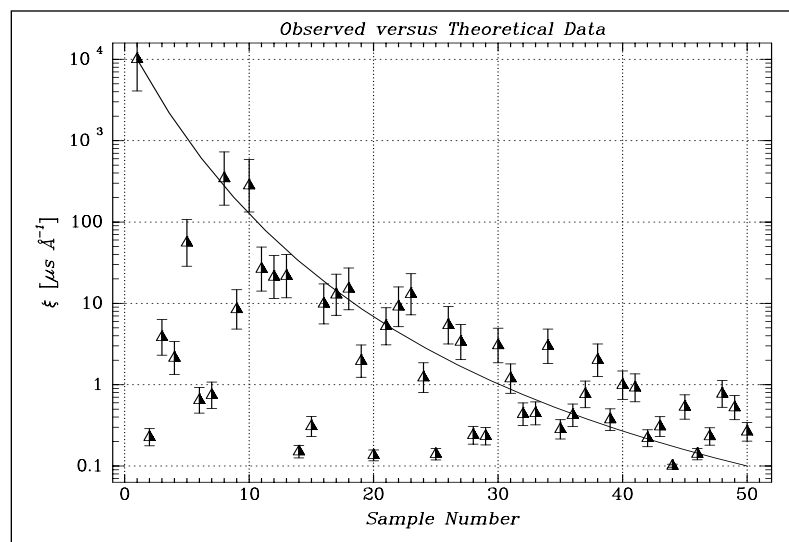
The above commands would write a plot in landscape-mode into a temporary PostScript file, named `/tmp/pskxxx` by a UNIX system. See

the online help for more about **psikern**, including plotting in color and incorporating PostScript fonts into your plots.

igi

As your plotting needs grow more sophisticated—and especially as you try preparing presentations or publication-quality plots—you should investigate the Interactive Graphics Interpreter, or **igi**. This task, in the STSDAS **stplot** package, can be used with images as well as two- and three-dimensional tables and can draw axes, error bars, labels, and a variety of other features on plots. Different line weights, font styles, and feature shapes are available, enabling you to create complex plots. Figure 3.4 shows a sample plot created in **igi**, however, because **igi** is a complete graphics environment in itself, it is well beyond the scope of this document. You can learn more about **igi** in the *IGI Reference Manual*, available through the [STSDAS Web pages](#).

Figure 3.4: Sample igi Plot.



If you are working in the Python/PyRAF environment, the plotting library **matplotlib** is available. It uses most of the MATLAB syntax.

3.6 Analyzing HST Spectra

This section describes some IRAF/STSDAS tasks that can be used for analyzing and manipulating spectral data. Some of these tasks operate directly on HST data files created by the pipeline. However, a number of the most useful IRAF tasks, such as **splot**, require special preparations of the data (except for STIS two-dimensional spectra). Before discussing these tasks we will first show how to recast your data into forms that are more generally accessible.

3.6.1 Preparing STIS Spectra for Analysis

Calibrated STIS spectra emerge from the pipeline either as two-dimensional images (`_x2d` files) or as one-dimensional spectra in tabular form (`_x1d` files.) You can analyze calibrated two-dimensional STIS spectra in IRAF as you would analyze any other long-slit spectral image, because their headers already contain the necessary wavelength information. Tabulated STIS spectra can be analyzed directly using STSDAS tasks that understand the *selectors* syntax described in Section 2.1.2. However, in order to use IRAF tasks that rely on the multispec format WCS, such as **splot**, or other STSDAS tasks that do not understand three-dimensional tables, you will have to prepare your data appropriately. This section describes some useful tasks for putting your data in the proper form:

- **tomultispec**: This task is the STIS analog to **mkmultispec**, described above. It extracts STIS spectra from tables and writes them as IRAF spectral images with wavelength information in the header.
- **txtable**: This task extracts specified data arrays from STIS table cells and places them in conventional two-dimensional tables for easier access.
- **tximage**: Extracts specified data arrays from STIS table cells and places them into 1-D images. This task can write single group GEIS files.

tomultispec

The **tomultispec** task in the `stdas.hst_calib.ctools` package extracts one spectral order (or several, for echelle data) from a STIS table, fits a polynomial dispersion solution to each wavelength array, and stores the spectra in an output file in original IRAF format (OIF), using the multispec WCS. This task is layered upon the **mkmultispec** task, which performs a similar operation for FOS and GHRS calibrated spectra (see “**mkmultispec**” on page 3-31). Most of the parameters for **tomultispec** echo those for **mkmultispec**. As a helpful navigational aid, the STIS spectral order numbers are written to the corresponding *beam* numbers in the multispec image; the aperture numbers are indexed sequentially starting from one. You can choose to fit the dispersion solution interactively, but the default fourth-order Chebyshev polynomial will likely suffice for all STIS spectral orders, except for prism-dispersed spectra. However, you cannot use the interactive option if you are selecting more than one order from the input file.

For example, if you want to write all spectral orders from the STIS file `myfile_x1d.fits` to a multispec file, type

```
cl> tomultispec myfile_x1d.fits new_ms.imh
```

The output file format of **tomultispec** will be OIF regardless of the specified extension. This format is similar to GEIS format, (see Section A.2.6). OIF format files have a header component (extension `.imh`) and a binary data component (extension `.pix`).

If you want to select particular spectral orders, rather than writing all the orders to the multispec file, you will need to use the *selectors* syntax. To select only the spectrum stored in row nine of the input table, the previous example would change to:

```
cl> tomultispec "myfile_x1d.fits[r:row=9]" new_ms.imh
```

Note that the double quote marks around the file name and row selector are necessary to avoid syntax errors. To select a range of rows, say rows nine through eleven:

```
cl> tomultispec "myfile_x1d.fits[r:row=(9:11)]" new_ms.imh
```

You can also select rows based upon values in some other column. For example, to select all rows whose spectral order lies in the range 270 to 272, type:

```
cl> tomultispec "myfile_x1d.fits[r:sporder=(270:272)]" \  
>>> new_ms.imh
```

Be careful not to restrict the search for matching rows too heavily.

Tomultispec extracts the calibrated flux by default. However, other intensity data (e.g., NET counts) can be extracted by specifying the `flux_col` parameter appropriately.



Column selectors cannot be used with tomultispec.



Choose the type of fitting function for the tomultispec dispersion solution with care. Using the table option, which writes the entire wavelength array to the image header for each order, will fail if more than about three orders are selected. This restriction results from a limit to the number of keywords that can be used to store the dispersion relation.

txtable

STIS spectral information is stored as data arrays within individual cells of FITS binary tables (see Section 2.1.2). These tables are effectively three-dimensional, with each column holding a particular type of quantity (e.g., wavelengths, fluxes), each row holding a different spectral order, and each cell holding a one-dimensional array of values spanning the wavelength space of each spectral order. The **txtable** task in the **tables.ttools** package extracts these data arrays from the cells specified with the *selectors* syntax and stores them in the columns of conventional two-dimensional binary tables.

For example, suppose the first extension of the FITS file `data.fits` contains a STIS echelle spectrum and you want to extract only the wavelength and flux arrays corresponding to spectral order 68. You could then type:

```
tt> txtable "data.fits[1][c:WAVELENGTH,FLUX][r:sporder=68]" \
>>> out_table
```

This command would write the wavelength and flux arrays to the columns of the output table `out_table`. To specify multiple rows of a tabulated echelle spectrum, you would type:

```
tt> txtable "data.fits[1][c:WAVELENGTH,FLUX][r:row=(10:12)]" \
>>> echl
```

This command would generate three separate output files named `echl_r0010.tab`, `echl_r0011.tab`, and `echl_r0012.tab`. See the online help for more details on **txtable** and the *selectors* syntax, and remember to include the double quotation marks. The similar **tximage** task can be used to generate single-group GEIS files from STIS tabular data, which can then be used as input to tasks such as **resample**.

```
tt> tximage "data.fits[1][c:WAVELENGTH][r:row=4]" wave.hhh
tt> tximage "data.fits[1][c:FLUX][r:row=4]" flux.hhh
```

3.6.2 Preparing FOS and GHRS Data

The FOS and GHRS data reduction pipelines store fluxes and wavelengths in separate files. In GEIS format, the `.c1h` file contains the flux information and the `.c0h` file contains the wavelength information. Because IRAF tasks generally require both the flux and wavelength information to reside in the same file, you will probably want to create a new file that combines these arrays.

Several options for combining flux and wavelength information are available:

- **resample**: This simple task resamples your flux data onto a linear wavelength scale, creating a new flux file containing the starting wavelength of the new grid in the CRVAL1 keyword and the wavelength increment per pixel in the CD1_1 keyword. Encoding the wavelength information into these standard FITS header keywords makes this format quite portable, but the resampling process loses some of the original flux information. In addition, the error (.c2h) and data quality (.cqh) files cannot be similarly resampled, limiting the usefulness of this technique.
- **mkmultispec**: This task writes wavelength information into the header of a flux file while preserving all the original information. It is therefore a better choice than **resample** for most applications, and we describe it in more detail below.
- **imtab**: An alternative to writing wavelength information into the header is to use the **imtab** task to create a table recording the wavelength, flux, and if desired, the error data corresponding to each pixel. Many STSDAS tasks, such as those in the STSDAS **fitting** package, can access data in tabular form, so we describe this approach in more detail as well.

mkmultispec

The most convenient method of combining wavelength and flux information, and one that has no effect on the flux data at all, is to use the **mkmultispec** task. This task places wavelength information into the headers of your flux files according to the IRAF multispec format WCS. The multispec coordinate system is intended to be used with spectra having nonlinear dispersions or with images containing multiple spectra, and the format is recognized by many tasks in IRAF V2.10 or later. For a detailed discussion of the multispec WCS, type `help specwcs` at the IRAF prompt.

The **mkmultispec** task can put wavelength information into the flux header files in two different ways. The first involves reading the wavelength data from the .c0h file, fitting the wavelength array with a polynomial function, and then storing the derived function coefficients in the flux header file (.c1h) in multispec format. Legendre, Chebyshev, or cubic spline (spline3) fitting functions of fourth order or larger produce essentially identical results, all having rms residuals less than 10^{-4} Å, much smaller than the uncertainty of the original wavelength information.

Because these fits are so accurate, it is usually unnecessary to run the task in interactive mode to examine them.



If there are discontinuities in the wavelengths, which could arise due to the splicing of different gratings, you should run `mkmultispec` in interactive mode to verify the fits.



Because `mkmultispec` can fit only simple types of polynomial functions to wavelength data, this method will not work well with FOS prism data, because of the different functional form of the prism-mode dispersion solution. For prism spectra, use the header table mode of `mkmultispec` (see below) or create an STSDAS table using `imtab`.

There is another method by which `mkmultispec` can incorporate wavelength information into a flux file and that is simply to read the wavelength data from the `.c0h` file and place the entire data array directly into the header of the flux (`.c1h`) file. This method simply dumps the wavelength value associated with each pixel in the spectrum into the flux header and is selected by setting the parameter `function=table`. To minimize header size, set the parameter `format` to a suitable value. For example, using `format=%8.7g` will retain the original seven digits of precision of the wavelength values, while not consuming too much space in the flux header file.



Be aware that there is a physical limit to the number of header lines that can be used to store the wavelength array (approximately 1000 lines). This limit cannot be overridden. Under ordinary circumstances this limitation is not an issue. However, if many spectral orders have been spliced together, it may not be possible to store the actual wavelength array in the header, and a fit must be done instead.

imtab

Another way to combine wavelengths with fluxes is to create an STSDAS table from your spectrum. The `imtab` task in the STSDAS `ttools` package reads a GEIS format spectral image and writes the list of data values to a column of an STSDAS table, creating a new output table if

necessary. The following example shows how to create a flux, wavelength, and error table from group eight of a GEIS-format FOS dataset:

```
c1> imtab y0cy0108t.c0h[8] y0cy0108t.tab wavelength
c1> imtab y0cy0108t.c1h[8] y0cy0108t.tab flux
c1> imtab y0cy0108t.c2h[8] y0cy0108t.tab error
```

The last word on each command line labels the three columns “wavelength”, “flux”, and “error”.

Constructing tables is necessary if you plan to use certain tasks—such as those in the STSDAS **fitting** package—that do not currently recognize the multispec format WCS header information. Tabulating your spectra is also the best option if you want to join two or more spectra taken with different gratings into a single spectrum covering the complete wavelength range. Because the data are stored as individual wavelength-flux pairs, you do not need to resample them (thereby degrading the individual spectra to a common linear dispersion scale) before joining them. Instead, you could create separate tables for spectra from different gratings, and then combine the two tables using, for example, the **tmerge** task:

```
c1> tmerge n5548_h13.tab,n5548_h19.tab n5548.tab append
```

Note that you will first have to edit out any regions of overlapping wavelength from one or the other of the input tables so that the output table will be monotonically increasing (or decreasing) in wavelength.

3.6.3 Photometry

For STIS, FOS and GHRS spectra, photometric correction is done by the pipeline during spectral extraction, resulting in flux-calibrated spectra. For STIS, see the “Photometric Calibration” section 3.4.12, in the *STIS Data Handbook*.

3.6.4 General Tasks for Spectra

IRAF has many tasks for analyzing both one- and two-dimensional spectral data. Many observers will already be familiar with **noao.onedspec** and **noao.twodspec** packages, and those who are not should consult the online help. Table 3.5 lists some of the more commonly used IRAF/STSDAS spectral analysis tasks, and below we briefly describe **splot**, one of the most versatile and useful. Remember that many of these tasks expect to find WCS wavelength information in the header, so you should first run **mkmultispec** or **tomultispec** on your data, if necessary.

Table 3.5: Spectral Analysis Tasks

Task	Package	Input Format	Purpose
boxcar	images.imfilter	Image	Boxcar smooth a list of images
bplot	noao.onedspec	Multispec image ¹	Plot spectra non-interactively
continuum	noao.onedspec	Image	Continuum normalize spectra
fitprofs	noao.onedspec	Image	Non-interactive Gaussian profile fitting to features in spectra and image lines
gcopy	stsdas.toolbox.imgtools	GEIS image	Copy multigroup images
grlist	stsdas.graphics.stplot	GEIS image	List file names for all groups in a GEIS image; used to make lists for tasks that do not use group syntax
grplot	stsdas.graphics.stplot	GEIS image	Plot arbitrary lines from 1-D image; overplots multiple GEIS groups; no error or wavelength information is used
grspec	stsdas.graphics.stplot	Multispec GEIS image	Plot arbitrary lines from 1-D image; stack GEIS groups
magnify	images.imgeom	Image	Interpolate spectrum on finer (or coarser) pixel scale
nfit1d	stsdas.analysis.fitting	Image, table	Interactive 1-D non-linear curve fitting (see Section 3.6.5)
ngaussfit	stsdas.analysis.fitting	Image, table	Interactive 1-D multiple Gaussian fitting (see Section 3.6.5)
poffsets	stsdas.hst_calib.ctools	GEIS image	Determine pixel offsets between shifted spectra
rapidlook	stsdas.hst_calib.ctools	GEIS image	Create and display a 2-D image of stacked 1-D images
rcombine	stsdas.hst_calib.ctools	GEIS image	Combine (sum or average) GEIS groups in a 1-D image with option of propagating errors and data quality values
resample	stsdas.hst_calib.ctools	GEIS image	Resample FOS and GHRS data to a linear wavelength scale (see Section 3.6.2)
sarith	noao.onedspec	Multispec image ¹	Spectrum arithmetic
scombine	noao.onedspec	Multispec image ¹	Combine spectra
sfit	noao.onedspec	Multispec image ¹	Fit spectra with polynomial function
sgraph	stsdas.graphics.stplot	Image, table	Plot spectra and image lines; allows overplotting of error bars and access to wavelength array (see Section 3.5.2)
specalign	stsdas.hst_calib.ctools	GEIS image	Align and combine shifted spectra (see ppoffsets)
specplot	noao.onedspec	Multispec image ¹	Stack and plot multiple spectra
splot	noao.onedspec	Multispec image ¹	Plot and analyze spectra & image lines (see “splot” on page 3-35)

1. Multispec image is a spectrum created with **tomutispec** or **mkmultispec**.

splot

The **splot** task in the IRAF **noao.onedspec** package is a good general analysis tool that can be used to examine, smooth, fit, and perform simple arithmetic operations on spectra. Because it looks in the header for WCS wavelength information, your file must be suitably prepared. Like all IRAF tasks, **splot** can work on only one group at a time from a multigroup GEIS file. You can specify which GEIS group you want to operate on by using the square bracket notation, for example:

```
c1> splot y0cy0108t.c1h[8]
```

If you don't specify a group in brackets, **splot** will assume you want the first group. In order to use **splot** to analyze your FOS or GHRIS spectrum, you will first need to write the wavelength information from your .c0h file to the header of your .c1h files in WCS, using the **mkmultispec** task (see “**mkmultispec**” on page 3-31).

The **splot** task has *many* available options described in detail in the online help. Table 3.6 summarizes a few of the more useful cursor commands for quick reference. When you are using **splot**, a log file saves results produced by the equivalent width or de-blending functions. To specify a file name for this log file, you can set the **save_file** parameter by typing, for example:

```
c1> splot y0cy0108t.c1h[8] save_file=results.log
```

If you have used **tomultispec** to transform a STIS echelle spectrum into .imh/.pix OIF files with WCS wavelength information (see “**tomultispec**” on page 3-28), you can step through the spectral orders stored in image lines using the “)”, “(”, and “#” keys. To start with the first entry in your OIF file, type:

```
c1> splot new_ms.imh 1
```

You can then switch to any order for analysis using the “)” key to increment the line number, the “(“ key to decrement, and the “#” key to switch to a specified image line. Note that the beam label which indicates the spectral order cannot be used for navigation. See the online help for details.

Table 3.6: Useful plot Cursor Commands

Command	Purpose
<i>Manipulating spectra</i>	
f	Arithmetic mode; add and subtract spectra
l	Convert spectrum from f_ν to f_λ
n	Convert spectrum from f_λ to f_ν
s	Smooth with a boxcar
u	Define linear wavelength scale using two cursor markings
<i>Fitting spectra</i>	
d	Mark two continuum points & de-blend multiple Gaussian line profiles
e	Measure equivalent width by marking points around target line
h	Measure equivalent width assuming Gaussian profile
k	Mark two continuum points and fit a single Gaussian line profile
m	Compute the mean, RMS, and S/N over marked region
t	Enter interactive curve fit function (usually used for continuum fitting)
<i>Displaying and redrawing spectra</i>	
a	Expand and autoscale data range between cursor positions
b	Set plot base level to zero
c	Clear all windowing and redraw full current spectrum
r	Redraw spectrum with current windowing
w	Window the graph
x	Etch-a-sketch mode; connects two cursor positions
y	Overplot standard star values from calibration file
z	Zoom graph by a factor of two in X direction
\$	Switch between physical pixel coordinates and world coordinates
<i>General file manipulation commands</i>	
?	Display help
g	Get another spectrum
i	Write current spectrum to new or existing image
q	Quit and go on to next input spectrum

3.6.5 STSDAS Fitting Package

The STSDAS **fitting** package contains several tasks, as listed in Table 3.7, for fitting and analyzing spectra and images. The **ngaussfit** and **nfit1d** tasks, in particular, are very good for interactively fitting multiple Gaussians and nonlinear functions, respectively, to spectral data. These tasks do not currently recognize the multispec WCS method of storing wavelength information. They recognize the simple sets of dispersion keywords such as W0, WPC and CRPIX, CRVAL, and CDELTA, but these forms only apply to linear coordinate systems and therefore would require resampling of your data onto a linear wavelength scale first. However, these tasks do accept input from STSDAS tables, in which you can store the wavelength and flux data value pairs or wavelength, flux, error value triples (see “imtab” on page 3-32).

Table 3.7: Tasks in the STSDAS fitting Package

Task	Purpose
function	Generate functions as images, tables, or lists
gfit1d	Interactive 1-d linear curve fit to images, tables, or lists
i2gaussfit	Iterative 2-d Gaussian fit to noisy images (script)
nfit1d	Interactive 1-d non-linear curve fit to images, tables, or lists
ngaussfit	Interactive 1-d multiple Gaussian fit to images, tables, or lists
n2gaussfit	2-d Gaussian fit to images
prfit	Print contents of fit tables created by fitting task

When using tasks such as **ngaussfit** and **nfit1d**, you must provide initial guesses for the function coefficients as input to the fitting algorithms. You can either specify these initial guesses via parameter settings in the task’s parameter sets (psets) or enter them interactively. For example, suppose you want to fit several features using the **ngaussfit** task. Using the default parameter settings, you can start the task by typing:

```
fi> ngaussfit n4449.hhh linefits.tab
```

This command reads spectral data from the image `n4449.hhh` and stores the results of the line fits in the STSDAS table `linefits.tab`. After you start the task, your spectrum should appear in a plot window and the task will be left in cursor input mode. You can use the standard IRAF cursor mode commands to rewindow the plot, restricting your display to the region around a particular feature, or features, that you want to fit. You may then want to:

- Define a sample region (using the cursor mode `S` command) over which the fit will be computed so that the task will not try to fit the entire spectrum.
- Define an initial guess for the baseline coefficients by placing the cursor at two baseline locations (one on either side of the feature to be fitted) using the `B` keystroke.
- Use the `R` keystroke to redraw the screen and see the baseline that you've just defined.
- Set the initial guesses for the Gaussian centers and heights by placing the cursor at the peak of each feature and typing `P`.
- Press `F` to compute the fit once you've marked all the features you want to fit.

The results will automatically be displayed. You can use the `:show` command to see the coefficient values.

Note that when the `ngaussfit` task is used in this way (i.e., starting with all default values), the initial guess for the FWHM of the features will be set to a value of one. Furthermore, this coefficient and the coefficients defining the baseline are held fixed by default during the computation of the fit, unless you explicitly tell the task through cursor `colon` commands² to allow these coefficients to vary. It is sometimes best to leave these coefficients fixed during an initial fit, and then to allow them to vary during a second iteration. This rule of thumb also applies to the setting of the `errors` parameter which controls whether or not the task will estimate error values for the derived coefficients. Because the process of error estimation is very CPU-intensive, it is most efficient to leave the error estimation turned off until you have a good fit, and then turn the error estimation on for one last iteration.

Figure 3.5 and Figure 3.6 show the results of fitting the H β (4861Å) and [OIII] (4959 and 5007 Å) emission features in the spectrum of NGC 4449. The resulting coefficients and error estimates (in parentheses) are shown in Figure 3.6.

2. To see the online help for details and a complete listing of cursor mode colon commands: type `help cursor`.

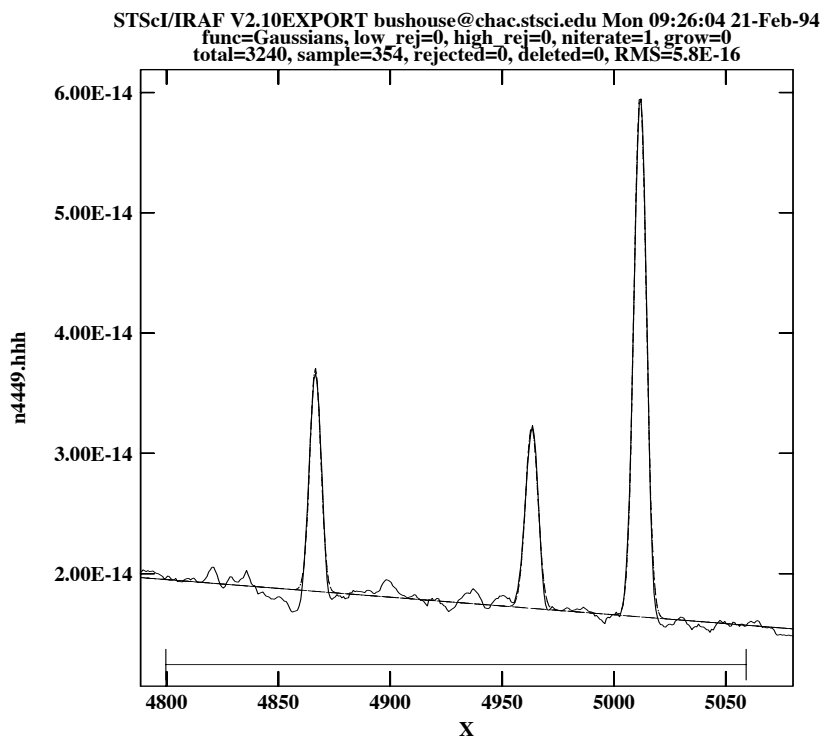
Figure 3.5: Fitting H β and [OIII] Emission Features in NGC 4449

Figure 3.6: Coefficients and Error Estimates

```

function = Gaussians
coeff1 = 8.838438E-14 (0.) - Baseline zeropoint (fix)
coeff2 = -1.435682E-17 (0.) - Baseline slope (fix)
coeff3 = 1.854658E-14 (2.513048E-16) - Feature 1: amplitude (var)
coeff4 = 4866.511 (0.03789007) - Feature 1: center (var)
coeff5 = 5.725897 (0.0905327) - Feature 1: FWHM (var)
coeff6 = 1.516265E-14 (2.740680E-16) - Feature 2: amplitude (var)
coeff7 = 4963.262 (0.06048062) - Feature 2: center (var)
coeff8 = 6.448922 (0.116878) - Feature 2: FWHM (var)
coeff9 = 4.350271E-14 (2.903318E-16) - Feature 3: amplitude (var)
coeff10 = 5011.731 (0.01856957) - Feature 3: center (var)
coeff11 = 6.415922 (0.03769293) - Feature 3: FWHM (var)
rms = 5.837914E-16
grow = 0.
naverage = 1
low_reject = 0.
high_reject = 0.
niterate = 1
sample = 4800.132:5061.308

```

3.6.6 Specfit

The **specfit** task, in the STSDAS **contrib** package, is another powerful interactive facility for fitting a wide variety of emission-line, absorption-line, and continuum models to a spectrum. This task was written by Gerard Kriss. Extensive online help is available to guide you through the task,³ although because it is a contributed task, little-to-no support is provided by the STSDAS group.

The input spectrum to **specfit** can be either an IRAF image file or an ASCII file with a simple three-column (wavelength, flux, and error) format. If the input file is an IRAF image, the wavelength scale is set using values of W0 and WPC or CRVAL1 and CDELTA1. Hence, for image input, the spectral data must be on a linear wavelength scale. In order to retain data on a non-linear wavelength scale, it is necessary to provide the input spectrum in an ASCII file, so that you can explicitly specify the wavelength values associated with each flux value. The online help explains a few pieces of additional information that must be included as header lines in an input text file.

By selecting a combination of functional forms for various components, you can fit complex spectra with multiple continuum components, blended emission and absorption lines, absorption edges, and extinction. Available functional forms include linear, power-law, broken power-law, blackbody, and optically thin recombination continua, various forms of Gaussian emission and absorption lines, absorption-edge models, Lorentzian line profiles, damped absorption-line profiles, and mean galactic extinction.

3.7 References

3.7.1 Available from STScI

From the STSDAS web page,

http://www.stsci.edu/resources/software_hardware/stsdas/

the following documents are available:

- *STSDAS Users Guide*, version 1.3, September 1994.
- *STSDAS Site Manager's Installation Guide and Reference*, version 3.0, July 2002.
- *Synphot Users Guide*, December 1998.
- *IGI Reference Manual*, version 2.0, September 1998
- *Dither Handbook*, version 2.0, January 2002.

3. Additional information is available in the *Astronomical Data Analysis Software and Systems III*, ASP Conference Series, Vol. 61, page 437, 1994.

3.7.2 Available from NOAO

From the NOAO web page, <http://iraf.noao.edu/docs/photom.html>, the following documents are available:

- *A Beginners Guide to Using IRAF*, 1993, J. Barnes.
- *Photometry Using IRAF*, version 2.10, 1994, L. Wells.
- *A User's Guide to Stellar CCD Photometry with IRAF*, 1992, P. Massey and L. Davis.

3.7.3 Other References Cited in This Chapter

- Fruchter, A. S & Hook, R. N. 2002, PASP 114, 144
- Horne, K., 1988, in *New Directions in Spectrophotometry*, A.G.D. Philip, D.S. Hayes, and S.J. Adelman, eds., L. Davis Press, Schenectady NY, p. 145.
- Koekemoer, A. M., Fruchter, A. S., Hook, R. N., & Hack, W. 2002, HST Calibration Workshop, ed. S. Arribas, A. M. Koekemoer, & B. Whitmore (STScI: Baltimore), 337
- Koekemoer, A. M. et al. 2004, HST Dither Handbook, V3.0
- Koorneef, J., R. Bohlin, R. Buser, K. Horne, and D. Turnshek, 1986, in *Highlights of Astronomy*, Vol. 7, J.-P. Swinds, ed., Reidel, Dordrecht, p. 833.
- Kriss, G., 1994, in *Astronomical Data Analysis Software and Systems III*, PASP Conference Series, Vol. 61, p. 437.
- www.stsci.edu/resources/software_hardware/pyraf
- www.stsci.edu/resources/software_hardware/pyfits
- www.stsci.edu/resources/software_hardware/numarray



PART II:

ACS Data Handbook

This handbook is designed to help users manipulate, process and analyze data from the Advanced Camera for Surveys (ACS) which will be installed on-board the Hubble Space Telescope (HST) during the 2002 servicing mission (SM3B).

■ **Part II:ACS Data Handbook**

ACS Introduction

How to Use this Handbook

This handbook is designed to help users manipulate, process and analyze data from the Advanced Camera for Surveys (ACS) which was installed on-board the Hubble Space Telescope (HST) during the 2002 servicing mission (SM3B). It is presented as an independent and self-contained document and is designed for users familiar with HST data but new to ACS. Users who wish to find more general information including instructions for retrieving data from the HST archive, a description of HST file formats, and a discussion of IRAF/STSDAS software for displaying and processing these data, are referred to a companion volume, the Introduction to the *HST Data Handbook* at:

http://www.stsci.edu/hst/HST_overview/documents/datahandbook

While the purpose of this handbook is to provide comprehensive information for the treatment of ACS data and present the in-flight data calibration program results, a brief discussion of the ACS capabilities, design and basic instrument operations (Chapter 1) is also included. For a detailed discussion please refer to the ACS Instrument Handbook. One important concern for users is the large volume of ACS data compared to other HST instruments, addressed in Chapter 2, with a discussion of the ACS file structure. Chapter 3 describes the ACS calibration pipeline software, the ACS reference files, and instructions for manually recalibrating data. Given the large ACS field of view, and hence the importance of geometric distortion, Chapter 4 is devoted to a thorough discussion of the ACS distortion. This includes the new Python package **PyDrizzle** which corrects for distortion in ACS images and drizzles multiple dithered images within **CALACS**. We also recommend the new **MultiDrizzle** software for users who wish to improve or fine-tune the drizzling process, choosing parameters specific to their scientific goals. Chapter 5 describes the sources of uncertainty in ACS data and chapter 6 describes the data reduction issues relevant for work with ACS data.

4 ■ ACS Introduction

Since many of the instrument characteristics may be revised over a short time frame, readers are advised to consult the ACS web pages (<http://www.stsci.edu/hst/acs/>) for the latest information regarding ACS performance and calibration.

Cheryl Pavlovsky, Adam Riess, Jennifer Mack & Ron Gilliland
(Editors, ACS Data Handbook)

ACS Overview

In this chapter. . .

1.1 Instrument Design and Capabilities / 1-1

1.2 Basic Instrument Operations / 1-7

This chapter provides an overview of the capabilities and design of the Advanced Camera for Surveys (ACS) and describes the basic instrument operations. For more information we refer you to the *ACS Instrument Handbook* which gives a technical description of the instrument's properties, expected performance, operations, and calibration.

1.1 Instrument Design and Capabilities

The ACS camera is designed to provide HST with a deep, wide-field survey capability from the visible to near-IR, high resolution imaging from the near-UV to the near-IR, and solar blind far-UV imaging. The primary design goal of the ACS Wide-Field Channel is to achieve a factor of 10 improvement in discovery efficiency compared to WFPC2, where discovery efficiency is defined as the product of imaging field of view (FOV) and instrument throughput.

ACS comprises three channels, each optimized for a specific goal:

- Wide field channel (WFC): $\sim 202 \times 202$ arcsecond field of view from 3700–11,000 Å, and peak efficiency of 48% (including the Optical Telescope Assembly). The plate scale of ~ 0.049 arcsec/pixel provides critical sampling at 11,600 Å.

- High resolution channel (HRC): 29×25 arcsecond field of view from 2000–11,000 Å, and peak efficiency of 29%. The plate scale of ~ 0.027 arcsec/pixel provides critical sampling at 6300 Å.
- Solar Blind Channel (SBC): 35×31 arcsecond field of view from 1150–1700 Å, and peak efficiency of 7.5%. The plate scale of ~ 0.032 arcsec/pixel provides a good compromise between resolution and field of view.

In addition to these three prime capabilities, ACS also provides:

- Grism spectroscopy: Low resolution ($R \sim 100$) wide field spectroscopy from 5500–11,000 Å available in both the WFC and the HRC.
- Objective prism spectroscopy: Low resolution ($R \sim 100$ @ 2000 Å) near-UV spectroscopy from 2000–4000 Å available in the HRC.
- Objective prism spectroscopy: Low resolution ($R \sim 100$ @ 1216 Å) far-UV spectroscopy from 1150–1700 Å available in the SBC.
- Coronagraphy: Aberrated beam coronagraphy in the HRC from 2000–11,000 Å with 1.8 arcsecond and 3.0 arcsecond diameter occulting spots.
- Imaging Polarimetry: Polarimetric imaging in the HRC and WFC with relative polarization angles of 0° , 60° and 120° .

ACS is a versatile instrument that can be applied to a broad range of scientific programs. For example, the high sensitivity and wide field of the WFC in the visible and near-infrared will make it the instrument of choice for deep imaging programs in this wavelength region. The HRC, with its excellent spatial resolution, provides full sampling of the HST PSF at $\lambda > 6000$ Å and can be used for high precision photometry in stellar population programs. The HRC coronagraph can be used for the detection of circumstellar disks and QSO host galaxies.

1.1.1 Detectors

ACS uses one or more large-format detectors in each channel:

- The WFC detector, called ACS/WFC, employs a mosaic of two 2048×4096 Scientific Imaging Technologies (SITe) CCDs, with ~ 0.049 arcsecond pixels, covering a nominal $\sim 202 \times 202$ arcsecond field of view, and a spectral response from ~ 3700 to 11,000 Å.
- The HRC detector, called ACS/HRC, is a 1024×1024 SITe CCD, with $\sim 0.028 \times 0.025$ arcsecond pixels, covering a nominal 29×25 arcsecond field of view, and spectral response from ~ 2000 to 11,000 Å.

- The SBC detector, called the ACS/SBC, is a solar-blind CsI Multi-Anode Microchannel Array (MAMA), with 1024×1024 $\sim 0.034 \times 0.030$ arcsecond pixels, and a nominal 35×31 arcsecond field of view, with far-UV spectral response from ~ 1150 to 1700 \AA .

The WFC & HRC CCDs

The ACS CCDs are thinned, backside-illuminated devices cooled by thermo-electric cooler (TEC) stacks and housed in sealed, evacuated dewars with fused silica windows. The spectral response of the WFC CCDs is optimized for imaging at visible to near-IR wavelengths. The HRC CCD covers wavelengths similar to the WFC but the spectral response has been additionally optimized for the near-UV. Both CCD cameras produce a time-integrated image in the ACCUM data-taking mode. The HRC can also be operated in target acquisition (ACQ) mode for coronagraphic observations. As with all CCD detectors, there is noise (*readout noise*) and time (*read time*) associated with reading out the detector following an exposure. The minimum exposure time is 0.1 sec for HRC, and 0.5 sec for WFC, and the minimum time between successive identical exposures is 45 sec (HRC) or $\sim 135 \text{ sec}$ (WFC) for full-frame and can be reduced to $\sim 36 \text{ sec}$ for subarray readouts. The dynamic range for a single exposure is ultimately limited by the depth of the CCD full well ($\sim 85,000 e^-$ for the WFC and $155,000 e^-$ for the HRC), which determines the total amount of charge that can accumulate in any one pixel during an exposure without saturation. Cosmic rays will affect all CCD exposures: CCD observations should be broken into multiple exposures whenever possible, to allow removal of cosmic rays in post-observation data processing.

The SBC MAMA

The SBC MAMA is a *photon-counting* detector which provides a two-dimensional ultraviolet capability, optimized for the far-UV. It can only be operated in ACCUM mode. The ACS MAMA detector is subject to both *scientific* and *absolute* brightness limits. At high local ($\geq 50 \text{ counts sec}^{-1} \text{ pixel}^{-1}$) and global ($> 285,000 \text{ counts sec}^{-1}$) illumination rates, counting becomes nonlinear in a way that is not correctable. At only slightly higher illumination rates, the MAMA detectors are subject to damage. We have therefore defined absolute local and global count-rate limits, which translate to a set of configuration-dependent bright-object screening limits. Sources which violate the absolute count rate limits in a given configuration *cannot be observed in those configurations*.

1.1.2 ACS Optical Design

The ACS design incorporates two main optical channels: one for the WFC and one which is shared by the HRC and SBC. These channels are illustrated in figures 3.1 and 3.2 of the *ACS Instrument Handbook*. Each channel has independent corrective optics to compensate for HST's spherical aberration. The WFC has three optical elements, coated with silver, to optimize instrument throughput in the visible. The silver coatings cut off at wavelengths shortward of 3700 Å. The WFC has two filter wheels which it shares with the HRC, offering the possibility of internal WFC/HRC parallel observing for some filter combinations. The HRC/SBC optical chain comprises three aluminized mirrors, overcoated with MgF₂. The HRC or SBC channels are selected by means of a plane fold mirror. The HRC is selected by inserting the fold mirror into the optical chain so that the beam is imaged onto the HRC detector through the WFC/HRC filter wheels. The SBC channel is selected by moving the fold mirror out of the beam to yield a two mirror optical chain which images through the SBC filter wheel onto the SBC detector. The aberrated beam coronagraph is accessed by inserting a mechanism into the HRC optical chain. This mechanism positions a substrate with two occulting spots at the aberrated telescope focal plane and an apodizer at the re-imaged exit pupil.

While there is no mechanical reason why the coronagraph could not be used with the SBC, for health and safety reasons (due to brightness limits of the MAMA detector) **use of the coronagraph is forbidden with the SBC.**

1.1.3 ACS Geometric Distortion

The ACS detectors exhibit significantly more distortion than previous HST instruments. All ACS observations must be corrected for distortion before any photometry or astrometry is derived. For a thorough discussion of ACS Geometric Distortion, we refer the reader to Chapter 4.

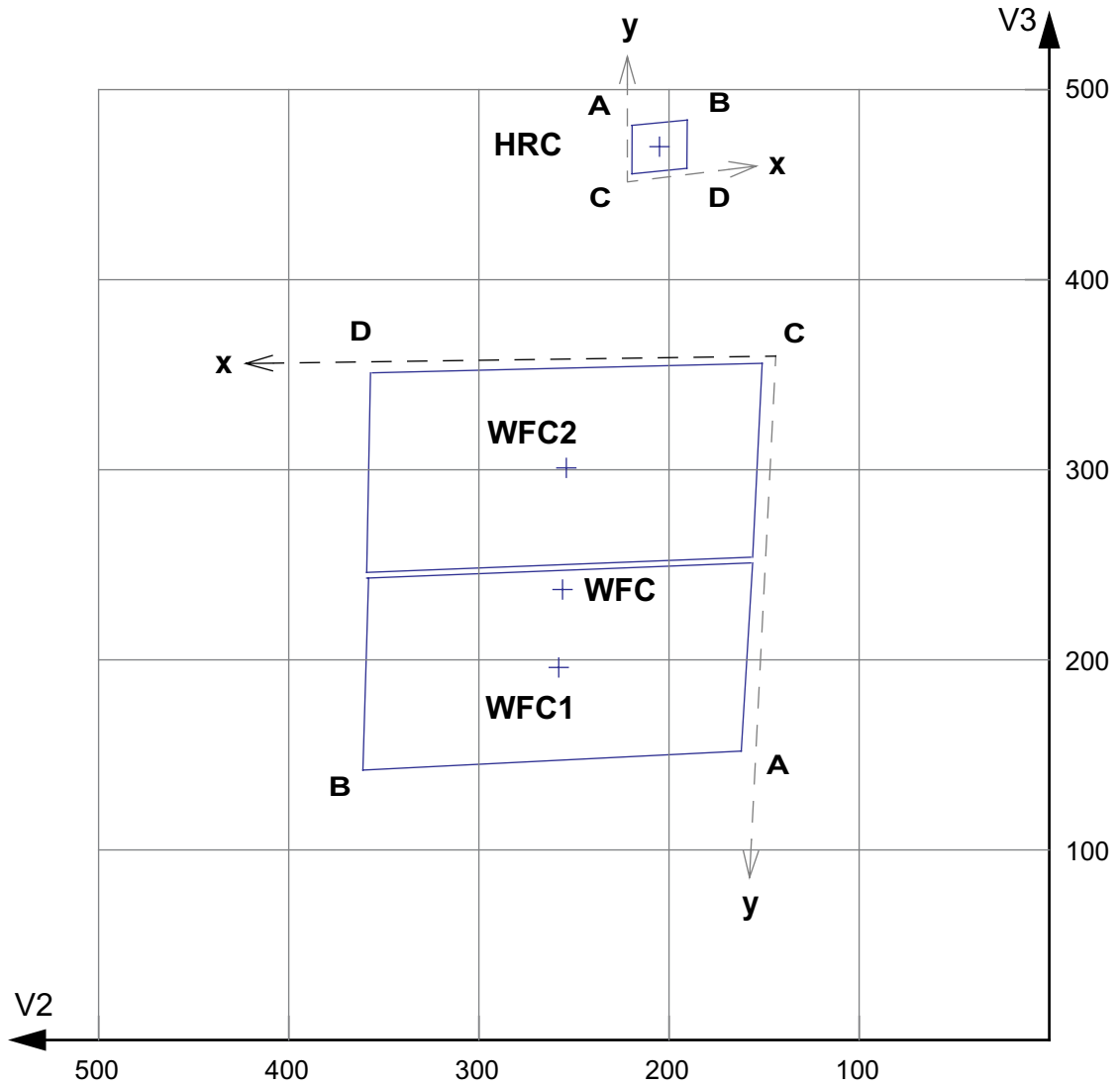
The principal cause of the ACS distortion is that the optics have been designed with a minimum number of components, consistent with correcting for the spherical aberration induced by the Optical Telescope Assembly (OTA), without introducing coma. The result is a high throughput, but focal surfaces far from normal to the principal rays. The WFC detector is tilted at 22 degrees giving an elongation of 8% along the diagonal. The HRC and SBC detectors have a 25 degree tilt giving an elongation of 12%. In each case, the scale in arcseconds per pixel is smaller along the radial direction of the OTA field of view than along the tangential direction. When projected on the sky, this causes each detector to appear "rhombus-shaped" rather than square. The angle on the sky between the x and y axes is 84.9 degrees for the WFC1, 86.1 for the WFC2 and 84.2 degrees for the HRC.

The orientations of the ACS detector edges are approximately in line with the V2 and V3 coordinate axes of the telescope. Consequently, the eigenaxes of the scale transformation are along the diagonals for WFC and the apertures and pixels appear non-rectangular in the sky projection. For the HRC and SBC the situation is even more irregular because the aperture diagonals do not lie along a radius of the HST field of view. Figure 1.1 shows the ACS apertures in the telescope's V2/V3 reference frame and illustrates the "rhombus" shape of each detector. A telescope roll angle of zero degrees would correspond to an on-sky view with the V3 axis aligned with North and the V2 with East. The readout amplifiers are also indicated for each detector.

If these were the only distortions present, their impact on photometry and mosaicing/dithering could be simply computed. A more problematic effect is the variation of scale and pixel area across each detector. For the WFC this amounts to a change of $\sim 10\%$ in scale from corner to corner. For the HRC and SBC this variation is only about 1% , since these detectors cover much smaller fields of view. The area on the sky covered by a WFC pixel varies by $\sim 18\%$ from corner to corner, allowance for which *must* be made in photometry.

Dithering and mosaicing are complicated by the fact that an integral pixel shift near the center of the detector will translate into a non-integral displacement for pixels near the edges. This is not a fundamental limitation, but will imply some computational complexity in registering images and will depend on an accurate measurement of distortions.

Figure 1.1: ACS apertures compared with the V2/V3 reference frame. The read-out amplifiers (A,B,C,D) are indicated on the figure. The WFC data products from the calibration pipeline are oriented so that WFC1 (chip 1, which uses amps A,B) is on top. The HRC data products are also oriented such that amps A,B are on top, but they are inverted from the WFC images with respect to the sky.



1.2 Basic Instrument Operations

1.2.1 Target Acquisitions

For the majority of ACS observations target acquisition is simply a matter of defining the appropriate aperture for the observation. Once the telescope acquires its guide stars, the target will be within $\sim 1\text{--}2$ arcseconds of the specified pointing. For observations with the ramp filters, one must specify the desired central wavelength for the observation. For the special case of coronagraphic observations, an onboard target acquisition will need to be specified. The nominal accuracy of the combined target acquisition and slew procedure is ~ 0.03 arcseconds, comparable to that achieved by STIS.

1.2.2 Typical ACS Observing Sequence

An important issue for observers to consider is the “packaging” of their observations, i.e. how observations are CR-SPLIT to mitigate the impact of cosmic rays, whether sub-stepping or “dithering” of images is required, and how, if necessary, to construct a mosaic pattern to map the target. For an online library of pointing patterns, refer to the ACS dither web page:

<http://www.stsci.edu/hst/acs/proposing/dither>.

HRC observations and narrowband observations with the WFC are more likely to be read-noise limited, requiring consideration of the optimum CR-SPLIT times. Observations with the MAMA detectors do not suffer from cosmic rays or read noise, but long integration times are often needed to obtain sufficient signal-to-noise in the photon-starved ultraviolet.

A typical ACS observing sequence consists of a series of CR-SPLIT and dithered $\sim 10\text{--}20$ minute exposures for each program filter. Coronagraphic observations require an initial target acquisition observation to permit centering of the target under the occulting mask. Observers generally need not take their own calibration exposures.

1.2.3 Data Storage and Transfer

At the conclusion of each exposure, the science data are read out from the detector and placed in ACS's internal buffer memory, where they are stored until it can be transferred to the HST solid state data recorder (and thereafter to the ground). The internal buffer memory is large enough to hold one WFC image, or sixteen HRC or SBC images, and so the buffer will typically need to be dumped during the following WFC exposure, assuming it is longer than 340 seconds. For shorter exposures an extra overhead of this length is imposed.

ACS's internal buffer stores the data in a 16 bit-per-pixel format. This structure imposes a maximum of 65,535 counts per pixel. For the MAMA detectors this maximum is equivalent to a limit on the total number of detected photons per pixel which can be accumulated in a single exposure. For the WFC and HRC, the full well (and not the 16 bit buffer format) limits the photons per pixel which can be accumulated without saturating in a single exposure when $GAIN > 1$ and $GAIN > 2$, respectively, are selected.

1.2.4 Parallel Operations

Parallel observations with the WFC and HRC are possible with ACS for certain filter combinations. ACS can be used in parallel with any of the other science instruments on HST within restrictions described in detail in the *ACS Instrument Handbook*. There are significant constraints on the use of the MAMA detectors in parallel. The policy for applying for parallel observing time is described in the *HST Call for Proposals*.



CHAPTER 2:

ACS Data Structure

In this chapter. . .

2.1 Types of ACS Files / 2-2

2.2 ACS File Structure / 2-5

2.3 Data Storage Requirements / 2-7

2.4 Headers and Keywords / 2-9

This chapter describes the ACS data products, including a discussion of file suffixes, association tables, and trailer files. It describes the FITS file structure and data storage requirements for ACS images. Finally, a description of the FITS header keywords is provided for both primary and extension headers.

2.1 Types of ACS Files

2.1.1 Data Files and Suffixes

The file suffixes given to ACS data products are described in Table 2.1 and closely mimic the suffixes used by STIS. The initial input files to the calibration pipeline are the raw (RAW) files from Generic Conversion and the association (ASN) table, if applicable, for the complete observation set.

For CCD images, a temporary file, with the suffix BLV_TMP, is created once bias levels are subtracted and the overscan regions are trimmed. This file is renamed with the FLT extension after the standard calibrations (flat fielding, dark subtraction, etc.) are complete. The FLT files will serve as input for cosmic ray rejection (if required). For CR-SPLIT exposures, a temporary ‘*cr-combined image*’ (CRJ_TMP) is created and then renamed with the CRJ extension once basic calibrations are complete. Single MAMA images are given the FLT suffix once calibrations are complete. By definition, these images do not have an overscan region and are not affected by cosmic rays. The calibrated products of a REPEAT-OBS association will be several individually calibrated FLT exposures and a summed flat-fielded (but not cosmic-ray cleaned) SFL image.

Table 2.1: ACS File Suffixes:

File Suffix	Description	Units
_RAW	Raw data	<i>DN</i>
_ASN	Association file for observation set	
_SPT	Telemetry and engineering data	
_TRL	Trailer File with processing comments	
_BLV_TMP	Overscan-trimmed individual exposure (renamed to _FLT)	<i>DN</i>
_CRJ_TMP	Uncalibrated, CR-rejected Combined image (renamed to _CRJ)	<i>DN</i>
_FLT	Calibrated, Flat fielded individual exposure	<i>electrons</i>
_CRJ	Calibrated, CR-rejected, Combined image	<i>electrons</i>
_SFL	Calibrated, Repeat-Obs, Combined image	<i>electrons</i>
_DRZ	Calibrated, Geometrically Corrected, Dither-Combined image	<i>electrons/sec</i>

2.1.2 Association Tables

Association tables are useful for keeping track of the complex set of relationships that can exist between exposures taken with ACS, especially with REPEAT-OBS, CR-SPLIT, and dithered exposures. Images taken at a given dither position may be additionally CR-SPLIT into multiple

exposures. In these cases, associations are built to describe how each exposure relates to the desired final product. As a result, ACS associations will create one or more science products from the input exposures, unlike NICMOS or STIS associations. The relationships defined in the association table determine how far through the calibration pipeline the exposures are processed and when the calibrated exposures are combined into sub-products for further calibration.

ACS data files are given the following definitions:

- An *exposure* is a single image, the “atomic unit” of HST data.
- A *dataset* is a collection of files having a common root name (first 9 characters).
- A *sub-product* is a dataset created by combining a subset of the exposures in an association.
- A *product* is a dataset created by combining sub-products of an association.

ACS association tables were designed to closely resemble the NICMOS association format, with three primary columns: MEMNAME, MEMTYPE, and MEMPRSNT. The column MEMNAME gives the name of each exposure making up the association and output product name(s). The column MEMTYPE specifies the role the file has in the association. A unique set of MEMTYPES specific to ACS were adopted to provide the support for multiple products. These types are summarized in Table 2.2.

Table 2.2: Exposure types in ACS associations. The suffix “n” is appended to the MEMTYPE when multiple sets are present within a single association.

MEMTYPE	Description
EXP-CRJ	Input CR-SPLIT exposure (single set)
EXP-CR n	Input CR-SPLIT exposure for CR-combined image n (multiple sets)
PROD-CRJ	CR-combined output product (single set)
PROD-CR n	CR-combined output product n (multiple sets)
EXP-RPT	Input REPEAT-OBS exposure (single set)
EXP-RP n	Input REPEAT-OBS exposure for repeated image n (multiple sets)
PROD-RPT	REPEAT-OBS combined output product (single set)
PROD-RP n	REPEAT-OBS combined output product n (multiple sets)
EXP-DTH	Input dither exposure
PROD-DTH	Dither-combined output product

A sample association table for a two-position dithered observation with `CR-SPLIT=2` is presented in Table 2.3. This example shows how both `MEMNAME` and `MEMTYPE` are used to associate input and output products. The `MEMTYPE` for each component of the first `CR-SPLIT` exposure, `JxxxxxECM` and `JxxxxxEGM`, are given the type `EXP-CR1`. The sub-product `Jxxxxx011` is designated in the table with a `MEMTYPE` of `PROD-CR1`. The last digit of the product filename corresponds to the output product number in the `MEMTYPE`. A designation of zero for the last digit in the filename is reserved for the dither-combined product.

The column `MEMPRSNT` indicates whether a given file already exists. For example, if cosmic ray rejection has not yet been performed by `CALACS`, the `PROD-CRn` files will have a `MEMPRSNT` value of "no". The sample association table in Table 2.3 shows the values of `MEMPRSNT` prior to `CALACS` processing.

Table 2.3: Sample Association Table `Jxxxxx010_ASN`

<code>MEMNAME</code>	<code>MEMTYPE</code>	<code>MEMPRSNT</code>
<code>JxxxxxECM</code>	<code>EXP-CR1</code>	yes
<code>JxxxxxEGM</code>	<code>EXP-CR1</code>	yes
<code>Jxxxxx011</code>	<code>PROD-CR1</code>	no
<code>JxxxxxEMM</code>	<code>EXP-CR2</code>	yes
<code>JxxxxxEOM</code>	<code>EXP-CR2</code>	yes
<code>Jxxxxx012</code>	<code>PROD-CR2</code>	no
<code>Jxxxxx010</code>	<code>PROD-DTH</code>	no

2.1.3 Trailer Files

Each task in the `CALACS` package creates messages during processing which describe the progress of the calibration and which are sent to `STDOUT`. In the pipeline processing for other HST instruments, trailer files were created by simply redirecting the `STDOUT` to a file. Because multiple output files can be processed in a single run of `CALACS`, creating trailer files presents a unique challenge. Each task within the `CALACS` package must decide which trailer file should be appended with comments and automatically open, populate, and close each trailer file.

`CALACS` will *always overwrite* information in trailer files from previous runs of `CALACS` while preserving any comments generated by Generic Conversion. This ensures that the trailer files accurately reflect the most recent processing performed. The string `CALACSBEG` will mark the first comment added to the trailer file. If a trailer file already exists, `CALACS` will search for this string to determine where to append

processing comments. If it is not found, the string will be written to the end of the file and all comments will follow. Thus any comments from previous processing are overwritten and only the most current calibrations are recorded.

As each image is processed, an accompanying trailer file with the `.trl` ending will be created. Further processing will concatenate all trailer files associated with the output product into a single file. Additional messages will then be appended to this concatenated file. Thus, some information is duplicated across multiple trailer files, but for any product processed within the pipeline, the trailer file is ensured to contain processing comments from each input file.

Linking trailer files together can result in multiple occurrences of the *CALACSBEG* string. Only the first, however, determines where **CALACS** will begin overwriting comments if an observation is reprocessed.

2.2 ACS File Structure

The ACS calibration pipeline assembles data received from HST into datasets, applies the standard calibrations, and stores the uncalibrated datasets in the HST Data Archive. The structure of these data products is based on the STIS and NICMOS file format and consists of multi-extension FITS files which store science (SCI), data quality (DQ) and error (ERR) arrays, as shown in Figure 2.1.

ACS WFC data come from two CCD chips and are treated as separate observations, with a SCI, DQ and ERR array for each chip, and with both chips being stored in the same FITS file. The result is a FITS file with 6 data extensions plus a global header for a single WFC exposure. The WFC apertures are plotted with respect to the V2/V3 reference frame in Figure 2.2 and are oriented such that the x-axis is approximately to the right and the y-axis is approximately straight up. For pipeline data products, chip 2 is below chip 1 in y-pixel coordinates and was therefore defined as extension 1. Thus, the chip/extension notation is counterintuitive. To display the science image for chip 1, the user must specify the extension `'file.fits[sci,2]'`. Similarly, the data quality and error array for chip 1 are specified as `[dq,2]` and `[err,2]`.

A single HRC or SBC exposure comes from a single chip and has only 3 data extensions plus a global header. While the raw and calibrated WFC images contain 6 data extensions, the drizzled product will always contain 3 data extensions, no matter which detector was used. When WFC data is drizzled, both chips are included in a single FITS extension.

Figure 2.1 illustrates the basic format for storing ACS images. An uncalibrated (RAW) exposure contains a primary header plus, for each chip, a SCI extension (in 16-bit integer format), an empty ERR extension, and an empty DQ array. The calibrated product from **CALACS** contains a

primary header plus, for each chip, a SCI extension (in 32-bit float format), an ERR extension (32-bit float), and a DQ extension (16-bit integer).

PyDrizzle removes the effects of the geometric distortion and produces a drizzled (DRZ) product which has the multi-extension FITS file shown in Figure 2.1. The SCI extension contains the distortion-corrected data (32-bit float), the WHT extension contains the weight mask (32-bit float), and the CTX extension contains the context image (32-bit integer). For more information on these DRZ product extensions, refer to "Data Products" in Section 4.5.

Figure 2.1: Data format for calibrated and drizzled ACS modes. Note that for calibrated science data, WFC1 (chip 1) corresponds to extension [sci,2].

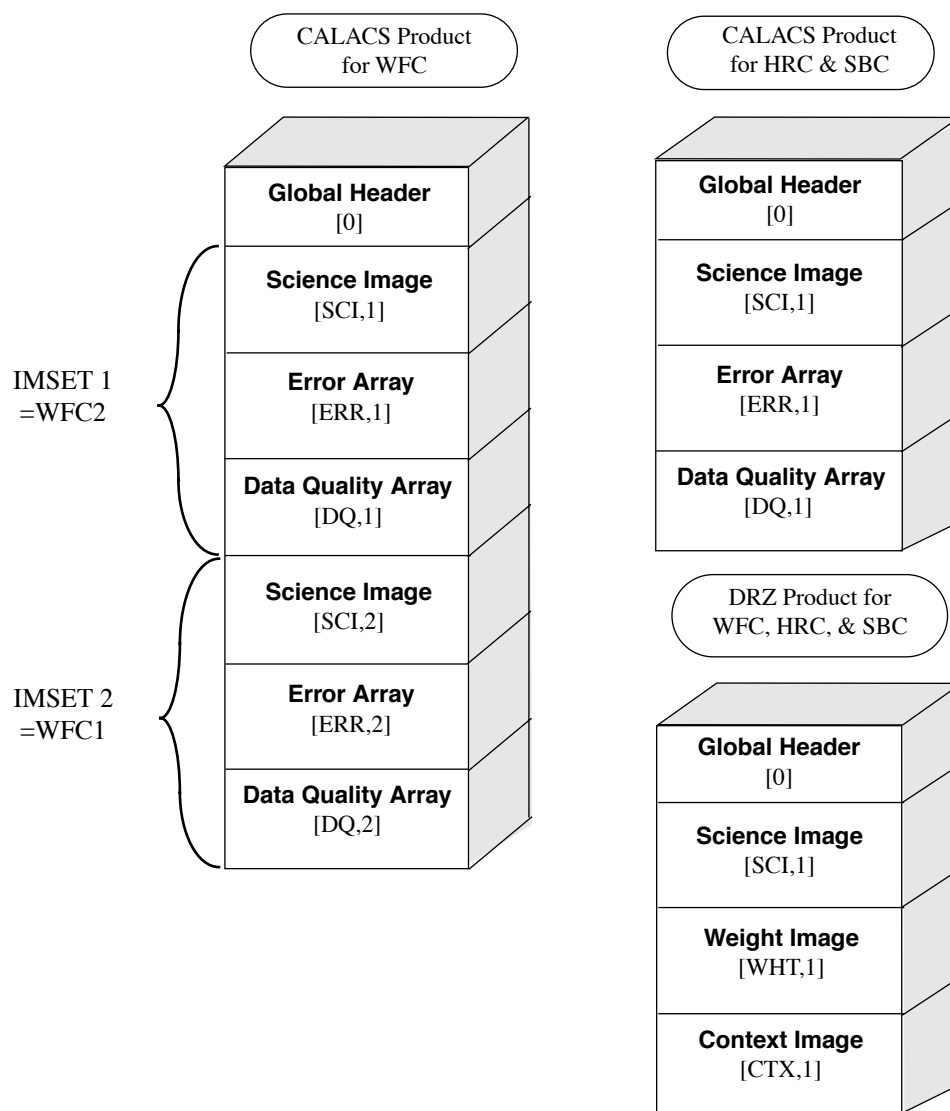
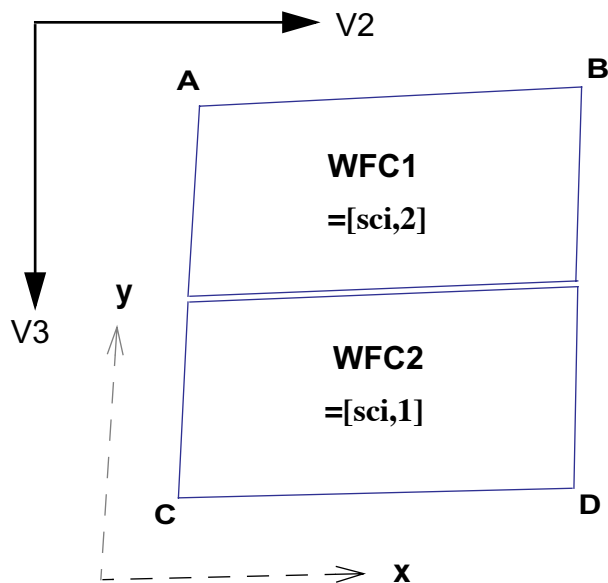


Figure 2.2: WFC apertures compared with the V2/V3 reference frame. The read-out amplifiers (A,B,C,D) are indicated on the figure. The WFC data products from the calibration pipeline are oriented so that WFC1 (chip 1, which uses amps A,B) is on top. The science image for chip 1 may be specified as [sci,2].



2.3 Data Storage Requirements

Users are reminded to consider the large size of WFC exposures when allocating disk space for storing and reprocessing ACS data. Raw images serve as input to the calibration pipeline and have the file sizes (in MB) given in Table 2.4. The WFC image includes two SCI arrays, while the HRC and SBC detectors have a single array. The raw image sizes presume that both the SCI and DQ arrays are populated with short integer values, but that the ERR arrays are NULL (all pixels have a value of zero).

During **CALACS** processing, the SCI arrays are converted from integer to floating point values. The null ERR array is populated with floating point values, and the null DQ array is populated with integer values. As a result, the size of calibrated images (in MB) is much larger. The image size in pixels is given in Table 2.5. Calibrated images taken with the WFC and HRC detectors are smaller than raw images because the overscan regions have been trimmed during processing.

PyDrizzle combines the science, data quality, and error arrays for each detector/array into a single science, weight, and context extension. Once the distortion is corrected, the size of the drizzled images (in pixels and therefore also in MB) will be larger than the calibrated images due to the fact that their projection on the sky is rhombus-shaped, rather than square. The specific dimensions of the drizzled image will depend on which

distortion model is currently in use in the pipeline and will vary slightly (~1-2 pixels) due to the effects of velocity aberration.

Table 2.4: Size (in MB) of raw, calibrated, and drizzled ACS images for each detector. The drizzled image sizes are distortion-corrected only and assume no dither offset or scale change.

Detector	Size of FITS file (S_{raw})	Size of FITS file (S_{cal})	Size of FITS file (S_{drz})
WFC (2 Chips)	68.7 MB	168 MB	215 MB
HRC (1 Chip)	4.4 MB	10.5 MB	16 MB
SBC (1 Chip)	4.2 MB	10.5 MB	16 MB

Table 2.5: Size (in pixels) for raw, calibrated, and drizzled ACS images for each detector. The drizzled image sizes are distortion-corrected only and assume no dither offset or scale change. The size of drizzled images will vary slightly due to velocity aberration effects.

Detector	X_{raw}	Y_{raw}	X_{cal}	Y_{cal}	X_{drz}	Y_{drz}
WFC (2 Chips)	4144	2068	4096	2048	4213	4240
HRC (1 Chip)	1062	1044	1024	1024	1164	1138
SBC (1 Chip)	1024	1024	1024	1024	1397	1341

While the size of calibrated, drizzled HRC and SBC images is comparable to that of STIS or WFPC2, the ACS WFC images are over 16 times as large. The following equation can be used to estimate the minimum amount of free storage required for processing associated ACS data:

$$D_{min} = n \cdot S_{raw} + (1 + n) \cdot S_{cal} + (2.1 + p^2) \cdot S_{drz}$$

where:

- D_{min} is the minimum free disk space required for processing,
- n is the number of exposures in each CR-SPLIT set or REPEAT-OBS set,
- S_{raw} is the size of the raw exposure (from Table 2.4),
- S_{cal} is the size of the calibrated exposure (from Table 2.4),
- S_{drz} is the size of the distortion-corrected exposure (from Table 2.4),
- p is the percentage shift (in pixels) across all dither positions

2.4 Headers and Keywords

While the ACS image headers for the WFC, HRC, HRC-ACQ, and SBC observations have very similar formats, two main differences are apparent. The MAMA (SBC) headers require a slightly modified structure from the CCD (WFC, HRC) headers, and the HRC-ACQ headers require several unique keywords.

In Table 2.6 and Table 2.7, we describe the ACS header keywords, with a note indicating to which detector those keywords refer. The primary header keywords (FITS extension 0) are given in Table 2.6, and the extension header keywords (FITS extensions > 0) are given in Table 2.7. For more information about primary and extension FITS headers, refer to Section 2.1.1 of the HST Data Handbook Introduction.

Table 2.6: ACS Primary Header Keywords (FITS extension 0)

KEYWORD	DESCRIPTION
SIMPLE	data conform to FITS standard (T/F)
BITPIX	bits per data value
NAXIS	number of data axes
EXTEND	file may contain standard extensions (T/F)
NEXTEND	number of standard extensions
GROUPS	image is in group format? (T/F)
DATE	date this file was written (yyyy-mm-dd)
FILENAME	name of file
FILETYPE	type of data found in data file
TELESCOP	telescope used to acquire data
INSTRUME	identifier for instrument used to acquire data
EQUINOX	equinox of celestial coordinate system
DATA DESCRIPTION KEYWORDS (All Detectors)	
ROOTNAME	rootname of the observation set
IMAGETYP	type of exposure identifier
PRIMESI	instrument designated as prime
TARGET INFORMATION (All Detectors)	
TARGNAME	proposer's target name
RA_TARG	right ascension of the target (deg) (J2000)
DEC_TARG	declination of the target (deg) (J2000)
PROPOSAL INFORMATION (All Detectors)	
PROPOSID	PEP proposal identifier
LINENUM	proposal log line number
PR_INV_L	last name of principal investigator
PR_INV_F	first name of principal investigator

EXPOSURE INFORMATION (All Detectors)	
SUNANGLE	angle between sun and V1 axis
MOONANGL	angle between moon and V1 axis
SUN_ALT	altitude of the sun above Earth's limb
FGSLOCK	commanded FGS lock (fine, coarse, gyros, unknown)
DATE-OBS	UT date of start of observation (yyyy-mm-dd)
TIME-OBS	UT time of start of observation (hh:mm:ss)
EXPSTART	exposure start time (modified Julian date)
EXPEND	exposure end time (modified Julian date)
EXPTIME	exposure duration (seconds)--calculated
EXPFLAG	exposure interruption indicator
POINTING INFORMATION (All Detectors)	
PA_V3	position angle of V3-axis of HST (deg)
TARGET OFFSETS (POSTARGS) (All Detectors)	
POSTARG1	POSTARG in axis 1 direction
POSTARG2	POSTARG in axis 2 direction
DIAGNOSTIC KEYWORDS (All Detectors)	
OPUS_VER	OPUS software system version number
CAL_VER	CALACS code version
PROCTIME	pipeline processing time (MJD)
SCIENCE INSTRUMENT CONFIGURATION (All Detectors)	
OBSTYPE	observation type (imaging, spectroscopic, coronagraphic)
OBSMODE	operating mode
CTEIMAGE	type of charge transfer image, if applicable
SCLAMP	lamp status: NONE or name of lamp which is on
NRPTXP	number of repeat exposures in set (default=1)
SUBARRAY	data from a subarray (T) or full frame (F)
DETECTOR	detector in use: WFC, HRC or SBC
FILTER1	element selected from filter wheel 1
FILTER2	element selected from filter wheel 2
FWOFFSET	computed filter wheel offset (0, +1, -1)
FWERROR	filter wheel position error flag (T, F)
LRFWAVE	proposed linear ramp filter wavelength
APERTURE	aperture name
PROPAPER	proposed aperture name
CRSPLIT	number of cosmic ray split exposures
DIRIMAGE	direct image for grism or prism exposure
MAMA OFFSETS (SBC Only)	
MOFFSET1	axis 1 MAMA offset
MOFFSET2	axis 2 MAMA offset

LOCAL RATE CHECK IMAGE (SBC Only)	
LRC_XSTS	local rate check image exists (T/F)
LRC_FAIL	local rate check failed (T/F)
CALIBRATION SWITCHES: PERFORM, OMIT (All Detectors)	
STATFLAG	calculate statistics? (T/F)
WRERR	write out error array extension? (T/F)
DQICORR	data quality initialization?
ATODCORR	correct for A to D conversion errors?
BLEVCORR	subtract bias level computed from overscan image?
BIASCORR	subtract bias image?
FLSHCORR	subtract post flash image?
CRCORR	combine observations to reject cosmic-rays?
EXPSCORR	process individual observations after cr-reject?
SHADCORR	apply shutter shading correction?
GLINCORR	correct for global detector non-linearities? (SBC only)
LFLGCORR	flag pixels for local and global nonlinearities? (SBC only)
DARKCORR	subtract dark image?
FLATCORR	apply flat field correction?
PHOTCORR	populate photometric header keywords?
RPTCORR	add individual repeat observations?
DRIZCORR	process dithered images?
CALIBRATION REFERENCE FILES (All Detectors)	
BPIXTAB	bad pixel table
CCDTAB	CCD calibration parameter table
ATODTAB	analog-to-digital correction file
OSCNTAB	CCD overscan table
BIASFILE	bias image file name
FLSHFILE	post flash image file name
CRREJTAB	cosmic-ray rejection parameter table
SHADFILE	shutter shading correction file
MLINTAB	MAMA linearity correction table (SBC only)
DARKFILE	dark image file name
PFLTFILE	pixel-to-pixel flat field file name
DFLTFILE	delta-flat field file name
LFLTFILE	low order flat
PHOTTAB	photometric throughput table (not used)
GRAPHTAB	HST graph table
COMPTAB	HST components table
IDCTAB	image distortion correction table

COSMIC RAY REJECTION ALGORITHM PARAMETERS (WFC, HRC, HRC ACQ)	
MEANEXP	reference exposure time
SCALENSE	multiplicative scale factor applied to noise
INITGUES	initial guess method (min, med)
SKYSUB	sky value subtracted (mode, none)
SKYSUM	sky level from the sum of all constituent images
CRSIGMAS	statistical rejection criteria
CRRADIUS	rejection propagation radius (pixels)
CRTHRESH	rejection propagation threshold
BADINPDQ	data quality flag used for rejection
REJ_RATE	rate at which pixels are affected by cosmic rays
CRMASK	flag CR-rejected pixels in input files? (T/F)
PATTERN KEYWORDS (WFC, HRC, SBC)	
PATTERN1	primary pattern type
P1_SHAPE	primary pattern shape
P1_PURPS	primary pattern purpose
P1_NPTS	number of points in primary pattern
P1_PSPAC	point spacing for primary pattern (arcsec)
P1_LSPAC	line spacing for primary pattern (arcsec)
P1_ANGLE	angle between sides of parallelogram patt (deg)
P1_FRAME	coordinate frame of primary pattern
P1_ORINT	orientation of pattern to coordinate frame (deg)
P1_CENTR	center pattern relative to pointing (yes/no)
PATTSTEP	position number of this point in the pattern
POST FLASH PARAMETERS (WFC, HRC, HRC ACQ)	
FLASHDUR	exposure time in seconds: 0.1 to 409.5
FLASHCUR	post-flash current (off, low, med, high)
FLASHSTA	status (successful, aborted, not performed)
SHUTRPOS	shutter position (A or B)
ENGINEERING PARAMETERS (WFC, HRC, HRC ACQ)	
CCDAMP	CCD amplifier read out configuration
CCDGAIN	commanded gain of CCD (electrons/DN)
CCDOFSTA	commanded CCD bias offset for amplifier A (electrons)
CCDOFSTB	commanded CCD bias offset for amplifier B (electrons)
CCDOFSTC	commanded CCD bias offset for amplifier C (electrons)
CCDOFSTD	commanded CCD bias offset for amplifier D (electrons)

CALIBRATED ENGINEERING PARAMETERS (WFC, HRC, HRC ACQ)	
ATODGNA	calibrated gain for CCD amplifier A (electrons/DN)
ATODGNB	calibrated gain for CCD amplifier B (electrons/DN)
ATODGNC	calibrated gain for CCD amplifier C (electrons/DN)
ATODGND	calibrated gain for CCD amplifier D (electrons/DN)
READNSEA	calibrated read noise for amplifier A (electrons)
READNSEB	calibrated read noise for amplifier B (electrons)
READNSEC	calibrated read noise for amplifier C (electrons)
READNSED	calibrated read noise for amplifier D (electrons)
BIASLEVA	bias level for amplifier A (electrons)
BIASLEVB	bias level for amplifier B (electrons)
BIASLEVC	bias level for amplifier C (electrons)
BIASLEVD	bias level for amplifier D (electrons)
TARGET ACQUISITION PARAMETERS (HRC only)	
ACQTYPE	type of acquisition
ACQNAME	rootname of acquisition exposure
TARGET ACQUISITION PARAMETERS (HRC ACQ only)	
ACQTYPE	type of acquisition
CENTMETH	target acquisition centering method
CRELIM	perform cosmic ray rejection in acquisition
CCDBIASS	CCD bias subtracted from target acq image? (y/n)
BIASLEV	CCD bias level used to process acquisition exp.
CHECKBOX	size of checkbox for finding algorithms
TARGAREA	area of target (detector pixels)
ASSOCIATION KEYWORDS (All Detectors)	
ASN_ID	unique identifier assigned to association
ASN_TAB	name of association table
ASN_MTYP	role of exposure in association

Table 2.7: ACS Extension Header Keywords (FITS extensions > 0)

KEYWORD	DESCRIPTION
XTENSION	extension type
BITPIX	bits per data value
NAXIS	number of data axes
NAXIS1	length of first data axis
NAXIS2	length of second data axis
PCOUNT	number of group parameters
GCOUNT	number of groups
INHERIT	inherit the primary header
EXTNAME	extension name
EXTVER	extension version number
ROOTNAME	rootname of the observation set
EXPNAME	9 character exposure identifier
DATAMIN	minimum data value (electrons)
DATAMAX	maximum data value (electrons)
BUNIT	brightness units (counts, electrons, electrons/sec)
BSCALE	scale factor for array value to physical value
BZERO	physical value for an array value of zero
WORLD COORDINATE SYSTEM AND RELATED PARAMETERS (All Detectors)	
CRPIX1	x-coordinate of reference pixel
CRPIX2	y-coordinate of reference pixel
CRVAL1	first axis value at reference pixel
CRVAL2	second axis value at reference pixel
CTYPE1	the coordinate type for the first axis
CTYPE2	the coordinate type for the second axis
CD1_1	partial of first axis coordinate w.r.t. x
CD1_2	partial of first axis coordinate w.r.t. y
CD2_1	partial of second axis coordinate w.r.t. x
CD2_2	partial of second axis coordinate w.r.t. y
LTV1	offset in X to subsection start
LTV2	offset in Y to subsection start
LTM1_1	reciprocal of sampling rate in X
LTM2_2	reciprocal of sampling rate in Y
RA_APER	RA of aperture reference position
DEC_APER	Dec of aperture reference position
PA_APER	position angle of aperture reference position (deg)
ORIENTAT	position angle of image y axis (degrees E of N)
VAFactor	velocity aberration plate scale factor

WFC CCD CHIP IDENTIFICATION (WFC only)	
CCDCHIP	CCD chip (1 or 2)
READOUT DEFINITION PARAMETERS (All Detectors)	
CENTERA1	subarray axis1 center point in unbinned detector pixels
CENTERA2	subarray axis2 center point in unbinned detector pixels
SIZAXIS1	subarray axis1 size in unbinned detector pixels
SIZAXIS2	subarray axis2 size in unbinned detector pixels
BINAXIS1	axis1 data bin size in unbinned detector pixels
BINAXIS2	axis2 data bin size in unbinned detector pixels
PHOTOMETRY KEYWORDS (All Detectors)	
PHOTMODE	observation configuration for photometric calibration
PHOTFLAM	inverse sensitivity ($\text{erg s}^{-1} \text{cm}^{-2} \text{A}^{-1}$ for 1 electron/sec)
PHOTZPT	ST magnitude zero point
PHOTPLAM	pivot wavelength
PHOTBW	rms bandwidth of filter plus detector
REPEATED EXPOSURES INFO (WFC, HRC, HRC ACQ)	
NCOMBINE	number of image sets combined during CR rejection
DATA PACKET INFORMATION (All Detectors)	
FILLCNT	number of segments containing fill
ERRCNT	number of segments containing errors
PODPSFF	podps fill present? (T/F)
STDCFFF	ST DDF fill present? (T/F)
STDCFFP	ST DDF fill pattern (hex)
ON-BOARD COMPRESSION INFORMATION (WFC Only)	
WFCMPRSD	WFC data compressed? (T/F)
COMPTYP	compression type performed? (partial/full)
CBLKSIZ	size of compression block in bytes
LOSTPIX	number of pixels lost due to buffer overflow
TARGET ACQUISITION PARAMETERS (HRC ACQ Only)	
TARGA1	fsw located subarray axis1 coord. of target
TARGA2	fsw located subarray axis2 coord. of target
APERA1	fsw located subarray axis 1 coord. of aperture
APERA2	fsw located subarray axis 2 coord. of aperture
APERLKA1	axis 1 detector pixel of acq aperture center
APERLKA2	axis 2 detector pixel of acq aperture center
MAXCHCNT	counts in the brightest checkbox
BOPOFFA1	axis 1 offset (arcs) object moved off aperture
BOPOFFA2	axis 2 offset (arcs) object moved off aperture

ENGINEERING PARAMETERS (SBC Only)	
GLOBRATE	global count rate
GLOBLIM	was global linearity level exceeded?
IMAGE STATISTICS AND DATA QUALITY FLAGS (All Detectors)	
NGOODPIX	number of good pixels
SDQFLAGS	serious data quality flags considered 'bad' by CALACS
GOODMIN	minimum value of good pixels (electrons)
GOODMAX	maximum value of good pixels (electrons)
GOODMEAN	mean value of good pixels (electrons)
SNRMIN	minimum signal to noise of good pixels
SNRMAX	maximum signal to noise of good pixels
SNRMEAN	mean signal to noise of good pixels
SOFTERRS	number of soft error pixels (DQF=1)
MEANBLEV	average of the subtracted bias levels (electrons)
MEANDARK	average of the subtracted dark values (electrons)
MEANFLSH	average of the subtracted post flash values (electrons)

ACS Calibration Pipeline

In this chapter. . .

- | |
|--|
| <ul style="list-style-type: none">3.1 On The Fly Reprocessing (OTFR) / 3-23.2 Pipeline Overview / 3-33.3 Structure of CALACS / 3-63.4 CALACS Processing Steps / 3-93.5 Manual Recalibration of ACS Data / 3-25 |
|--|

This chapter describes the ACS calibration pipeline and the On-the-Fly-Reprocessing, which is standard for all HST data requested from the STScI archive. The pipeline software consists of two separate packages: **CALACS** and **PyDrizzle**. **CALACS** removes various instrumental signatures, combines multiple exposures, and updates header keyword values. **PyDrizzle** corrects images for geometric distortion and combines observations which were taken as part of a dither pattern.

The data products from OTFR are described, as well as considerations and requirements for manual recalibration. The remaining bulk of the chapter is devoted to a detailed description of **CALACS**. While **PyDrizzle** is also used during pipeline processing, a detailed discussion of this software is deferred to Chapter 4.

3.1 On The Fly Reprocessing (OTFR)

OTFR is now the standard way of processing data which has been requested by the user from the STScI archive. It provides the best calibrated products by reprocessing the raw telemetry files “on-the-fly” for distribution each time data is requested. Each observation requires calibrations suited to the particular mode used for taking data. Calibration data specific to the various modes of ACS operation are prepared and archived in the Calibration Data Base System (CDBS).

The STScI calibration pipeline consists of two main software systems: the Operations Pipeline Unified System (OPUS) and the Data Archive and Distribution System (DADS). Raw spacecraft telemetry data from Goddard Spaceflight Center are transmitted to STScI in the form of POD files. When a user requests data from the HST Archive, OTFR uses the POD files as input to the OPUS step called Generic Conversion and generates the uncalibrated “raw” data. At the same time, Generic Conversion queries the CDBS to determine the most up-to-date reference files for the observation configuration. OPUS then runs **CALACS** to process the uncalibrated data, using specific ACS reference images and tables from the CDBS. With these data, DADS populates a database which is accessible to users via StarView. DADS then distributes any data (both calibrated and uncalibrated) requested for download to the user. (See Chapter 1 of the HST Introduction for more information.)

The most current ACS reference files are used by **CALACS** each time OTFR is run. The calibration reference files (e.g. flat fields, bad pixel tables) are also available from the HST Archive. Since reference files such as CCD biases and darks are frequently updated, OTFR may use different reference files depending on the date of reprocessing. Previously, the archived data from DADS had to be manually recalibrated by running the calibration software on the user’s home workstation with the updated reference files. OTFR replaces this step by automatically recalibrating with the most current reference data available. The user simply waits until the contemporaneous reference files are in place and requests the data via StarView or the HST web-based archive request form. The image header keywords are then updated with the names of the reference files used during that OTFR run. The PROCTIME keyword is updated to reflect the pipeline processing date and time in MJD.

OTFR allows the user to avoid downloading archived data which is outdated because of software changes made for bug fixes, improved algorithms, new capabilities, or header keyword changes. An archive request for calibrated data will result in OTFR reprocessing and distributing data which uses the latest software versions available.

Currently OTFR will distribute *all* files associated with an ACS observation, including both the uncalibrated and calibrated files. Future

versions could enable users to select only certain parts of the dataset, for example the final calibrated image. Future versions could also enable users to set calibration parameters for a particular OTFR run. Until then, OTFR will process with default values. Of course, the option of recalibrating aside from OTFR always exists. (See “Manual Recalibration of ACS Data” in Section 3.5.)

3.2 Pipeline Overview

Pipeline processing is carried out by 2 separate packages: **CALACS**, which corrects for instrumental effects and produces calibrated products, and **PyDrizzle**, which corrects for distortion and combines associated dithered data.

3.2.1 CALACS: Image Calibration

The **CALACS** software consists of a series of individual tasks that:

- Orchestrate the flow of data through the pipeline.
- Perform the initial tasks of basic two-dimensional image reduction (e.g., overscan subtraction, bias subtraction) for CCD data.
- Reject cosmic rays from CR-SPLIT CCD data.
- Perform the remaining tasks of basic two-dimensional image reduction (e.g., dark subtraction, flat fielding) for CCD and MAMA data.
- Sum any REPEAT-OBS exposures.

The calibrated products from the pipeline may still contain artifacts such as hot pixels and cosmic-rays. If hot pixels are to be detected and flagged by **CALACS**, dithered exposures are required. Similarly, cosmic ray rejection by the pipeline requires the specification of CR-SPLIT exposures in the observing program. In the future, improved pipeline software will allow all associated data (including REPEAT-OBS exposures and dithered exposures) to be automatically corrected for cosmic-rays.

While intermediate steps in **CALACS** make use of sky subtraction, for example for identifying cosmic-rays, all data products created by the pipeline will *not* be sky subtracted. Fully calibrated data products (with suffixes FLT, CRJ, SFL) will be in units of *electrons*.

CALACS and Single exposures

A single raw ACS exposure is processed by **CALACS** following the steps outlined in Section 3.3. These steps include the standard detector calibrations: bias subtraction, dark subtraction, flat fielding, etc. When

CALACS processes a single raw exposure, the calibrated product is given the FLT suffix. The data in the calibrated SCI and ERR extensions are in units of *electrons*.

CALACS and Multiple exposures

The same processing steps performed on single images are performed for multiple images which are part of an association. The result is a single calibrated product in units of *electrons*.

CALACS will recognize and correctly process CR-SPLIT or REPEAT-OBS exposures by interpreting the association table and determining which exposures should be combined during calibration. REPEAT-OBS exposures are individually run through all the calibration steps to create calibrated, flat-fielded (FLT) product for each input file. The FLT images are then summed to create the summed flat-fielded (SFL) product. No cosmic-ray rejection is performed for REPEAT-OBS images.

Observations taken as part of a CR-SPLIT association, on the other hand, are combined during cosmic-ray rejection. First, the bias and dark subtraction is performed. Then, the images are combined while at the same time rejecting cosmic rays. The product is then flat fielded to create a single calibrated, cosmic-ray rejected, flat-fielded (CRJ) image.

CALACS and Dithered exposures

Observations which use the dither patterns provided in the proposal instructions will be automatically associated for combining in the pipeline. Any pattern designed with POS-TARGs will not be associated. When processing observations which are part of a dither pattern, **CALACS** will produce a calibrated FLT, CRJ, or SFL file at each dither position. It will not, however, combine images from multiple positions. Further pipeline processing by **PyDrizzle** will correct for geometric distortion and combine dithered images.

Table 3.1: The input and output image suffixes from **CALACS** and **PyDrizzle** are given for various observing modes.

Image Type	CALACS Input	CALACS Output	PyDrizzle Input	PyDrizzle Output	Cosmic Ray Rejected?
Single	RAW	FLT	FLT	DRZ	No
RPT-OBS	ASN+RAW	FLT+SFL	ASN+SFL	DRZ	No
CR-SPLIT	ASN+RAW	FLT+CRJ	ASN+CRJ	DRZ	Yes
DITH-PATTERN	ASN+RAW	FLT	ASN+FLT	DRZ	No

3.2.2 PyDrizzle: Distortion Correction and Dither Combination

All ACS data will be automatically corrected for distortion during pipeline processing using a task called **PyDrizzle**. This task relies on the IDCTAB reference file for a description of the distortion model. **PyDrizzle** understands the ACS association tables which allow the pipeline to combine dithered observations. Only observations which use the dither patterns provided in the proposal instructions will be automatically associated for combining in the pipeline. Programs which rely on explicit POS-TARG commands will NOT be associated, resulting in separately calibrated images for each position. **PyDrizzle** will automatically produce images which are both astrometrically and photometrically accurate regardless of whether they were taken as a single exposure or a set of dithered exposures.

The flat-fielded, calibrated products from **CALACS** are listed in Table 3.1. The calibrated product of a single ACS exposure is an FLT file. The product of a CR-SPLIT or REPEAT-OBS association is a CRJ or SFL file, respectively. The product of a dither pattern is an FLT, CRJ or SFL file at *each* dither position. Using these files as input, **PyDrizzle** performs the geometric correction on *all* data (dithered or not) and combines multiple dithered images into a single output image with the DRZ suffix. For WFC observations, both chips are combined into a single extension. **PyDrizzle** then converts the data to units of count rate (*electrons/sec*).

It is important to note that **PyDrizzle**, when combining dithered images, does *not* remove cosmic-rays. This can be done offline with a new task called **MultiDrizzle** using the individual FLT files as input. Subsequent reprocessing with **PyDrizzle** or **MultiDrizzle** offline may be required to obtain the desired scientific result. For information on obtaining this software, we refer the reader to Section 3.5.1.

3.2.3 When is OTFR *not* Appropriate?

The goal of the ACS pipeline is to provide data calibrated to a level suitable for initial evaluation and analysis for all users. Observers frequently require a detailed understanding of the calibrations applied to their data and the ability to repeat, often with improved products, the calibration process at their home institution. There are several occasions when OTFR is not ideal and when off-line interactive processing by the user is required:

- when running **CALACS** with personal versions of reference files,
- when running **CALACS** with non-default calibration switch values,
- when images must be cleaned of artifacts such as new hot pixels or cosmic-rays.

Many ACS datasets do not contain multiple exposures at the same pointing and hence will not be cleaned of cosmic rays by OTFR

processing. In addition, the increasing prevalence of hot pixels, particularly in the WFC mode, makes dithered observations highly desirable. In these cases the pipeline will provide products which are geometrically corrected but which still contain cosmic rays.

To address this problem, the **MultiDrizzle** script has been developed to be run interactively outside the pipeline. **MultiDrizzle** provides a single-step interface to the complex suite of tasks in the STSDAS **dither** package. It makes use of the **PyDrizzle** software and can only be executed in the PyRAF environment. For more information, we refer the reader to Chapter 4.

3.3 Structure of CALACS

The **CALACS** package consists of four individual tasks listed in Table 3.2. These tasks are called automatically by **CALACS**, but each may be run separately should some variation in the normal processing be desired by the user.

Table 3.2: Tasks in the **CALACS** pipeline

ACSCCD	CCD specific calibrations
ACSREJ	Cosmic-ray rejection task
ACS2D	Basic MAMA and CCD calibrations
ACSSUM	Repeat-obs summing task

The flow of data through the ACS calibration pipeline and the decisions made while working with associated data are diagrammed in Figure 3.1. These calibration steps are also outlined below:

- Calculate a noise model for each pixel and record in the error array
- Flag known bad pixels and saturated pixels in the data quality array
- Subtract bias-level determined from overscan regions (CCD only)
- Subtract bias image (CCD only)
- Subtract post-flash image, if required (CCD only)
- Perform cosmic-ray rejection and combining of **CR-SPLIT** data (CCD only)
- Perform global linearity corrections (MAMA only)
- Scale and subtract dark image and calculate mean dark value
- Divide by flat field and multiply by gain
- Calculate photometry header keywords for flux conversion
- Calculate image statistics

Additional calibration steps are performed by **PyDrizzle**, including:

- Correct the geometric distortion of non-dithered images
- Convert the data units from counts to count rates
- Combine calibrated dithered observations into a single product

As indicated in Figure 3.1, the calibration tasks have been split to handle CCD-specific calibrations (**ACSCCD**) separately from those steps which can be applied to any ACS data (**ACS2D**). For example, the MAMA data obtained with the SBC do not have the overscan regions found in CCD data.

The initial processing performed on CCD data alone is shown in Figure 3.2. The reference files appropriate for each processing step and the calibration switches controlling them are also given beside the name of the task they control. The output (overscan-trimmed image) from **ACSCCD** is then sent through **ACS2D** as shown in Figure 3.3. Raw MAMA data are processed from the start with **ACS2D**, which initializes the error and DQ arrays (already performed for CCD data) and which performs linearity corrections.

Figure 3.1: Flow Diagram for ACS data with **CALACS** task names in bold.

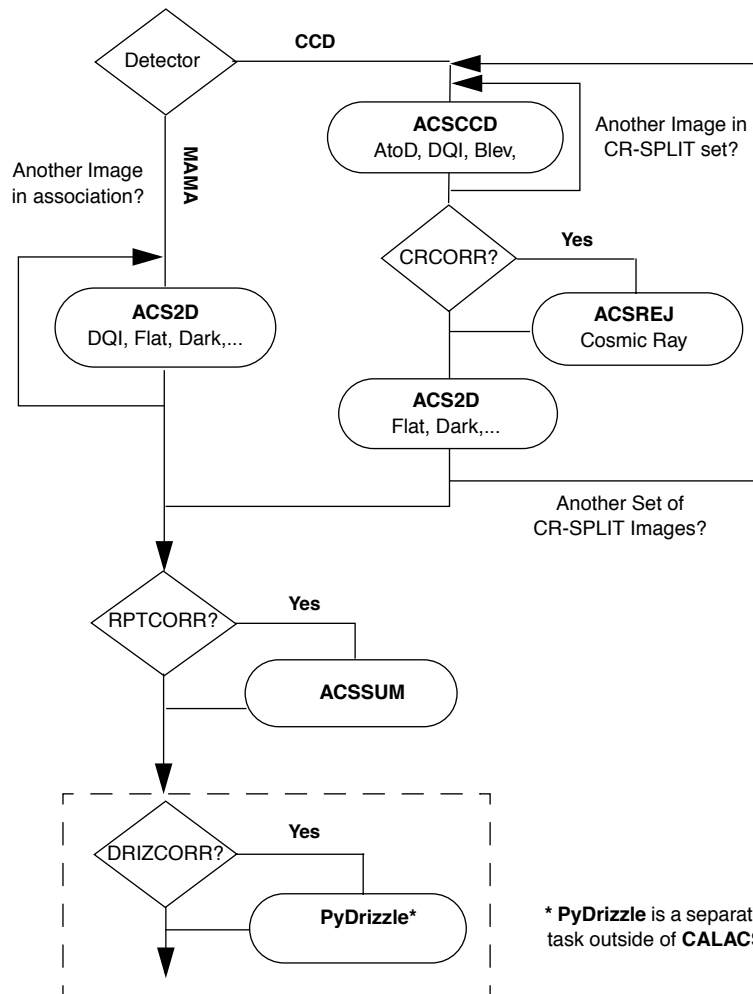


Figure 3.2: Flow diagram for CCD data using ACSCCD in CALACS.

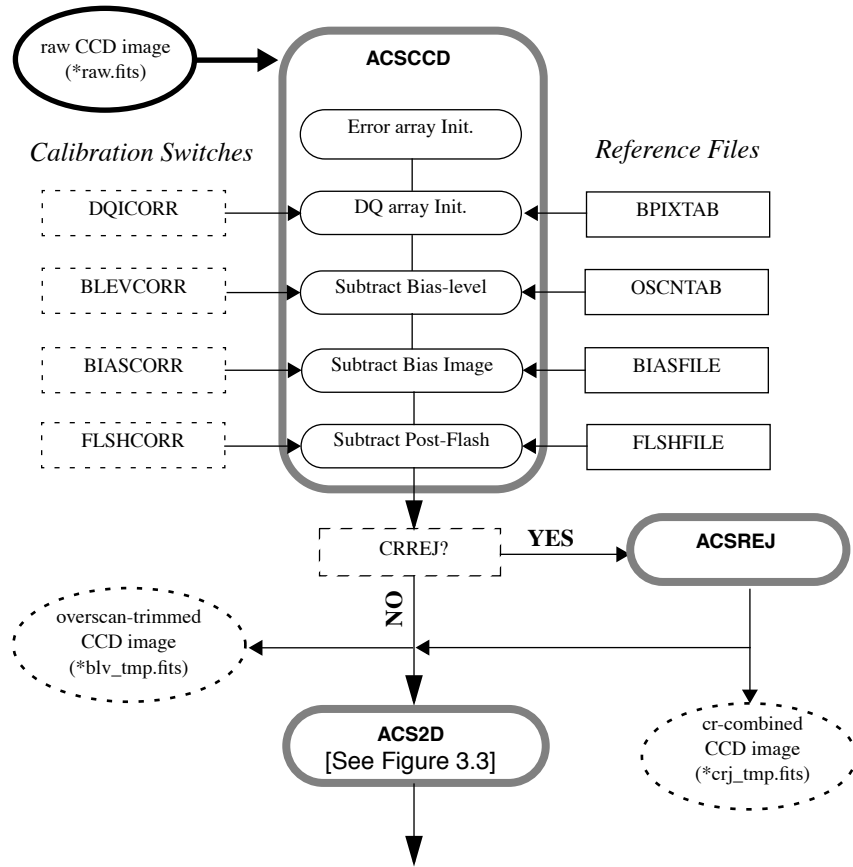
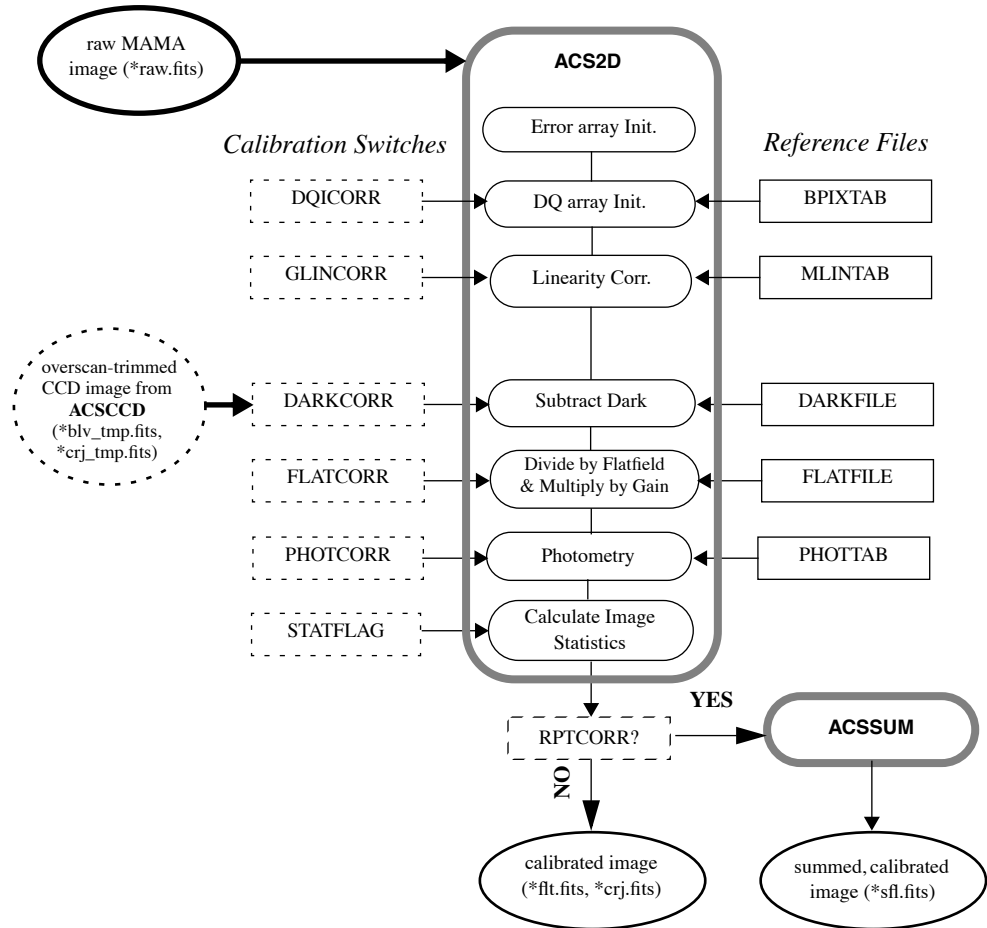


Figure 3.3: Flow diagram for MAMA and overscan-trimmed CCD data using ACS2D in CALACS.



3.4 CALACS Processing Steps

The CALACS pipeline consists of four individual calibration tasks: ACS2D, ACSREJ, ACS2D, and ACSSUM. These tasks are listed in Table 3.2 and diagrammed in Figure 3.1. CALACS is responsible for controlling the processing rather than actually calibrating the data. The individual tasks apply the desired calibration steps to the data and create the output products, including the trailer files.

In the following four sections, we describe each CALACS task, give a detailed description of the calibrations steps performed within each task, and give a brief description of the reference files used for each step.

3.4.1 ACSCCD

This routine contains the initial processing tasks for all ACS CCD data. These tasks are listed in operational order in Table 3.3. The bulk of this routine was based on the code for **CALSTIS1**. **ACSCCD** processes each image in the input list one at a time, using the header keywords to determine which calibrations are performed. Upon completion of **ACSCCD**, the overscan regions will be trimmed from the image and an output image with the file extension ‘_blv_tmp.fits’ is created.

Table 3.3: ACSCCD processing tasks

doNoise	Initialize error array
doDQI	Initialize data quality array
doAtoD	Perform A to D correction (currently skipped)
doBlev	Subtract bias level from overscan region
doBias	Subtract bias image
doFlash	Subtract post-flash image (if required)
Final Output	Output overscan-trimmed or full image

doNoise - Error Array Initialization

- Header Switch: None
- Header Keywords Updated: None
- Reference File: CCDTAB (*_ccd.fits)

The first processing step is initializing the image error array. The **doNoise** function examines the ERR extension of the input data to determine the state of the array. Input ‘_raw.fits’ images delivered by Generic Conversion will contain a NULL ERR array, defined by the keywords NPIX1, NPIX2 and PIXVALUE, where PIXVALUE=0. If the ERR array has already been expanded and contains values other than zero, then this function does nothing. Otherwise, **CALACS** will initialize the array and **doNoise** will assign a simple noise model.

The noise model reads the science array and for each pixel calculates the error value σ (in DN for the CCD, in counts for the MAMA):

$$\sigma_{\text{CCD}} = \sqrt{(DN - \text{bias}) / (\text{gain}) + (\text{readnoise} / \text{gain})^2}$$

$$\sigma_{\text{MAMA}} = \max(1, \sqrt{\text{counts}})$$

Because MAMA data is not processed with **ACSCCD**, the **doNoise** function is later called as part of **ACS2D**.

The CCDTAB reference file, the CCD Characteristics Table, is used to initialize the ERR array by determining the bias, gain, and readnoise for each observation. The table contains one row for each amp configuration which can be used during readout. Each configuration is uniquely identified by the list of amps possible (CCDAMP), the particular chip being read out (CCDCHIP), the commanded gain (CCDGAIN), the commanded bias level

(CCDBIAS), and the pixel bin size (BINAXIS). CCDTAB uses these commanded values to determine each amp's physical read-out characteristics, including readnoise (READNSE), A-to-D gain (ATODGN), and bias level (CCDOFST). Note that by using the bias level from the CCDTAB, instead of from a full bias image, to create the ERR array, slight errors may be introduced which could be relevant for faint sources.

doDQI - Bad Pixel Determination

- Header Switch: DQICORR
- Header Keywords Updated: None
- Reference File: BPIXTAB (*_bpx.fits), CCDTAB (*_ccd.fits)

The function **doDQI** initializes the data quality (DQ) array by combining it with a table of known bad pixels for the detector, stored in the Bad-Pixel reference table (BPIXTAB). The type of bad pixel flags are listed in Table 3.4.

The DQ array may have already been partially populated to reflect pixels which were affected by telemetry problems or compression buffer overflow. Other DQ flags will be added in further processing steps (such as cosmic-ray rejection). For CCD data, the values in the SCI extension are also checked for saturation by comparison with the value of the SATURATE column in the CCD parameters table CCDTAB.

The **doDQI** function combines the DQ flags from pre-processing, from the BPIXTAB, and from saturation tests into a single result for the particular observation. These values are combined using a “bitwise/logical OR operator” for each pixel. Thus, if a single pixel is affected by two DQ flags, those flag values will be added in the final DQ array. This array then serves as a mask so that **CALACS** can ignore bad pixels during processing.

The BPIXTAB reference file maintains a record of the (x,y) position and DQ value for all known bad pixels in each CCD. These pixels may change as some hot pixels are annealed and others appear over time, for example. Permanently bad pixels due to chip defects may be flagged during Generic Conversion, and it is the job of the BPIXTAB to maintain the list of bad pixels applicable for a given time period.

Table 3.4: Flags for the DQ array

Flag Value	Definition
0	good pixel
1	Reed-Solomon decoding error
2	data replaced by fill value
4	bad detector pixel or beyond aperture
8	masked by aperture feature
16	new hot pixel
32	“blob” at end of hot column
64	pre-existing hot pixel
128	bias level pixel
256	saturation (full well or a-to-d)
512	bad pixel in reference file
1024	weak trap
2048	a-to-d saturation*
4096	<i>reserved</i>
8192	cosmic ray rejected during image combination

*Note: The ability to flag pixels affected by a-to-d situation as 2048 will be implemented in the pipeline in Spring 2004.

doAtoD - A-to-D Correction (skipped)

- Header Switch: ATODCORR
- Header Keywords Updated: None
- Reference File: ATODTAB (*_a2d.fits)

The **doAtoD** function converts pixel values in the raw input image from units of DN (16-bit integer) to counts (32-bit float). An analog-to-digital correction would be applied if the CCD electronic circuitry, which performs the analog-to-digital conversion, were biased toward the assignment of certain DN (data number) values. Ground test results show that this correction is not currently needed, so the ATODCORR switch will always be set to OMIT.

doBlev - Bias-level Correction

- Header Switch: BLEVCORR
- Header Keywords Updated: BIASLEV[A,B,C,D], MEANBLEV, CRPIX[1,2], LTV[1,2]
- Reference File: OSCNTAB (*_osc.fits)

The **doBlev** function fits the bias level from the overscan region and subtracts it from the image data. The definition of the overscan region is

taken from the OSCNTAB reference file. With these regions defined, the overscan level for each row of the input image is measured and a straight line is fit as a function of image line number. The fit will then be evaluated for each row of the input image, and that value will be subtracted from each pixel of that row. If the measured overscan level for a given row is greater than 3 sigma times the mean of all rows, that row will be excluded from the linear fit. If the overscan level cannot be determined, or if the overscan region is not present in the input image, a default value (the CCDBIAS from the CCD parameters table) will be subtracted instead. In this case, an error message will be written to the image trailer file.

In **CALSTIS**, the overscan regions were removed from the image just after the bias-level subtraction, since no subsequent processing was required. **CALACS**, on the other hand, retains the output image in memory until the completion of all the processing steps in **ACSCCD**. The overscan regions will not be trimmed until the image is written to disk at the completion of **ACSCCD**.

The positional keywords **CRPIX*** and **LTV*** will be updated to reflect the offset due to removing the overscan. In addition, the mean value of all overscan levels will be computed and written to the SCI extension header as **MEANBLEV**. The individual bias level measured for each amplifier will be written to the SCI header as **BIASLEV***.

The OSCNTAB reference file (Overscan Region Table) describes the overscan regions to be used for measuring the bias level of the observation. Each row corresponds to a specific configuration, given by the CCD amp and chip used. The columns **TRIMX*** give the number of columns to trim off the beginning and end of each line (the physical overscan region), while the **TRIMY*** columns give the number of rows to trim off the top and bottom of each column (the virtual overscan region). The result of trimming (**TRIMX1 + TRIMX2**) columns and (**TRIMY1 + TRIMY2**) rows gives the final calibrated image sizes, 4096x2048 for a full WFC image and 1024x1024 for a full HRC image. The OSCNTAB columns **BIASSECTA*** and **BIASSECTB*** give the range of image columns to be used for determining the bias level in the leading and trailing regions, respectively, of the physical overscan region. The virtual overscan region is also defined in the OSCNTAB, where (**VX1, VY1**) indicates the beginning of the upper or lower overscan region, extending to the position (**VX2, VY2**).

To determine which overscan regions were actually used for determining the bias level, users are encouraged to check the OSCNTAB reference file. If desired for manual calibration, users may modify the bias section and virtual overscan region definitions in the reference table, but the **TRIMX***, **TRIMY*** columns must not be changed.

doBias - Bias Image Subtraction

- Header Switch: **BIASCORR**
- Header Keywords Updated: **None**
- Reference File: **BIASFILE (*_bia.fits)**

Subtraction of the bias image is performed prior to cosmic-ray rejection using the function **doBias**. The appropriate reference image, **BIASFILE**, is selected using the **DETECTOR**, **CCDAMP**, and **CCDGAIN** keywords. The dimensions of the science image are used to distinguish observations which use the entire chip from sub-array images. Because the bias image has already been scaled by the gain and had the overscan values subtracted, the bias image may have a mean pixel value which is less than one.

Due to the way the bias reference image is created, part of the dark subtraction is also included in this step. Dark counts accumulate for an additional time beyond the exposure time, primarily the time required to read out the detector, and this portion of the dark current is subtracted along with the bias. This is described further in the section on Dark Image Subtraction.

The bias reference files are updated frequently (~weekly). The **BIASFILE** has the same dimensions as a full-size science image, 1062x1044 for HRC and 4144x2068 for WFC, allowing for simple image arithmetic on arrays of equal size. Only after the completion of **ACSCCD** are the images trimmed to 1024x1024 (HRC) and 4096x2048 (WFC). **CALACS** assumes that CCD data will not be read-out in binned mode, so any image which is not the full size is assumed to be a sub-array.

For sub-array images, **CALACS** uses the **LTV[1,2]** keywords to extract the appropriate region from the reference file and apply it to the sub-array input image. For users who take their own bias files, the **BIASFILE** keyword will need to be updated manually before recalibrating. This is described in detail in Section 3.5.2.

doFlash - Post-Flash Subtraction

- Header Switch: **FLSHCORR**
- Header Keywords Updated: **MEANFLSH**
- Reference File: **FLSHFILE** (*_fls.fits)

ACS has a post-flash capability to provide the means of mitigating the effects of Charge Transfer Efficiency degradation. The proposer controls whether to use this capability via the **FLASHEXP** optional parameter in the Phase II proposal.

The function **doFlash** will subtract the Post-Flash reference file, **FLSHFILE**, from the science image. This file has the same dimensions as a full-size science image, 1062x1044 for HRC and 4144x2068 for WFC, allowing for simple image arithmetic on arrays of equal size. The appropriate **FLSHFILE** is selected using the following keywords from the image header: **DETECTOR**, **CCDAMP**, **CCDGAIN**, **FLASHCUR** and **SHUTRPOS**.

First, the success of the Post-Flash exposure is verified by checking the keyword **FLASHSTA**. If any problems were encountered, a comment will be added to the history comments in the **SCI** extension header. The **FLSHFILE** is then normalized to one second for the appropriate post-flash current level (**LOW**, **MED**, **HIGH**), given by the **FLASHCUR** keyword.

The `FLSHFILE` is then multiplied by the flash duration in seconds (`FLASHDUR`) and is subtracted from the science image. Finally, the mean value of the scaled Post-Flash image is written to the output SCI extension header as the keyword `MEANFLSH`.

Final Output

Upon completion of `ACSCCD`, the overscan regions will be trimmed from the image when it is written out, but only if `doBlev` is performed successfully. Otherwise, the full image array will be written out.

3.4.2 ACSREJ

- Header Switch: `CRCORR`
- Header Keywords Updated: `TEXPTIME`, `SKYSUM`, `EXPEND`, `REJ_RATE`, `EXPTIME`, `NCOMBINE`, `ROOTNAME`
- Header Keywords added as ‘`HISTORY`’ Comments: `INITGUES`, `SKYSUB`, `CRSIGMAS`, `MEANEXP`, `CRRADIUS`, `CRTHRESH`, `SCALENSE`, `CRMASK`, `NEXTEND`
- Reference File: `CRREJTAB (*_crr.fits)`

`ACSREJ`, the cosmic-ray rejection task in `CALACS`, combines `CR-SPLIT` exposures into a single image, first detecting and then replacing flagged pixels. The task uses the same statistical detection algorithm developed for `STIS` data (`OCRREJ`) and `WFPC2` data (`CRREJ`), providing a well-tested and robust procedure.

First, `ACSREJ` will compute the sky background using the mode of each image. Sky subtraction is performed before any statistical checks are made for cosmic rays. Next, `ACSREJ` constructs an initial comparison image from each input exposure. The comparison image can either be a median- or minimum-value sky-subtracted image constructed from all the input images and represents the ‘first guess’ of a cosmic-ray free image. The comparison image serves as the basis for determining the statistical deviation of each pixel from the input image.

A detection threshold is then calculated for each pixel based on the comparison image. This threshold is equal to a constant times sigma squared, given in the equation below:

$$\tau_n = \sigma^2 \times (\text{noise} + \text{value}(x, y)/\text{gain} + \text{scale}(\text{value}(x, y) - \text{sky}_n)^2) / T_n^2$$

where:

- σ is the sigma value used as the detection limit,
- *noise* is the readnoise squared (in DN) and *gain* is the CCD gain (in e⁻/DN) for the amp used to read the pixel,
- *scale* is the scale factor for the noise model,
- T_n is the exposure time (in seconds) for the input image, and
- *value* is the pixel value (in DN) from the median or minimum combined comparison image.

The actual cosmic-ray detection criteria at each pixel is determined as:

$$\Delta = ((pix_n(x, y) - sky_n) / T_n - median(x, y))^2$$

where:

- $pix_n(x, y)$ is the pixel value (in DN) from input image n ,
- sky_n is the sky background (in DN) of image n , and
- $median_n(x, y)$ is the median or minimum pixel value (in DN/sec) from the comparison image.

If $\Delta > \tau_n$, the pixel is flagged as a cosmic-ray in the input image's DQ array and is ignored when images are summed together. Surrounding pixels within a given expansion radius are marked as 'SPILL' pixels and are given less stringent detection thresholds.

When all input images have been processed, the values of the acceptable pixels are summed over all input images. Each pixel in the summed output array is then scaled by the total exposure time:

$$pixout(x, y) = T \cdot \frac{\sum_n (pix_n(x, y) - sky_n) m_n(x, y)}{\sum_n T_n m_n(x, y)} + \sum_n sky_n$$

where:

- T_n is the exposure time for image n ,
- $m_n(x, y)$ is the mask value (0 for cr-rejected pixels, 1 for good data) for the n 'th image at pixel (x, y) ,
- T is the total exposure time (regardless of whether all input images were used for that particular pixel). This corresponds to the header keywords `TEXPTIME`, `EXPTIME`.

The following keywords are derived from the variables in this computation:

- `TEXPTIME` = `EXPTIME` = T
- `SKYSUM` = $\sum_n sky_n$
- `REJ_RATE` = $(\sum_n T_n m_n(x, y) / T)$ averaged over all pixels
- `NCOMBINE` = n

The remaining keywords `EXPSTART`, `EXPEND` are updated based on the input image headers.

In summary, the cosmic ray rejection task sums all accepted pixel values, computes the true exposure time for that pixel, and scales the sum to correspond to the total exposure time. The final scaled, cleaned pixel is written to the comparison image to be used for the next iteration. This process is then repeated with successively less stringent detection thresholds, as specified by `CRSIGMAS`. Further processing by `CALACS` will scale the $pixout(x, y)$ array by the gain, resulting in the summed, cosmic-ray eliminated, but not sky-subtracted product (`*_crj.fits`) in units of *electrons*.

Cosmic-ray Rejection Table

ACSREJ uses the Cosmic-ray Rejection Parameter Table (CRREJTAB) to determine the number of iterations for cosmic-ray rejection, the sigma levels to use for each iteration, and the spill radius to use during detection. This allows the rejection process to be tuned to each detector, with suitable defaults being applied during pipeline processing. Observers may fine-tune the cosmic-ray rejection parameters when manually reprocessing data with **ACSREJ** by editing the CRREJTAB.

The CRREJTAB reference file contains the basic parameters necessary for performing cosmic-ray rejection. The column names and default values for the CRREJTAB are given in Table 3.5. The appropriate row is selected based on the chip being processed (CCDCHIP), the number of images into which the exposure was split (CR-SPLIT), and the exposure time of each CR-SPLIT image (MEANEXP). The sky fitting algorithm is controlled by the parameter SKYSUB which can have values of ‘mode’ or ‘none’. The ‘first guess’ CR-combined image is then created using the median or minimum value of the input exposures, as specified by the column INITGUES.

Table 3.5: Columns in Cosmic-Ray Rejection Parameters Table

Column Name	Default Value	Contents
CRSPLIT		Number of exposures into which observation was split
MEANEXP	INDEF	Average exposure time (sec) for each image
SCALENSE	30.0	Multiplicative term (in percent) for the noise model
INITGUES	minimum	Method for computing initial-guess image (minimum, median)
SKYSUB	mode	Sky fitting algorithm (mode, none)
CRSIGMAS	6.5, 5.5, 4.5	Rejection thresholds (sigma) for consecutive iterations
CRRADIUS	2.1	Radius (in pixels) for propagating cosmic ray
CRTHRESH	0.5555	Propagation factor
BADINPDQ	39	Data quality file bits to reject
CRMASK	yes	Flag CR-rejected pixels in input files?
CCDCHIP		Chip to which this conversion applies

Cosmic-ray detection requires the specification of a threshold above which a pixel value is considered a cosmic-ray. This threshold is defined above as $\tau_n = \sigma^2 \times constant$ and uses the sigma rejection thresholds σ . These sigmas correspond to the column CRSIGMAS in the CRREJTAB reference file. SCALENSE is a multiplicative term (in percent) for the noise model and is given as *scale* in the threshold equation. This term can be useful when the pointing of the telescope has changed by a small fraction of a pixel between images. Under such circumstances, the undersampling of the image by the CCD will cause stars to be rejected as cosmic rays if a scale noise term is not included. This is a crude but effective step taken to satisfy the maxim of “first do no harm”. However,

for cases in which there have been no frame-to-frame offsets or the image is locally well-sampled, this will unduly bias *against* rejecting cosmic rays.

Pixels within a given radius, CRRADIUS, of a cosmic-ray will also be treated as cosmic-rays. A less stringent rejection threshold, CRTHRESH, can be used for detecting pixels adjacent to a cosmic-ray. As for CRSIGMAS, CRTHRESH is also given as a sigma value. If CRTHRESH is exceeded, pixels within a defined radius of the cosmic ray will also be flagged. All pixels determined to be affected by a cosmic-ray will have their DQ values set to 8192, as described in Table 3.4.

A much more detailed discussion of cosmic ray elimination in repeated ACS exposures may be found in [ACS ISR 2002-08](#), Section 5.1. Of particular note is the appropriate value of SCALENSE to use; the pipeline adopts a conservative value to avoid doing harm. In recalibrating several frames for a new cosmic ray elimination, it would be advisable to determine the full range of relative x,y offsets. An appropriate value of SCALENSE is $100 * (\text{maximum offset in pixels})$, thus if the full offset range was 0.1 pixels an appropriate SCALENSE is 10.0. To alter this value, one may use **tedit** on the CRREJTAB file (calibration file with the extension 'crr.fits'). The number of exposures obtained in a repeated observation set can be larger than the maximum CR-SPLIT=8 allowed in Phase II proposals. For example, for program 9662 used in Examples 3 and 4 in Section 3.5.2, there were 14 individual 1.0s HRC exposures all at nominally the same pointing. A check of relative offsets shows that a shift of about 0.2 pixels occurred between the first and last exposures. To obtain a crj combined image of all 14 inputs at 1.0s and with better sensitivity to cosmic ray elimination than provided by the conservative default of SCALENSE = 30.0, the following should be done:

1. Create a new asn.fits table with one EXP-CRJ entry for each of the 14 raw 1.0s images, and an appropriately named PROD-CRJ file line.
2. Use **tedit** on the crr.fits table adding a line with a CR-SPLIT value of 14, and a SCALENSE value of 20.0.
3. Given such a large number of inputs it would also make sense here to change to 'median' as the INITGUES value. Then rerun **CALACS** on the new association table with 14 entries to obtain a CRJ extension image based on the full stack of 1.0s images.

For a detailed discussion on manually recalibrating ACS data, refer to Section 3.5.

3.4.3 ACS2D

Every observation, whether taken with the MAMA or CCD detectors, will be processed by **ACS2D**. The **ACS2D** primary functions are listed in Table 3.6 and include the dark current subtraction, flat fielding, and photometric keyword calculations. **ACS2D** contains the same data quality and error array initialization functions used in **ACSCCD**, but **ACS2D** will check to ensure that the array initialization is not performed twice on CCD

data. Calibration switches in the image header control the performance of the remaining calibration functions, with MAMA-specific functions being initiated only when the relevant calibration switches are set.

Table 3.6: The functions performed in **ACS2D** (in operational order)

doNoise	Apply a simple noise model (if not done in ACSCCD)
doDQI	Initialize data quality array (if not done in ACSCCD)
doNonLin	Correct and flag non-linear data (MAMA only)
doDark	Subtract dark image
doFlat	Divide by flat field and multiply by gain
doShad	Perform CCD shutter shading correction (currently skipped)
doPhot	Compute photometric keyword values for header
doStat	Compute image statistics

doNoise - Error Array Initialization

- Header Switch: None
- Header Keywords Updated: None
- Reference File: None

ACS2D will first check that the image error array has not been expanded, indicating that no previous processing has been performed. In this case, the value of all pixels in the ERR array will have a value of zero. **ACS2D** will then perform the same initialization as described in “doNoise - Error Array Initialization” for **ACSCCD**. If, however, the input image has already been processed, no changes to the ERR array will be made.

doDQI - Bad Pixel Determination

- Header Switch: DQICORR
- Header Keywords Updated: None
- Reference File: BPIXTAB (*_bpx.fits)

If the DQICORR header keyword switch is set to COMPLETE (e.g. CCD data), this step will be skipped. Otherwise, the same initialization will be performed as described in “doDQI - Bad Pixel Determination” for **ACSCCD**.

doNonLin - Linearity Correction for MAMA Data

- Header Switch: LFLGCORR, GLINCORR
- Header Keywords Updated: GLOBLIM
- Reference File: MLINTAB (*_lin.fits)

This routine is capable of flagging global and local nonlinearity in ACS MAMA observations, where the term global refers to the entire ACS detector and local refers to an individual detector pixel. The MAMA Linearity Table, MLINTAB, provides the basic parameters for determining linearity. The global limit (GLOBAL_LIMIT) column from this table refers to the total count rate at which the data are affected by greater than 10% non-linearity across the detector.

CALACS will attempt to correct for non-linearity up to the global limit using the non-linearity time constant in the column TAU. The global linearity correction is computed for every pixel below the global linearity limit specified by iteratively solving the equation $n = Ne^{(-\tau N)}$ to get the true count rate N.

The LOCAL_LIMIT can actually be much higher than the global limit and is difficult to correct using a simple algorithm. Each pixel found to exceed this limit will simply be marked as non-linear in the DQ file. This DQ flag will be extended by a fixed radius from the original pixel, given in the EXPAND column and is currently set to 2 pixels.

If the LFLGCORR switch is set to PERFORM, **ACS2D** will *flag* excessive global and local nonlinearity in the DQ array. If GLINCORR is set to PERFORM, it will *correct* excessive global nonlinearity in the SCI array, if it is not too large. If the global linearity limit is exceeded, the keyword GLOBLIM in the SCI extension header will be set to “EXCEEDED”. Otherwise, it will have the value “NOT-EXCEEDED”.

doDark - Dark Image Subtraction

- Header Switch: DARKCORR
- Header Keywords Updated: MEANDARK
- Reference File: DARKFILE (*_drk.fits)

This task is responsible for subtracting the dark image from the input image. For CCD data, the dark image (in *electron s⁻¹*) is multiplied by the exposure time and then divided by the gain before subtracting. (MAMA data do not require a gain correction.) The dark reference file, DARKFILE, is read in line-by-line and subtracted from the input image in memory. The mean dark value is then computed from the scaled dark image and used to update the MEANDARK keyword in the SCI image header. The “dark time” is simply the exposure time; it does NOT include the idle time since the last flushing of the chip or the readout time. Any dark accumulation during readout time or post-flash idle will be included automatically into the BIASFILE.

A unique dark reference file is created (approximately) daily using four 1000 sec frames for both the WFC and HRC. The four darks are combined (with cosmic ray rejection) and bias-corrected to form a “daydark”. Every two weeks, a high signal-to-noise “basedark” is created from all the frames (~56) in that period. It does not contain the newest hot pixels present in the daydarks since they are rejected with the cosmic rays during image combination. A copy of the basedark is made for each day in the two-week period, and daily hot pixels are skimmed from the respective daydarks and added to the basedark copies. Only the hot pixels above 0.08 e⁻/sec are identified with flag 16 in the data quality (DQ) array of WFC and HRC reference darks, which propagates to the DQ array of the calibrated science data. The flagged hot pixels are typically excluded in subsequent processing steps, as they are likely too noisy to be correctable. But the

many "warm" pixels below this threshold are assumed to be adequately corrected by the dark calibration. This produces a reference file with high signal-to-noise, which accurately reflects (and corrects/flags) the hot pixels present for a given observation date. The “best” dark file is typically not available in the pipeline until 2-3 weeks after the date of observation, because it takes a few weeks to collect enough frames to make a basedark (see [ACS ISR 04-07](#) for more information).

The reference file for dark subtraction, DARKFILE, is selected based on the values of the keywords DETECTOR, CCDAMP, and CCDGAIN in the image header. The dark correction is applied after the overscan regions are trimmed from the input science image. As for the BIASFILE, CALACS assumes that the images have not been binned, so any input image smaller than the full detector size will be interpreted as a sub-array image.

Sub-array science images will use the same reference file as a full-sized image. CALACS simply extracts the appropriate region from the reference file and applies it to the sub-array input image.

doFlat - Flat field image(s) Correction

- Header Switch: FLATCORR
- Header Keywords Updated: None
- Reference File: PFLTFILE (*_pfl.fits), LFLTFILE (*_lfl.fits), DFLTFILE (*_dfl.fits)

As done for STIS, the ACS **doFlat** routine corrects for pixel-to-pixel and large-scale sensitivity gradients across the detector by dividing the data by the flat field image. This contrasts with WFPC2 and NICMOS who deliver an “inverse” flat to the pipeline and then multiply the data by the flat field. When performing the flat field correction, CALACS multiplies by the gain so that the calibrated data will be in units of electrons.

Because of geometric distortion effects, the area of the sky seen by a given pixel is not constant, and therefore observations of a constant surface brightness object will have count rates per pixel that vary over the detector, even if every pixel has the same sensitivity. In order to produce images that appear uniform for uniform illumination, the flat fields make an implicit correction for the geometric distortion across the field that is equivalent to dividing each pixel by the optical distortion which is normalized to unity at the center of the field. A consequence of this procedure is that two stars of equal brightness do not have the same total counts after the flat fielding step. Thus, point source photometry extracted from a flat fielded image must be multiplied by the effective pixel area map, as shown in Figure 6.1. This correction is accounted for in pipeline processing by **PyDrizzle** which uses the geometric distortion solution to correct all pixels to equal areas. Thus, in the drizzled images photometry is correct for both point and extended sources.

The flat field image used to correct the data is created using up to three flat field reference files: the pixel-to-pixel file (PFLTFILE), the low-order

flat (LFLTFILE), and the delta flat (DFLTFILE). The PFLTFILE is a pixel-to-pixel flat field correction file containing the small scale flat field variations. The LFLTFILE is a low-order flat which will correct for any large-scale flat field variations across each detector. This file is stored as a binned image which is expanded when being applied by **CALACS**. The DFLTFILE is a delta-flat containing any needed changes to the small-scale PFLTFILE.

If the LFLTFILE and DFLTFILE are not specified in the SCI header, only the PFLTFILE will be used for the flat field correction. If all three reference files are specified, they will be read in line-by-line and multiplied together to form a complete flat field correction image. Currently, the LFLTFILE and DFLTFILE flats are not used for ACS data. The PFLTFILE reference flat in the pipeline is actually a combination of the pixel-to-pixel flats taken during the ground calibration and the low-order flat correction derived in-flight.

All flat field reference images will be chosen based on the detector, amplifier, and filters used for the observation. Any sub-array science image will use the same reference file as a full-size image. **CALACS** will extract the appropriate region from the reference file and apply it to the sub-array input image.

doShad - Shutter Shading File Correction (skipped)

- Header Switch: SHADCORR
- Header Keywords Updated: None
- Reference File: SHADFILE (*_shd.fits)

The SHADCORR calibration switch is currently set to OMIT and is unlikely to be used in future versions of **CALACS**. Calibration data show that a shading correction is not needed for ACS data. For more information, refer to [ACS ISR 03-03](#), “Stability and Accuracy of HRC and WFC Shutters” and to the report “WFC and HRC Shutter Shading and Accuracy” on the ACS Science and Engineering Team web site at:

http://acs.pha.jhu.edu/instrument/calibration/results/by_instrument/shutter/shading_accuracy_feb2001

The **doShad** routine applies the shutter shading correction image (SHADFILE) to the science data if SHADCORR is set to PERFORM. The input image is corrected for the differential exposure time across the detector caused by the shutter opening. This correction only has a significant effect on images with short exposure times (less than 10 seconds).

Pixels are corrected based on the exposure time using the relation:

$$\text{corrected} = \text{uncorrected} \times \text{EXPTIME} / (\text{EXPTIME} + \text{SHADFILE}) .$$

The SHADFILE is selected using the DETECTOR keyword in the input science image. At present, this reference file is an over scanned-trimmed

image, binned by a factor of 8 for each dimension, resulting in a 128x128 HRC and a 256x512 WFC SHADFILE.

The shutter shading correction can either be applied during **ACS2D** processing for single or REPEAT-OBS exposures or during cosmic-ray rejection in **ACSREJ** for CRSPLIT exposures.

doPhot - Photometry Keyword Calculation

- Header Switch: PHOTCORR
- Header Keywords Updated: PHOTMODE, OBSMODE, PHOTFLAM, PHOTZPT, PHOTPLAM, PHOTBW
- Reference Files: GRAPHTAB (*_tmg.fits), COMPTAB (*_tmc.fits)

Before photometry can be performed on ACS observations, a transformation from electrons to absolute flux units must be performed. **CALACS** follows the WFPC2 methodology for calculating the photometry keywords in the calibration pipeline. The calibration reference files, GRAPHTAB and COMPTAB, point to the SYNPHOT tables containing the latest ACS component tables. These tables contain the throughput as a function of wavelength for the various ACS detector and filter combinations. This reliance on SYNPHOT allows the ACS team to maintain the latest throughput files in SYNPHOT to keep **CALACS** up to date. For further discussion of SYNPHOT, refer to Chapter 3 of the HST Data Handbook Introduction.

The keywords PHOTMODE and OBSMODE are built to reflect the configuration of the instrument during the exposure. **CALACS** then uses the OBSMODE string with SYNPHOT to compute the total throughput array for the observation, based on the aperture and filter throughputs, and the photometry keywords: PHOTFLAM (the inverse sensitivity in units of $\text{erg s}^{-1} \text{cm}^{-2} \text{A}^{-1}$ for 1 *electron s*⁻¹), PHOTZPT (the Space Telescope magnitude zero point), PHOTPLAM (the bandpass pivot wavelength), and PHOTBW (the bandpass RMS width). Users who wish to convert calibrated images (in units of *electrons*) to flux units may simply divide the image by the exposure time and then multiply by the PHOTFLAM keyword. Drizzled DRZ images are already in units of *electrons per second* and may simply be multiplied by the PHOTFLAM value to attain flux units.

doStat - Image Statistics Determination

- Header Switch: STATFLAG
- Header Keywords Updated: NGOODPIX, GOODMIN, GOODMAX, GOODMEAN, SNRMIN, SNRMAX, SNRMEAN
- Reference File: None

This routine computes the number of pixels which are flagged as “good” in the data quality array. The minimum, mean, and maximum pixel values are then calculated for data flagged as “good” in both the science and error arrays. Similarly, the minimum, mean, and maximum signal-to-noise of “good” pixels is derived for the science array. These quantities are updated in the image header.

3.4.4 ACSSUM

- Header Switch: RPTCORR
- Header Keywords Updated: NCOMBINE, EXPTIME, EXPEND, ROOTNAME
- Reference File: None

ACSSUM sums a list of images one line at a time, using the association file to define the input. A straight pixel-to-pixel addition of the science values is applied, and the error calculated as the square root of the sum of the squares of the errors in the individual exposures. Note that for multiple exposures taken via REPEAT-OBS instead of the observationally equivalent CR-SPLIT, the final processed image is *not* cosmic-ray rejected.

The calibration switch RPTCORR is set to COMPLETE upon successful completion of the summation. In addition, the keywords NCOMBINE, EXPTIME, and EXPEND are adjusted to reflect the total of the summed images.

3.5 Manual Recalibration of ACS Data

3.5.1 Requirements for Manual Recalibration

Retrieving Software and Input Data Files

If the observer decides to recalibrate their ACS data, the following must be available on their system:

- **CALACS** software from the STSDAS **hst_calib.acs** package
- **PyDrizzle** task from the STSDAS **dither** package
- PyRAF, to run **PyDrizzle**, obtained from STScI
- Reference files obtained from STScI Archive, or personal versions
- Uncalibrated data (*_raw.fits) from OTFR
- Association table, if applicable.

Uncalibrated data and calibration reference files can be obtained from OTFR via StarView or the web-based archive request form:

<http://archive.stsci.edu/cgi-bin/hst>.

StarView is available for users to download from:

<http://starview.stsci.edu/html>.

The pages include instructions on installing and using StarView and on how to become a registered archive user.

The most recent version of STSDAS, which includes the **CALACS** package required for reprocessing, can be downloaded from:

http://www.stsci.edu/resources/software_hardware/stsdas

The most recent versions of PyRAF, **PyDrizzle**, and **MultiDrizzle** can be downloaded from:

http://www.stsci.edu/resources/software_hardware/stsdas/pydrizzle

Setting up ‘jref’

Before any recalibration may be done, the user’s local directory containing the calibration reference files must be defined. For ACS, this directory is referred to as ‘jref’. The image header already contains the appropriate keywords defining the reference files. The user must simply define the location of the ‘jref’ directory in the Unix environment.

```
setenv jref /mydisk/myjref/
```

Once this path is set, IRAF must be restarted in the same window in which the setup was performed.

Selecting Calibration Switches

OTFR will use the latest available calibration reference files by default. In order to use non-default reference files, manual recalibration is required. The calibration reference file keywords will need to be updated manually in the uncalibrated data with the non-default filenames before running **CALACS**. The selection criteria in Table 3.7 are used to set the values for the calibration switch header keywords in uncalibrated ACS data.

Table 3.7: Calibration Switch Selection Criteria.

Switch	Description	Criteria
DQICORR	Data Quality Array Initialization	DEFAULT = “ PERFORM ” If OBSMODE = ACQ then “ OMIT ”
ATODCORR	Analog to Digital Conversion	DEFAULT = “ OMIT ”
BLEVCORR	Overscan Region Subtraction	DEFAULT = “ PERFORM ”
BIASCORR	Bias Subtraction	DEFAULT = “ PERFORM ”
FLSHCORR	Post Flash Subtraction	DEFAULT = “ OMIT ”
CRCORR	Cosmic Ray Rejection	If CRSPLIT \geq 2 then “ PERFORM ” If CRSPLIT < 2 then “ OMIT ”
EXPSCORR	Calibrate individual exposures in an association	DEFAULT = “ PERFORM ”
SHADCORR	Shutter Shading Correction	DEFAULT = “ OMIT ”
DARKCORR	Dark Subtraction	DEFAULT = “ PERFORM ”
FLATCORR	Flat field Division	DEFAULT = “ PERFORM ”
PHOTCORR	Photometric Processing	DEFAULT = “ PERFORM ”
RPTCORR	Repeat Processing	DEFAULT = “ OMIT ” If NRPTEXP > 1 then “ PERFORM ”
DRIZCORR	Dither Processing	DEFAULT = “ PERFORM ”
GLINCORR	Global Non-Linearity Correction	SBC DEFAULT = “ PERFORM ”
LFLGCORR	Local and Global Non-Linearity Flagging in DQ Array	SBC DEFAULT = “ PERFORM ”

Bypassing the PHOTCORR Step

Raw ACS images are in units of *Data Numbers (DN)*, calibrated images are in units of *electrons*, and drizzled images are in units of *electrons/second*. The pipeline calibration tasks do not alter the units of the pixels in the image. Instead they calculate and write the inverse sensitivity conversion factor (PHOTFLAM) and the ST magnitude scale zero point (PHOTZPT) into header keywords in the calibrated data. Refer to the section “doPhot - Photometry Keyword Calculation” in Section 3.4.3 for more information.

When populating the photometric keywords during the PHOTCORR step, **CALACS** uses the STSDAS synthetic photometry package, **SYNPHOT**, which contains the throughput curves of all HST optical components. (For information on retrieving the **SYNPHOT** dataset, see Appendix A.3.2 of the HST Data Handbook Introduction.) The **SYNPHOT** dataset contains numerous files which may be updated on a regular basis, making it cumbersome for the user to set up. A simple alternative is to set the PHOTCORR calibration switch to OMIT in the primary header of the uncalibrated image. The user may simply copy the photometric keywords from the calibrated data into the raw image header and then run **CALACS** with the PHOTCORR switch set to OMIT.

Speed of Pipeline Processing

Reprocessing HRC or SBC data will not put a burden on most computing systems since the image sizes, both science data and reference files, are relatively small. Processing WFC observations, on the other hand, will require more computing power, including both CPU run time and disk space.

Great care has been taken to minimize the memory requirements of the pipeline software to accommodate most computing configurations. Line-by-line I/O is used during pipeline processing and is particularly useful when more than one image is operated on at a time, for example during flat fielding or co-adding images. This, unfortunately, places an extra burden on the I/O capabilities of the computing system. **CALACS** requires up to 130MB of memory to process WFC data, while **PyDrizzle** requires up to 400MB.

An Ultra-10 class workstation (300 MHz, 512MB RAM) using **CALACS** to process WFC data will take ~13 minutes for a CR-SPLIT=2 association or ~4.5 minutes for a single WFC exposure. **PyDrizzle** requires ~5 minutes to process a single WFC exposure. These times can be scaled by the processor speed of the computing system, and result in much faster run times for newer workstations.

3.5.2 CALACS Examples

We present several examples of **CALACS** reprocessing. The steps required for general **PyDrizzle** reprocessing are summarized in the “PyDrizzle Examples” on page 4-28.

Example 1: Reprocessing a Single Exposure

The following example uses HRC data from the flat field calibration program 9019 which observed the stellar cluster 47 Tucanae. The observations are from visit 07, exposure 12 and utilize the F814W filter. The exposures are CR-SPLIT into two exposures of 20 sec each. The association table for this exposure is `j8bt07020_asn.fits`. Typing ‘`tprint j8bt07020_asn.fits`’ reveals the rootnames of the individual exposures: `j8bt07oyq` and `j8bt07ozq`.

For the purposes of this first example, let us assume that the observations are *not* part of an association. This example will illustrate the steps required to reprocess a single exposure after changing the bias reference file from the default value to a file specified by the user.

1. In the Unix environment, we must specify the location of the ‘`jref`’ directory where the calibration reference files reside and then start IRAF. We assume here that the user has requested calibration files as part of the data archive request and stored these in their directory ‘`/mydisk/myjref/`’, where ‘`/mydisk/`’ specifies the directory containing the pipeline data.

(Note: This setup must be done in the same window in which IRAF will be used. Note also that setting ‘`jref`’ from within IRAF will not work even though typing ‘`show jref`’ would suggest it might.)

```
unix> setenv jref /mydisk/myjref/
unix> cl
```

2. To see what bias reference file is currently in use, we use the `hedit` task. Note that the period at the end of the line is required. The lines below beginning with the ‘`iraf>`’ symbol indicate commands typed into IRAF. Lines with no symbol indicate output from IRAF.

```
iraf> hedit j8bt07oyq_raw.fits[0] BIASFILE .
j8bt07oyq_raw.fits[0], BIASFILE = jref$m4r1753rj_bia.fits
```

3. Next, we edit the global image header of the raw image to reflect the name of the new bias file.

```
iraf> hedit j8bt07oyq_raw.fits[0] BIASFILE jref$mybias.fits verify-
j8bt07oyq_raw.fits[0], BIASFILE: jref$m4r1753rj_bia.fits->
jref$mybias.fits
j8bt07oyq_raw.fits[0] updated
```

- We then set the PHOTCORR processing step to ‘OMIT’ and copy the photometric keywords from the calibrated to the raw image. (Alternately, the user may keep track of these numbers in any other preferred manner. Most users will only require knowledge of the PHOTFLAM keyword for photometric calibration.) These steps allow users to bypass this SYNPHOT-based calibration step, due to the cumbersome requirements for setup. (See “Bypassing the PHOTCORR Step” on page 3-27 for more information.)

```

iraf> hedit j8bt07oyq_raw.fits[0] PHOTCORR omit verify-
j8bt07oyq_raw.fits[0], PHOTCORR: PERFORM -> omit
j8bt07oyq_raw.fits[0] updated
iraf> hedit j8bt07oyqflt.fits[1] phot* .
j8bt07oyqflt.fits[1], PHOTMODE = "ACS HRC F814W"
j8bt07oyqflt.fits[1], PHOTFLAM = 1.3260861E-19
j8bt07oyqflt.fits[1], PHOTZPT = -2.1100000E+01
j8bt07oyqflt.fits[1], PHOTPLAM = 8.1209805E+03
j8bt07oyqflt.fits[1], PHOTBW = 7.0114880E+02
iraf> hedit j8bt07oyq_raw.fits[1] PHOTFLAM 1.3260861E-19 verify-
j8bt07oyq_raw.fits[1],PHOTFLAM: 0.000000E+00 -> 1.3260861E-19
j8bt07oyq_raw.fits[1] updated

```

- Within the directory containing the pipeline products (“/mydisk/”), we create the subdirectory “recal”, copy the RAW and ASN files to this subdirectory, and ‘cd’ to this directory.
- Finally, we load the packages `stsdas.hst_calib.acs` and run `CALACS` on the single raw exposure.

(Note that invoking this package will result in the message: "Warning: package ‘acs’ includes Python tasks that require PyRAF". This applies only for `calacsdriz` which calls the `CALACS` and `PyDrizzle` tasks in succession.)

```

iraf> stsdas.hst_calib.acs
iraf> calacs j8bt07oyq_raw.fits

```

The product will be a single calibrated image with the FLT extension.

Example 2: Reprocessing Multiple Exposures within an Association

This example uses the same data from Example 1 and illustrates the steps required to reprocess an ACS association after changing the bias reference file from the default value to a file specified by the user. The steps

required are similar to the previous example, with a few modifications. IRAF output comments which are similar to Example 1 have been omitted.

1. First, we look at the contents of the image association.

```
iraf> tprint j8bt07020_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8bt07oyq      EXP-CRJ      yes
j8bt07ozq      EXP-CRJ      yes
j8bt07021      PROD-CRJ      yes
```

2. To see what bias reference file is currently in use, we use **hedit**.

```
iraf> hedit j8bt07*raw.fits[0] BIASFILE .
```

3. Next, we edit the global image header of all the raw images to reflect the name of the new bias file.

```
iraf> hedit j8bt07*raw.fits[0] BIASFILE jref$mybias.fits verify-
```

4. We then set the PHOTCORR processing step to ‘OMIT’ and copy the photometric keywords from the calibrated to the raw image.

```
iraf> hedit j8bt07*raw.fits[0] PHOTCORR omit verify-
iraf> hedit j8bt07*flt.fits[1] phot* .
iraf> hedit j8bt07*raw.fits[1] PHOTFLAM 1.3260861E-19 verify-
```

5. Within the directory containing the pipeline products (“/mydisk/”), we create the subdirectory “recal”, copy the RAW and ASN files to this subdirectory, and ‘cd’ to this directory.

6. Finally, we run **CALACS** on the image association.

```
iraf> stsdas.hst_calib.acs
iraf> calacs j8bt07020_asn.fits
```

(If executing this command in the same directory in which you have run the previous example, then one of the FLT files will already exist and **CALACS** will not overwrite existing images. To execute in this case, first delete the existing FLT files or move them to a separate directory.)

The product will be two separate calibrated images with the FLT extension (j8bt07oyqflt.fits, j8bt07ozqflt.fits) and a single cr-combined image with the CRJ extension (j8bt07021_crj.fits).

Example 3: Removing Cosmic-rays from Repeated (RPT-OBS) Exposures

The following example uses HRC data from the shutter stability calibration program 9662 which observed the stellar cluster 47 Tucanae. The observations are from visit 01, exposure 20 and utilize the CLEAR filter. The two repeated (RPT-OBS) exposures are 1.0 sec each.

In this example, we illustrate the steps required to remove cosmic-rays from repeated observations which use the RPT-OBS syntax. By definition, the standard calibration product of multiple repeated exposures is a summed image. We can fool the software into rejecting cosmic-rays from the images, but this requires modification of the image association table.

1. First, we look at the contents of the image association.

```

iraf> tprint j8is01020_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8is01j2q      EXP-RPT      yes
j8is01j3q      EXP-RPT      yes
j8is01021      PROD-RPT     yes

```

2. Update calibration switches in the global header using the **hedit** task.

```

iraf> hedit *raw.fits[0] CRCORR perform add-
iraf> hedit *raw.fits[0] RPTCORR omit add-

```

3. Update association table MEMTYPE column, by changing each instance of RPT to CR. This may be done using the text editor functionality of **tedit**.

```

iraf> tedit j8is01020_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8is01j2q      EXP-CRJ      yes
j8is01j3q      EXP-CRJ      yes
j8is01021      PROD-CRJ     yes

```

4. We then set the PHOTCORR processing step to ‘OMIT’ and copy the photometric keywords from the calibrated to the raw image.

```

iraf> hedit j8is01*raw.fits[0] PHOTCORR omit verify-
iraf> hedit j8is01*flt.fits[1] phot* .
iraf> hedit j8is01*raw.fits[1] PHOTFLAM 4.5803514E-20 verify-

```

5. Within the directory containing the pipeline products (“/mydisk/”), we create the subdirectory “recal”, copy the RAW and ASN files to this subdirectory, and ‘cd’ to this directory.
6. Finally, we run **CALACS** on the image association.

```

iraf> stsdas.hst_calib.acs
iraf> calacs j8is01020_asn.fits

```

The product will be two calibrated images with the FLT extension (j8is01j2qflt.fits, j8is01j3qflt.fits) and a cr-combined image with the CRJ extension (j8is01021crj.fits) instead of the usual SFL extension.

Example 4: Combining Exposures from Multiple Associations

This example uses the same data from Example 3 plus an additional exposure from the same program: visit 01, exposure 40. These additional data also use the CLEAR filter and contain two repeated 1.0 sec exposures. The association table for this exposure is j8is01040_asn.fits. In this example, we illustrate the steps required to combine two separate sets of repeated observations.

1. First we merge the two association tables and edit them appropriately (using **tmerge** and **tedit**, respectively).

```

iraf> tmerge j8is01020_asn.fits,j8is01040_asn.fits \
>>>  outtable="merged_asn.fits" option=append
iraf> tprint merged_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8is01j2q      EXP-RPT      yes
j8is01j3q      EXP-RPT      yes
j8is01021      PROD-RPT     yes      <---remove this line
j8is01j8q      EXP-RPT      yes
j8is01j9q      EXP-RPT      yes
j8is01041      PROD-RPT     yes      <---edit this line
iraf> tedit merged_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8is01j2q      EXP-RPT      yes
j8is01j3q      EXP-RPT      yes
j8is01j8q      EXP-RPT      yes
j8is01j9q      EXP-RPT      yes
j8is01xx1      PROD-RPT     yes      <---new product name

```


2. We then set the PHOTCORR processing step to ‘OMIT’ and copy the photometric keywords from the calibrated to the raw image using **hedit**.

```

iraf> hedit j8is01*raw.fits[0] PHOTCORR omit verify-
iraf> hedit j8is01*flt.fits[1] phot* .
iraf> hedit j8is01*raw.fits[1] PHOTFLAM 4.5803514E-20 verify-

```

3. Within the directory containing the pipeline products (“/mydisk/”), we create the subdirectory “recal”, copy the RAW and ASN files to this subdirectory, and ‘cd’ to this directory.
4. Finally, we run **CALACS** on the new image association.

```

iraf> stsdas.hst_calib.acs
iraf> calacs merged_asn.fits

```

The product will be 4 calibrated images with the FLT extension and a single summed image with the SFL extension (j8is01xx1_sfl.fits).

Example 5: Reprocessing Images taken as part of a Dither Pattern

The following example uses WFC data from the GOODS program 9425. These observations are from visit 54, exposure 219 and target the CDF-South with the F606W filter. The images are part of a 2-point line dither pattern and are 480 sec each. In this example, we illustrate the steps required to reprocess data which is part of a dither pattern, using a non-default dark reference file. The steps are similar to Example 2, but the format of the association and the data products are unique.

1. First, we look at the contents of the image association.

```

iraf> tprint j8e654010_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8e654c0q      EXP-DTH      yes
j8e654c4q      EXP-DTH      yes
j8e654011      PROD-DTH     yes

```

2. To see what dark reference file is currently in use, we use **hedit**.

```

iraf> hedit j8e654*raw.fits[0] DARKFILE .

```

3. Next, we edit the global image header of all the raw images to reflect the name of the new dark reference file.

```

iraf> hedit j8e654*raw.fits[0] DARKFILE jref$mydark.fits verify-

```

4. We then set the PHOTCORR processing step to ‘OMIT’ and copy the photometric keywords from the calibrated to the raw image.

```
iraf> hedit j8e654*raw.fits[0] PHOTCORR omit verify-  
iraf> hedit j8e654*flt.fits[1] phot* .  
iraf> hedit j8e654*raw.fits[1] PHOTFLAM 7.7792272E-20 verify-
```

5. Within the directory containing the pipeline products (“/mydisk/”), we create the subdirectory “recal”, copy the RAW and ASN files to this subdirectory, and ‘cd’ to this directory.
6. Finally, we run **CALACS** on the image association.

```
iraf> stsdas.hst_calib.acs  
iraf> calacs j8e654010_asn.fits
```

The product of the DITH-PATTERN will be two separate calibrated images with the FLT extension. In subsequent processing (see Chapter 4 for details), **PyDrizzle** will combine these FLT files into a single DRZ image (j8e654011_drz.fits).

Distortion Correction and Drizzling of ACS Images

In this chapter. . .

4.1 ACS Geometric Distortions / 4-2
4.2 Combination of Dithered Data / 4-14
4.3 PyDrizzle / 4-17
4.4 Reprocessing with PyDrizzle / 4-20
4.5 MultiDrizzle / 4-34
4.6 MultiDrizzle Example / 4-37

This chapter describes the geometric distortion of the ACS and presents the available tools and methods for removing the distortion and combining dithered datasets into clean, undistorted, and photometrically faithful output products.

The first section describes how the distortion has been quantified and how this information is held in reference files which are used by the calibration pipeline. Plots of the distortion field are given for the three ACS detectors.

The remaining sections give an overview of the drizzle method and discuss the higher level **PyDrizzle** and **MultiDrizzle** software tools available for distortion correction and image combination.

4.1 ACS Geometric Distortions

ACS produces the largest single images ever taken with HST. However, because the optics were designed with a minimum number of components, ACS focal surfaces are far from normal to the principal rays. This results in an image of the sky which is distorted in a large, but predictable manner. The distortion consists of two effects, discussed previously in Section 1.1.3. The first is the elongation of the ACS apertures, causing the pixel scale to be smaller along the radial direction of the Optical Telescope Assembly (OTA) field of view than along the tangential direction. The second effect is the variation of pixel area across the detector.

The results presented in the discussion below are derived from multiple observations of dense star fields with the images displaced from each other by amounts comparable with the size of the field of view. The following discussion is based on reports from the HST Calibration Workshop Proceedings: *Calibration of Geometric Distortion in the ACS Detectors* (Meurer et al. 2002) and *Astrometry with the Advanced Camera: PSFs and Distortion in the WFC and HRC* (Anderson 2002) at:

http://www.stsci.edu/hst/HST_overview/documents/calworkshop/workshop2002/CW2002_TableOfContents

4.1.1 Distortion Model

The heart of the distortion model relates detector pixel position (x, y) to sky position (see ACS ISR 2000-11) using a polynomial transformation given by:

$$x_c = \sum_{i=0}^k \sum_{j=0}^i a_{i,j} (x - x_r)^j (y - y_r)^{i-j}$$

$$y_c = \sum_{i=0}^k \sum_{j=0}^i b_{i,j} (x - x_r)^j (y - y_r)^{i-j}$$

where k is the polynomial order of the fit, (x_r, y_r) is the reference pixel position taken to be the center of each detector or WFC chip, and (x_c, y_c) is the undistorted position in arcseconds. The coefficients to the fits, $a_{i,j}$ and $b_{i,j}$, are free parameters. A similar form is used for the inverse relation which provides the conversion from units of arcseconds to distorted input pixels.

For the WFC, an offset is applied to put the two CCD chips on the same coordinate system:

$$X = x_c + \Delta x(chip), \quad Y = y_c + \Delta y(chip).$$

The offsets $\Delta x(chip)$, $\Delta y(chip)$ are 0,0 for chip 1 and correspond to the separation between chips 1 and 2 for chip 2. The chip 2 offsets are free parameters in the fit. (X, Y) correspond to tangential plane positions in

arcseconds which are tied to the HST ($V2, V3$) coordinate system. Finally, all frames are transformed to the same coordinate grid on the sky:

$$X_{sky} = \Delta X_i + \cos(\Delta\theta_i)X' - \sin(\Delta\theta_i)Y',$$

$$Y_{sky} = \Delta Y_i + \sin(\Delta\theta_i)X' + \cos(\Delta\theta_i)Y',$$

where the free parameters $\Delta X_i, \Delta Y_i, \Delta\theta_i$ are the position and rotation offsets of frame i .

There is one added complication in the scale term of the geometric distortion due to the effect of differential velocity aberration. The apparent position on the sky of any targets is displaced by the effect of velocity aberration. The speed of the earth in its orbit is about 30 km/sec. Figure 4.1 illustrates a target which is viewed at an angle α with respect to the velocity of the telescope. The apparent position of the target is at angle α' where:

$$\tan \alpha' = \frac{\sqrt{1 - \beta^2} \cdot \sin \alpha}{\cos \alpha + \beta}$$

and β is the net telescope velocity divided by the speed of light. The change in angle can amount to tens of arcseconds, but the HST flight software continually accounts for this and keeps the apparent position of the target at a fixed aperture reference point. However a secondary differential effect is a small change in the scale of magnitude equal to:

$$\frac{\sqrt{1 - \beta^2}}{1 + \beta \cos \alpha}$$

The scale change is of order 1 part in 10^4 . For targets near the ecliptic, this can cause a misalignment between different exposures of up to 1.4 pixels in the ACS field of view in WFC images taken 6 months apart. This scale change, illustrated in Figure 4.2, can complicate the removal of cosmic rays from images and can cause artifacts when constructing mosaics.

The effective scale change in arcsec per pixel is actually the reciprocal of the factor given above, i.e., if the apparent positions appear closer together than a single pixel sees a larger portion of the sky. The ACS science images now include a keyword VAFactor equal to the approximation $(1 + \beta \cos \alpha)$. This keyword is used by the **PyDrizzle** software when correcting for geometric distortion.

For a further discussion of the effect of velocity aberration, see *The Effect of Velocity Aberration Correction on ACS Image Processing*: HST Calibration Workshop Proceedings (Cox & Gilliland 2002) at:

http://www.stsci.edu/hst/HST_overview/documents/calworkshop/workshop2002/CW2002_Papers/CW02_cox

Figure 4.1: Schematic illustration of the angular change due to velocity aberration.

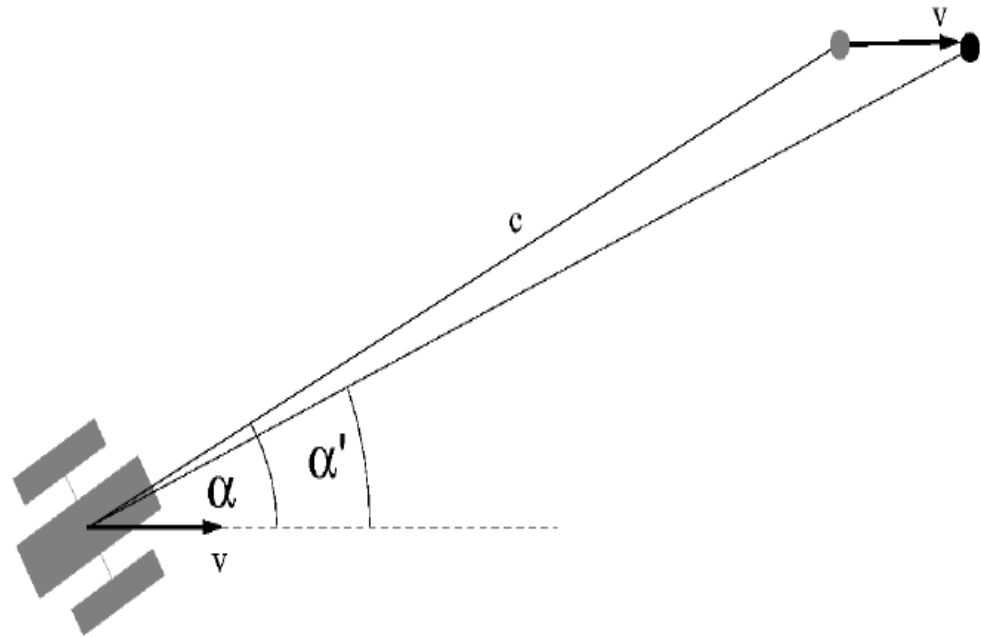
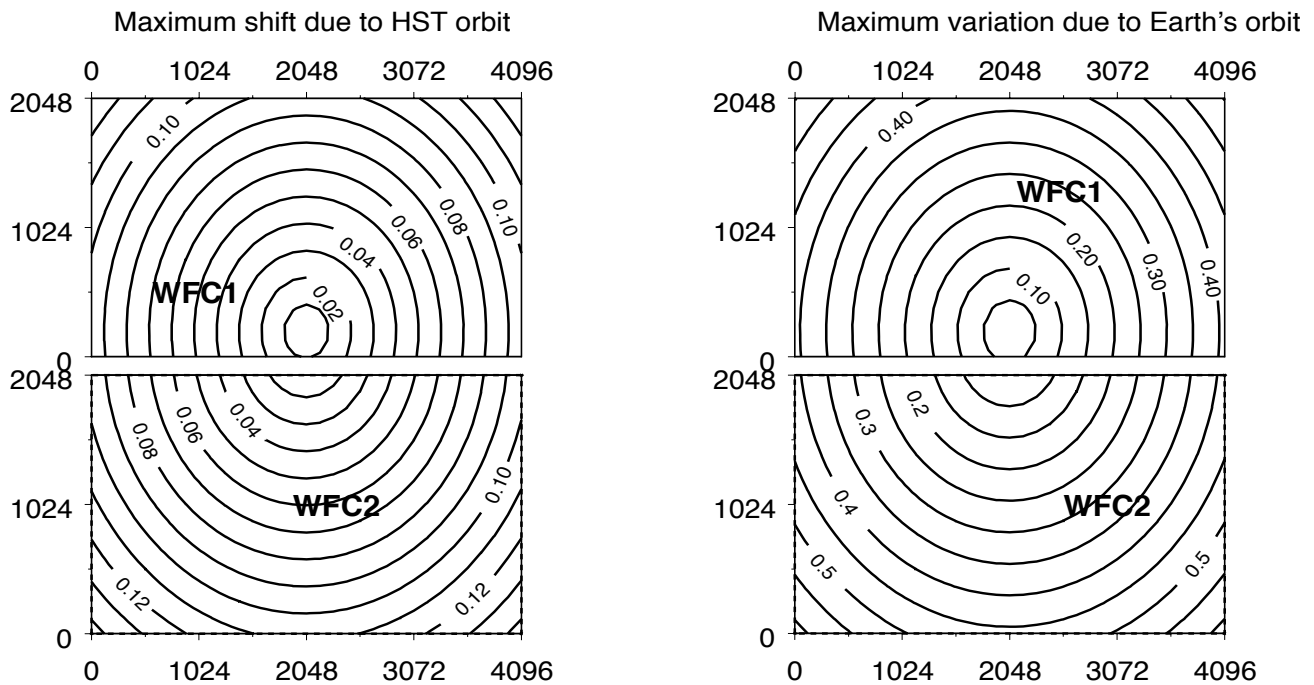


Figure 4.2: Pixel shifts on WFC caused by scale changes from differential velocity aberration.

PIXEL SHIFTS



4.1.2 On-orbit Calibration Programs

The positions of stars observed multiple times in the dithered star fields are used to iteratively solve for the free parameters in the distortion

solution: fit coefficients $a_{i,j}$, $b_{i,j}$; chip 2 offsets Δx , Δy (WFC only); frame offsets ΔX_i , ΔY_i , $\Delta \theta_i$; and tangential plane position X_{sky} , Y_{sky} of each star used in the fit.

The original ACS geometric distortion campaign observed the core of 47 Tucanae with the WFC and HRC (program 9028, Meurer). The exposures were designed to detect stars on the main sequence turn-off, allowing a high density of stars with relatively short exposures. The F475W filter (Sloan g') was used to minimize the number of saturated red giant branch stars in the field. For calibrating the distortion in the SBC, exposures of NGC6681 (program 9027, Hartig) were chosen for the relatively high density of UV emitters (hot horizontal branch stars).

Initially the distortion analysis was performed only with the F475W filter for both WFC and HRC detectors. Further measurements and analysis have provided solutions for 10 HRC filters and 12 WFC filters. Scale variations of a few parts in 10^4 are found between regular filters but changes of a percent or more are evident when polarizing filters are included.

Originally, only images taken with a single roll angle were used to define the distortion solutions. The solution using only these data is degenerate in the zeroth (absolute pointing) and linear terms (scale, skewness). Therefore the largest commanded offsets with a given guide star pair were used to set the linear terms. However, comparison of corrected coordinates to astrometric positions showed that residual skewness in the solution remained. Hence, as of November 2002, the results for the WFC and HRC are based on data from multiple roll angles, incorporating data obtained from the calibration outsourcing program 9443 (King & Anderson). The overall plate scale is set by the largest commanded offset. For the HRC, the linear scale is set by matching HRC and WFC coordinates, since the same field was used in the SMOV observations. The zeroth order terms (position of ACS apertures in the V2,V3 frame) were determined from observations of an astrometric field.

4.1.3 Results

The most obvious component of the ACS distortion is a marked skew which results in the X/Y detector axes lying at ~ 85 degrees to each other on the sky. In addition, the distortion in all ACS detectors is highly non-linear (see Figure 4.3-4.6). A quartic fit ($k=4$) is adequate for characterizing the distortion to an accuracy much better than 0.2 pixels over the entire field of view.

WFC

The WFC detector is tilted at 22 degrees giving an elongation of 8% along the diagonal. The non-square projected aperture shape is evident from Figure 1.1 in Chapter 1, where the x axis is approximately in the V2 direction and the y axis is in the -V3 direction.

The WFC distortion is also illustrated in Figure 4.3, a vector displacement diagram which shows the contribution of the non-linear part

of a quartic fit to the data. The vectors represent the degree of distortion to be expected in the WFC beyond the directional dependence of the plate scale. For display, the vectors are magnified by a factor of 5 compared to the scale of the x and y axes. The largest displacement indicated at the top left corner of the figure is ~ 82 pixels or about 4 arcseconds. While a quartic solution is adequate for most purposes, binned residual maps (Figure 4.6) show that there are significant coherent residuals in the WFC solutions which have amplitudes up to ~ 0.1 pixels. These residuals are not currently accounted for in pipeline calibration, although efforts are underway to do so in the future (see Section 4.1.4).

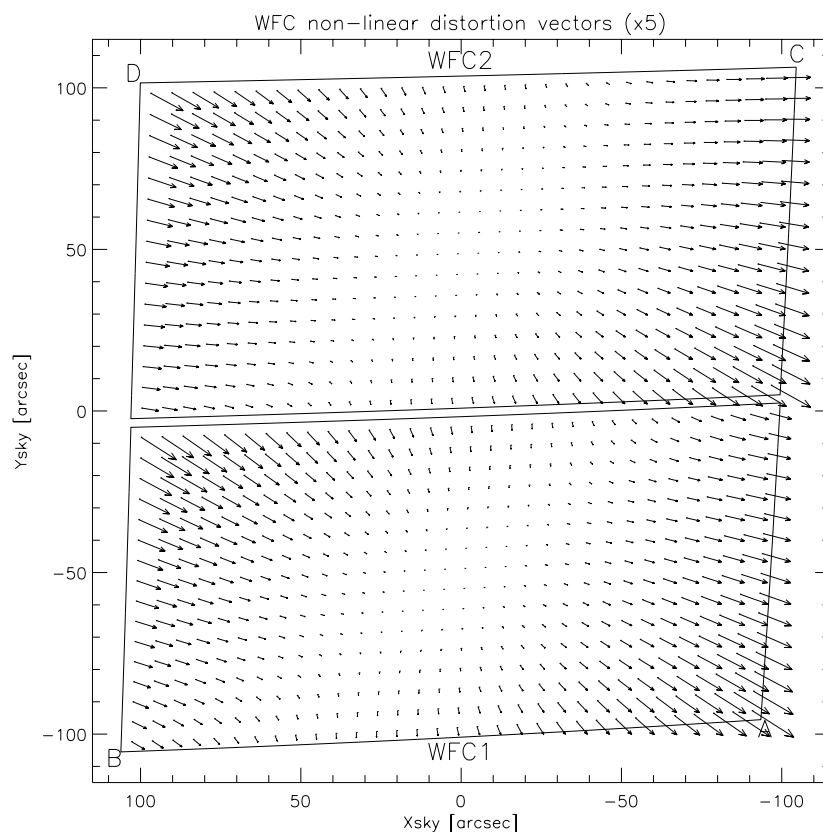
The pixel scales and array sizes are summarized in Table 4.1 for each ACS detector. A contour plot of relative pixel area across the WFC is shown in Figure 6.1. The range of area is from 0.89 to 1.08 times the central value.

When deriving the ACS flat fields, an implicit correction is made for the distortion which is equivalent to dividing each pixel by the relative pixel area. Thus, point source photometry extracted from a flat fielded image must be multiplied by the effective pixel area. This correction is accounted for in pipeline processing by **PyDrizzle** which uses the distortion solution to correct all pixels to equal areas. When working with FLT files this correction must be applied manually (see Section 6.1.3 for details).

Table 4.1: Pixel scale (*arcsec/pixel*) and array size (*pixels*) for each ACS detector.

Detector	Pixel Scale at Center $X_{\text{scale}}, Y_{\text{scale}}$	Average Pixel Scale $X_{\text{scale}}, Y_{\text{scale}}$	Array size (pixels)	Array size (arcsec)
WFC1	0.04927, 0.04864	0.04917, 0.04861	4096x2048	201"x100"
WFC2	0.04985, 0.05031	0.04976, 0.05029	4096x2048	204"x103"
HRC	0.02844, 0.02484	0.02843, 0.02484	1024x1024	29"x25"
SBC	0.03374, 0.03009	0.03377, 0.03010	1024x1024	35"x31"

Figure 4.3: Non-linear component of the ACS distortion for the WFC detector. Note that this figure is rotated 180 degrees with respect to the pipeline calibration products, where WFC2 is the lower half of the detector.



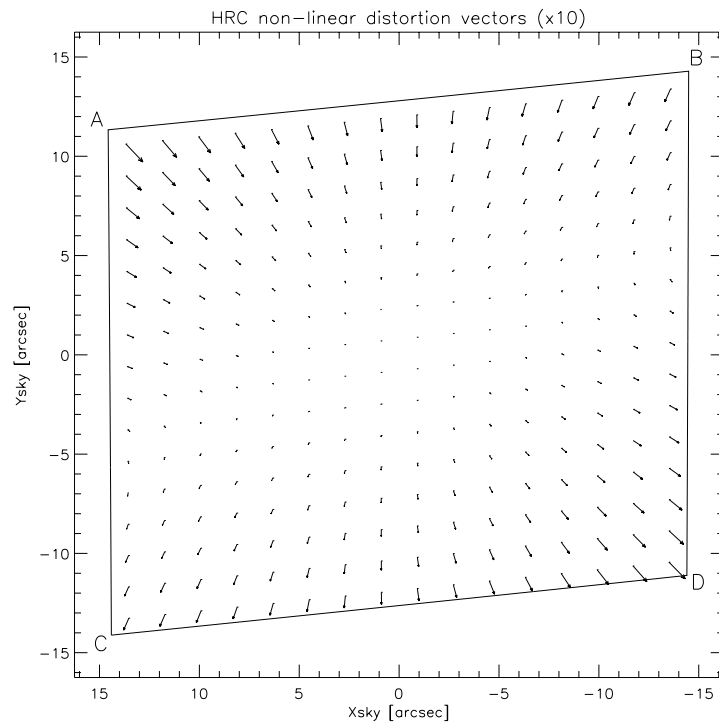
HRC

The High Resolution Channel has its edges aligned approximately along the V2 and V3 axes (see Figure 1.1). In this case, the center of the aperture lies on a line passing through the V2/V3 origin and making an angle of 22 degrees with the V3 axis. Because the aperture diagonal does not correspond to a radius of the HST field of view, the distortion has no particular symmetry with respect to the detector axes. Because the focal plane, and therefore the detector plane is 25 degrees away from the plane normal to the light path, the scales along the axes differ by $\sim 14\%$. However, since the HRC is less than 30 arcsec across, the *total* variation in scale across the detector is much less than for the WFC, being only about 1%.

A vector plot of the deviation from linearity is given in Figure 4.4 in which the deviations have been magnified by a factor of 10 for illustrative purposes. The largest displacement is 4.9 pixels in the top left corner and corresponds to about 0.1 arcsec. While a quartic solution is adequate for most purposes, binned residual maps (Figure 4.6) show that there are significant coherent residuals in the HRC solutions which have amplitudes up to ~ 0.1 pixels. These residuals are not currently accounted for in pipeline calibration, although efforts are underway to do so in the future (see Section 4.1.4).

The pixel scales and array sizes are summarized in Table 4.1 for the HRC. The variation of pixel area across the detector is shown in Figure 6.1. The maximum deviation from the central value is just over 2%. As explained for WFC, this must be taken into account when doing photometry, either by using **PyDrizzle** output products (DRZ files) or by manually correcting FLT files with a pixel area map (see Section 6.1.3 for details).

Figure 4.4: Non-linear component of the ACS distortion for the HRC detector.



SBC

The Solar Blind Channel contains the MAMA detector. It is centered at approximately the same place as the HRC in the V2/V3 plane and is slightly larger, about 35 x 30 arcsec, versus 29 x 25 arcsec for the HRC. The average scales in the x and y directions are given in Table 4.1 and are calculated by matching observations of the same sky area by HRC and SBC.

The optical distortions are close to those displayed by the HRC. The only difference in their light paths is the presence of a plane M3 fold mirror which reflects light away from the SBC onto the HRC. Because there is one less mirror in the SBC light path, the distortion corrected image (in sky coordinates) in Figure 4.5 appears flipped with respect to the HRC image in Figure 4.4. The MAMA has a small amount of extra distortion in the detector itself, arising from irregularities in the multi-channel plate. The largest difference from a square pattern is 2 pixels, or 0.06 arcsec. As explained for WFC, pixel area variations must be taken into account when doing photometry, either by using **PyDrizzle** output products (DRZ files) or by manually correcting FLT files with a pixel area map (see Section 6.1.3 for details).

Figure 4.5: Non-linear component of the ACS distortion for the SBC detector.

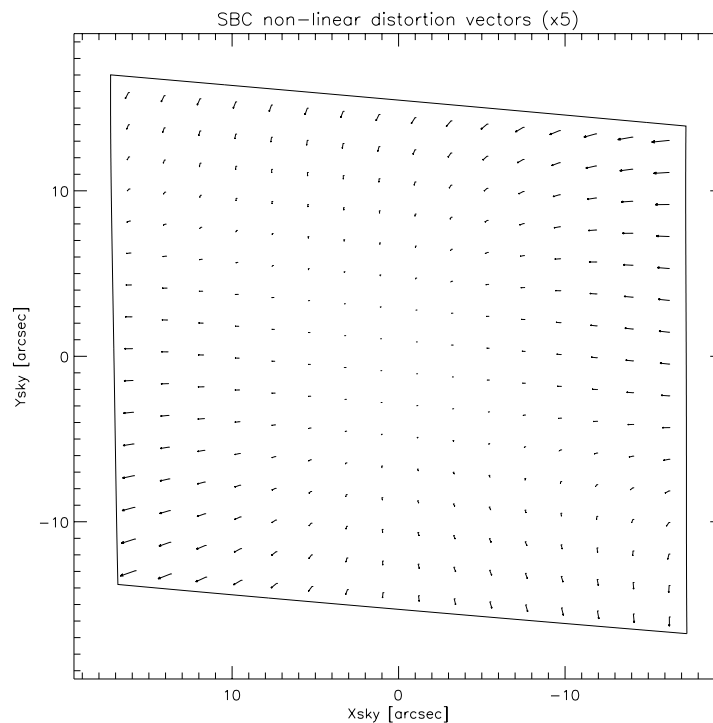
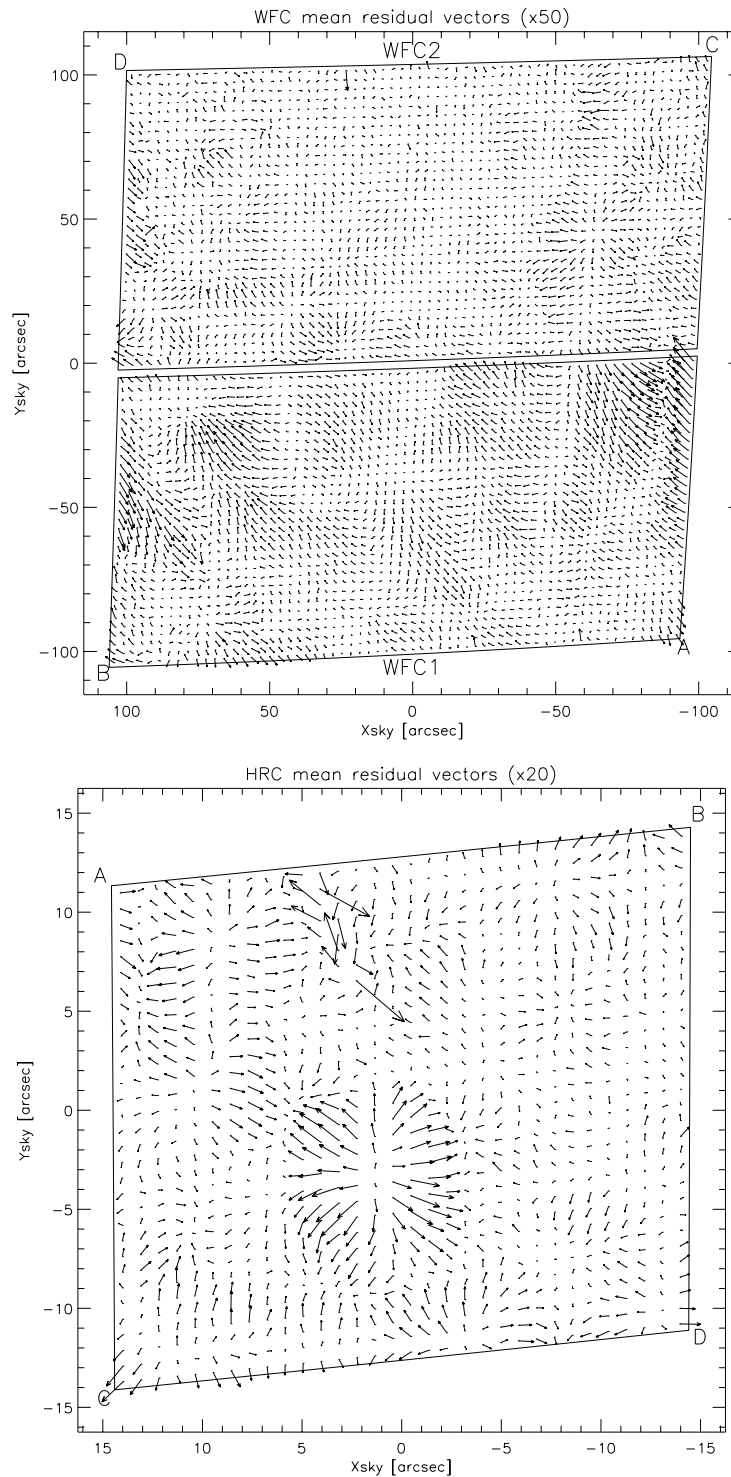


Figure 4.6: Binned residuals to quartic distortion model fits for the ACS WFC and HRC detectors. Note that the WFC figure is rotated 180 degrees with respect to the pipeline calibration products, where WFC2 is the lower half of the detector. The large residuals close to the upper edge of the HRC map, left of center, are due to the presence of the Fastie occulting finger. There is insufficient data to make the equivalent plot for the SBC.



4.1.4 Distortion Table (IDCTAB)

The Science Instrument Aperture File (SIAF) contains position and scale information for every aperture of each of HST's science instruments. Thus, it supports accurate target acquisitions, image processing, and photometry. The conversion between distorted and undistorted positions in the SIAF is controlled by a polynomial expansion which is derived by the individual instrument teams. A simple, common reference file can therefore be created for each instrument which describes the geometric distortion of each detector. In this section, we describe the format of this common reference file which has been developed to be as general as possible so that it can be used for many different instruments or detectors.

For ACS, the coefficients to the polynomial fit are found in the IDCTAB reference file. The distortion correction is not a part of **CALACS**. A separate task, **PyDrizzle** (see Section 4.3), converts the distortion model in the IDCTAB into a form which is used by **drizzle** to produce geometrically corrected images.

Application of the Model

The steps performed in the application of the IDCTAB can be summarized in the following steps:

- First, the offset in position of each pixel relative to the reference position is computed.
- These deltas are used to compute the undistorted positions in arcseconds with the coefficients of the polynomial fit given in the IDCTAB reference file.
- The undistorted positions are divided by the desired output plate scale, where the default value is taken from the SCALE column in the IDCTAB unless specified by the user.
- Finally, both chips (WFC only) are combined into a single image by adding the appropriate offset to each pixel to get the final position.

The fit for each chip ensures that the distorted and corrected Y axes for a chip remain at the same angle with respect to the telescope V3 axis. This allows each chip to have a different rotation relative to the other, though this effect has been empirically shown to be less than 0.001 degree.

Format of the IDCTAB Reference file

The IDCTAB is stored as a FITS binary table (calibration file with '_idc.fits' extension), where the primary header contains keywords INSTRUME and DETECTOR which define the model appropriate to a specific instrument and detector. The NORDER keyword gives the order of the polynomial stored in the table. While the original laboratory data provided a cubic solution to the distortion, more accurate in-flight data now supports a fourth-order solution (NORDER=4).

The first extension of the IDCTAB contains the geometric distortion solution, where the table columns are listed in Table 4.2. The table contains distinct rows for each detector chip and filter combination specified by the DETCHIP and FILTER columns. If the DIRECTION column is 'INVERSE' rather than 'FORWARD', the coefficients specify the conversion from undistorted to distorted pixel positions. The SCALE keyword gives the default pixel scale of the drizzled image, where the default output scale for drizzled WFC images is 0.05 arcsec/pixel and for HRC/SBC images is 0.025 arcsec/pixel.

Table 4.2: Columns in the IDCTAB

Column	Description
DETCHIP	ID of chip/detector used for observation
DIRECTION	Application direction of coefficients (FORWARD,INVERSE)
FILTER1	Name of filter in filter wheel 1
FILTER2	Name of filter in filter wheel 2
XSIZE	Raw image size in X direction (pixels)
YSIZE	Raw image size in Y direction (pixels)
XREF	X position of reference pixel
YREF	Y position of reference pixel
V2REF	V2 position of reference point (arcsec)
V3REF	V3 position of reference point (arcsec)
SCALE	Scale of square corrected pixel (arcsec/pix)
CX10, CX11,...	Distortion Coefficients for X position
CY10, CY11,...	Distortion Coefficients for Y position

The distortion coefficients provide a correction for each pixel relative to the detector chip reference point specified in the XREF and YREF columns. The relationship between each chip must be factored in separately. HST relies on a V2/V3 coordinate system to define each detector's position (in arcseconds) relative to the center of HST's field of view. The V2REF and V3REF columns contain the reference position of each detector in the V2/V3 system. The absolute position is secondary to the relative position of this reference point, because only the offset between V2REF/V3REF positions is used for combining multiple chips into a single output image. The XREF/YREF and V2REF/V3REF positions represent the same point in different coordinate systems.

The columns CX_{ij} and CY_{ij} contain the coefficients of the polynomial fit for each chip and convert an input pixel position to an undistorted

position. The column names contain the indices for the coefficients from the polynomial fit, where CX11 and CY11 correspond to the a_{11} and b_{11} coefficients. For subarray data, the input pixel position must be corrected for the subarray offset in order to put it in the coordinate frame of the full chip.

The presently implemented IDC tables include separate rows of coefficients for all the measured filter combinations. Uncalibrated combinations default to the F475W filter solution with scale allowances made in the case of polarizing filters.

The solutions described by the IDC tables are correct to about 0.1 pixels. However the distortion can be specified to 0.01 pixels by the inclusion of extra corrections (such as those shown in Figure 4.6) which can not be expressed by the fourth order polynomials. These further corrections are available in the form of two images for each detector for the x and y directions. We have given these images the extension DXY. These correction images are planned to become part of the pipeline process in late 2004. For filter combinations for which these images do not exist, the final DXY image correction step will be omitted. However note that for most purposes the polynomial solutions provide adequate precision.

4.2 Combination of Dithered Data

4.2.1 Introduction to Dithering

Many ACS datasets are dithered - they have shifts between successive images which may be small or large. Dithering is a well established technique for HST imaging and has many advantages. For a general introduction to the concepts behind dithering, please consult the [HST Dither Handbook](#) (Koekemoer et al. 2002).

The primary motivations for dithering are:

- To allow the detection and flagging of image artifacts. Cosmic rays may be detected by taking multiple CR-SPLIT images at the same position, but other artifacts such as hot pixels require exposures with small dithers if they are to be detected and flagged. In addition, small-scale dithers are useful for filling in regions not covered by the detector, like the WFC chip gap or the HRC occulting finger.
- To improve the sampling of the PSF. In many ACS observation modes the pixel grid spacing is inadequate to fully sample the optical PSF falling on it. In these cases sub-pixel dithering will allow some of this lost information to be regained and the subsequent reconstruction of an improved final image.

- Larger dithers are needed to create mosaics of larger regions of sky or to cover the gap between the two chips of the ACS/WFC.

Because the non-linear component of the ACS optical distortion is large, dither shifts will vary across the field of view. This must be taken into account when planning observations. For more information on ACS dither patterns, see the [ACS Patterns](#) section of the HST Phase II proposal instructions and additional variations on the ACS “dither” webpage:

<http://www.stsci.edu/hst/acs/proposing/dither>.

4.2.2 Introduction to Drizzling

The methods for combining dithered ACS data which are described in this chapter are all based on the drizzle method (Fruchter & Hook, 2002 *PASP*, 114, 144) and are described in detail in the *HST Dither Handbook* (Koekemoer et al. 2002). Here we give a brief summary of the method and refer the reader to the aforementioned paper for further details and quantitative information about its properties.

Drizzle maps each pixel of the input image onto pixels in a sub-sampled output image, taking into account shifts and rotations between images and the optical distortion of the camera. However, in order to avoid re-convolving the image with the large pixel of the camera, the user may shrink the pixel before it is averaged into the output image. This shrinking factor is known as the “pixfrac” and is normally between 0.0 (corresponding to interlacing input pixels onto single output ones) and 1.0 (which corresponds to shift-and-add). The degree of sub-sampling (the size of the output pixels relative to that of the original ones) is also a user controllable parameter called “scale” which is typically in the range 1-0.5 for the ACS.

This algorithm has a number of advantages. It preserves both absolute surface and point-source photometry, hence flux can be measured using an aperture whose size is independent of position on the chip. And because the method anticipates that a given output pixel may receive no information from a given input pixel, missing data (due for instance to cosmic rays or detector defects) do not cause a substantial problem, so long as there are enough dithered images to fill in the gaps caused by these zero-weight input pixels. Finally, the linear weighting scheme is statistically optimum when inverse variance maps are used as the input weights.

The original drizzle method mapped a shrunken version of each input pixel onto the output and distributed weight according to the overlaps with the output pixel grid. For the case of multiple sub-pixel dithers of significantly undersampled images, this was close to optimum. However, for some purposes this “kernel” is not ideal and the latest implementations of drizzle allow the user to select from other options. For example, if the original method is applied to a single image, where pixfrac and scale are

both normally unity, the method becomes equivalent to bilinear interpolation. This results in an output image having strongly correlated noise and also degraded resolution. For images from the ACS in the red, even the WFC channel is close to critically sampled, rather than strongly undersampled, and it would appear to be useful to consider more optimal interpolators as drizzle kernel options. Some of these are now implemented in **drizzle** and may also be used via the higher-level scripts described. The **drizzle** help file lists all the options. The “Lanczos3” kernel is particularly recommended as it minimizes apparent noise correlation in the output image and maximizes resolution. It is, however, significantly slower.

The basic drizzle method is implemented as an IRAF task within the STSDAS **dither** package which is invoked by **PyDrizzle** or **MultiDrizzle**.

4.2.3 Overview of Drizzling Software Tools

The tools for the combination of dithered HST images have evolved through the years, starting with those developed for the Hubble Deep Field in late 1995. First came low level tools which required considerable effort to use. Subsequently higher level wrapper scripts, written in the Python language, have been developed to ease such combination. Some of these capacities are now available within the pipeline. The main three levels applicable to current ACS imaging are as follows:

- The low level tools for mapping pixels from input to output images and support tasks for measuring shifts, detecting cosmic ray events, etc. are available within the dither package of STSDAS. These include **drizzle** itself and **blot** (reverse **drizzle**). The user will now probably prefer to access these tools via one of the wrappers rather than use them directly.
- **PyDrizzle** and associated Python scripts are layered on these and automate some of the processing. They convert the IDCTAB information into a form suitable for drizzle, create appropriate association tables and drizzle images into an appropriate output frame. These functions are implemented in the pipeline and are described in detail below. The user will again probably not use these tools directly. **PyDrizzle** is available within the STSDAS system and requires the PyRAF environment.
- The third level is the **MultiDrizzle** script which is again layered on top of the above tools. It aims to provide automated image combination, including the detection and flagging of cosmic rays. Although **MultiDrizzle** can be run using its default settings to do a good combination for many typical ACS dithered datasets, it also allows the

user considerable control and flexibility. **MultiDrizzle** is available as a beta-test version but will be implemented in the STSDAS system in 2004. Like **PyDrizzle**, it requires the PyRAF environment.

We now briefly describe the pipeline software **PyDrizzle** and several associated tools. A detailed description of **MultiDrizzle** follows, as this is the recommended interactive tool for drizzling ACS images.

4.3 PyDrizzle

4.3.1 Application by the Pipeline

The ACS calibration pipeline currently consists of 2 processing stages, **CALACS** and **PyDrizzle**, which are run one after the other. Standard **CALACS** processing includes bias and dark subtraction, removal of the overscan regions, and flat fielding. It will produce, among other products, a calibrated image with the FLT suffix for each input file. The **CALACS** portion of the pipeline will calibrate and combine CR-SPLIT images. It will not, however, correct for geometric distortion or combine images from multiple dither positions.

Geometric distortion is removed for all images (dithered or not) in the second stage of processing via **PyDrizzle**. This software understands the ACS association tables, allowing the pipeline to combine associated dithered observations into a single image. These images will still contain cosmic-rays unless CR-SPLIT observations were specified within the dither pattern. For WFC observations, both chips are combined into a single extension. Finally, **PyDrizzle** converts the data to units of count rate (*electrons/sec*).

The IRAF **dither** package has provided users the basic tools for linear image combination via the task **drizzle**. This task can produce mosaics, correct for geometric distortion, and even remove cosmic rays, creating a single, undistorted output image with uniform photometric and astrometric properties. A significant amount of work, however, is required to determine the necessary input parameters, causing **drizzle** to be fairly user-intensive. These steps include:

- requiring the offsets in x and y from the reference image as input.
- requiring the output image size and pixel scale as input.
- only working with a single image extension at a time.
- requiring the rotation of each image extension as input.

These requirements made it impossible to use **drizzle** in a pipeline environment to automatically process images.

PyDrizzle was written to automate the use of **drizzle** in the pipeline environment. It can also simplify using **drizzle** for manual image

reprocessing. Pipeline use relies on default parameter values which are taken from the IDCTAB reference file and include the default plate scale and the distortion model. This file provides information about the position of each ACS detector on the sky, allowing **PyDrizzle** to combine images from multiple detectors at once. **PyDrizzle**, like **drizzle**, works with count rates, allowing images with differing exposure times to be combined easily. Regardless of whether observations were taken as a single image or a set of dithered exposures, **PyDrizzle** produces images which are both astrometrically and photometrically accurate.

Processing comments are recorded in the trailer file for the DRZ image, including which version of **drizzle** was used, what parameters were used, and which images were combined (if dithered). The same default parameters are used for all observations in the pipeline. These parameters were chosen to avoid introducing any scale changes, shifts, or rotations relative to the original pointing, aside from those corrections incorporated in the distortion model itself. While the pipeline products will be properly corrected for distortion, the combination of dithered observations may not be ideal, and small pointing errors or defects, such as hot pixels or cosmic-rays, may still be present.

4.3.2 Association Tables

Before the **drizzle** task could be used in an automated manner, two obstacles had to be overcome: calculating the required input parameters and working with a wide range of input image formats. **PyDrizzle** relies on several features to address these problems:

- An association table will relate multiple input images to an output product.
- Python “structures” will be created in memory for each input image and for the output field.
- Output field parameters (RA/Dec of center, pixel size, field of view) will be specified using a ‘SkyField’ function.

The calibration pipeline uses association tables to define how multiple images, taken at different pointings, relate to a single output image. **PyDrizzle** relies on these association tables to automatically combine and correct ACS dithered observations or any other related set of observations.

Each input/output exposure within **PyDrizzle** knows about:

- its own world coordinate system,
- the distortion model for its detector,
- the range of output pixels covered by the image, and
- any instrument specific characteristics.

The **PyDrizzle** design provides the flexibility to support multiple instruments simultaneously, and to combine observations from several instruments into a single output field with arbitrary specification. The virtual output image is built using the World Coordinate System (WCS) information from each input image and the distortion model. This provides all the information necessary to calculate the input parameters for **drizzle**. These parameters are then passed to **drizzle** which actually performs the image combination.

4.3.3 PyDrizzle Data Products

The output from **PyDrizzle** is a single multi-extension FITS file with the suffix ‘_drz.fits’. The first extension contains the science (SCI) image which is corrected for distortion and which is dither-combined (or mosaiced), if applicable. The drizzled SCI image is in units of *electrons per second*. All pixels have equal area on the sky and equal photometric normalization across the field of view, giving an image which is both photometrically and astrometrically accurate for both point and extended sources. The dimensions of the output image will be computed on-the-fly by **PyDrizzle** and the default output plate scale will be read from the IDCTAB. These parameters, however, may be chosen by the user during reprocessing to best suit the actual data.

The second extension of the output image contains the weight (WHT) image. It gives the relative weight of the output pixels and can be considered an effective exposure time map. The data quality map created by **CALACS** is used by **PyDrizzle** in creating the weight image.

The third extension of the **PyDrizzle** output image contains the so-called context (CTX) image which encodes information about which input image contributes to a specific output pixel. This is done using a bitmask for each output pixel, where ‘bit set’ means that the image, in the order it was combined, contributed with non-zero weight to that output pixel. The context image starts as a single 32-bit integer image but is extended as a cube with additional 32-bit deep planes as required to handle all the input images.

4.4 Reprocessing with PyDrizzle

The goal of the ACS pipeline is to provide data calibrated to a level suitable for initial evaluation and analysis for all users. Observers frequently require a detailed understanding of the calibrations applied to their data and the ability to repeat, often with improved products, the calibration process at their home institution. There are several occasions when off-line interactive processing with **PyDrizzle** is required:

- Observations which have been manually reprocessed with **CALACS** will still require a correction for geometric distortion.
- Users may wish to modify the default **PyDrizzle** parameters, specifying an alternate output image orientation, pixel “shrinking factor” `pixfrac`, or pixel scale.
- When images have been taken across multiple visits, the observations are not automatically associated. Creating custom association tables will allow data from multiple visits to be combined.
- For optimal image registration, the user may wish to fine-tune the WCS information by supplying additional shifts and/or rotations.
- During pipeline processing, **PyDrizzle** makes assumptions about which DQ flags should be considered bad. These pixels will be excluded when creating the drizzled product. Users may wish to select which flagged pixels should actually be considered good and included in the drizzled product.

The **PyDrizzle** software uses the Python language and can only be executed in the PyRAF environment. PyRAF, developed at STScI, allows users to run tasks using the standard IRAF parameter-based interface or by

using separate Python programs. The PyRAF interface and the most recent version of **PyDrizzle** can be downloaded from:

http://www.stsci.edu/resources/software_hardware/stsdas/pydrizzle.

The calibrated products from **CALACS** and the corresponding input to **PyDrizzle** are described for various observing modes in Table 3.1. For example, a single image will produce an FLT file, a RPT-OBS exposure will produce an SFL file, a CR-SPLIT exposure will produce a CRJ file, and a DITH-PATTERN will produce a series of FLT, CRJ, or SFL files at *each* dither position. **PyDrizzle** may be executed using any of the calibrated data products or the association table itself. When executed on an association table, **PyDrizzle** looks only for the existing product file and geometrically corrects that image. **PyDrizzle** will create a single drizzled image with the DRZ suffix for each calibration product. In the case of dithered observations which are part of a pattern, the individual FLT files will be combined to create a single DRZ image.

PyDrizzle was not developed with the capability of detecting cosmic-rays, so unless cosmic-ray rejection is performed in the first stage by **CALACS**, the calibrated, drizzled product will still contain cosmic-rays. A new package, **MultiDrizzle**, addresses this deficiency and is discussed in detail in Section 4.5.

4.4.1 Specifying the ‘Bits’ Parameter

The data quality flags which were set during **CALACS** processing can be used as bit masks when drizzling, and the user may specify which bit values should actually be considered ‘good’ and included in image combination. This is done via the parameter ‘bits’. Any pixel flagged otherwise will be considered ‘bad’ and will be given zero weight. If a pixel is flagged as bad but has non-zero weight in any other image, the pixel will be replaced with the value from that image during image combination. Otherwise, the pixel will be replaced with the ‘fill value’. The default ‘fill value’ used by **PyDrizzle** during pipeline processing is zero. If the ‘fill value’ is set to INDEF during reprocessing with **PyDrizzle** or **MultiDrizzle**, and there are no pixels with non-zero weight, the pixel will retain its original value.

The flags for the DQ array were presented in Chapter 3 and are summarized in Table 3.4. The bits value used during pipeline processing is 8578 which is the sum of 8192 (corrected cosmic rays) + 256 (saturated data) + 128 (bad pixel in bias file) + 2 (pixel rejected in second iteration of cr-rejection). Note that these pixels were flagged during **CALACS** processing as “bad” or “suspect”, but may have been corrected in later processing steps. For example, cosmic rays are flagged 8192, but these pixels are replaced by unaffected pixels from other images in the

CR-SPLIT association. The bits parameter indicates which “suspect” pixels to keep. When the total counts in a given pixel have exceeded the full well, for example, the DQ flag is 256. However, testing shows that counts still accumulate in a highly linear manner and that photometry of saturated stars is quite practical if using a gain that samples the full well depth.

The default bits value in the **PyDrizzle** and **MultiDrizzle** software is zero, but should be reset by the user prior to reprocessing. The value chosen for this parameter is completely up to the user and should be selected based on the specific calibration needs. For HRC processing, the user may want to add 8 (masked by aperture) to the 8578 value so that pixels which lie behind the occulting finger mask are not given zero weight. (In Figure 4.9 of Section 4.4.4 “PyDrizzle Examples”, bit 8 was not included in the bits parameter prior to drizzling. Pixels behind the occulting finger were therefore given zero weight and replaced with the default fill value=zero.) Of course, the photometry in this region will be inaccurate due to improper flat fielding, so this bit would likely only be set for aesthetic reasons.

4.4.2 Creating Custom Association Tables

Association tables for dithered, REPEAT-OBS, or CR-SPLIT observations are generated by the **CALACS** pipeline. These tables keep track of the input exposure filenames and the output product filenames. Some types of observations, however, will not have association tables. Others will have multiple tables from different visits which need to be combined into one. In the following discussion, we present the methodology for creating custom association tables, either by merging association tables or creating them from scratch. Users also have the capability to manually edit association tables to include any known image offsets. This can be done using the TTOOLS task **tedit** in IRAF, where the user adds the columns XOFFSET, YOFFSET and/or ROTATION to the association table. Alternately, an ascii file with the known image offsets may be specified and the association automatically updated.

Merging Association Tables

Observing programs which cover a large portion of the sky will generally be made up of multiple pointings. Each of these pointings may be dithered or split for cosmic-ray rejection and will possess their own association table. In order for **PyDrizzle** to produce a single mosaic product, a composite association table must be built from the separate association tables. Users can easily create this composite by merging the individual tables using the TTOOLS task **tmerge** with “option = append”.

The default product rootname will be taken from the first listed DTH-PROD in the composite association table. This rootname can be modified by the user (with **tedit**) to suit the desired naming convention.

Generally, this rootname should match the rootname of the composite association table. A detailed example of merging association tables is given in step 1 of Example 4 from Section 3.5.2.

Creating Association Tables from Scratch

Some images will not have association tables. These may be single exposures of a given target which were taken at different times and orientations. They may even be exposures taken with different instruments. Observations taken using POS-TARG offsets, which are specified manually in the proposal, will not have association tables either.

A Python tool called **buildAsn** has been developed to create association tables which may then be used by **PyDrizzle**. This task can be run under PyRAF using an IRAF parameter-based “epar” interface or directly using a Python command-line interface. Since it resides alongside the **PyDrizzle** code, no extra setup is required to use this task.

A built-in help facility provides details on additional parameters which can be used with **buildAsn** when using the Python syntax. (For those needing a step-by-step introduction to starting PyRAF, see the examples in Section 4.4.4. Building an association table will be part of the **PyDrizzle** Examples.) It can be accessed using:

```
pyraf> from pydrizzle import buildasn
pyraf> buildasn.help()
```

The following illustrates how to use **buildAsn** to create an association table from all calibrated input files with the FLT suffix found in a single directory. Within PyRAF and using the IRAF syntax, we type:

```
pyraf> buildasn mymosaic suffix='flt'
```

Alternately, the same result may be obtained using the Python syntax:

```
pyraf> buildasn.buildAsnTable('mymosaic', suffix='flt')
```

In this example, the association table will have the name ‘mymosaic_asn.fits’, the product name will be ‘mymosaic_drz.fits’, and all files with the FLT suffix found in the working directory will be included in the association. To specify a subset of images within a given directory, the user may specify the ‘suffix’ parameter to be a filelist (‘@filename’), a wild-card list (‘*8cd*flt*’), or any portion of the filename (‘crj’ or ‘f555w’).

If user determined offsets are available, **buildAsn** has the capability of incorporating them into the association table. These offsets (XOFFSET,

YOFFSET, and/or ROTATION) are specified by the file listed in the ‘shiftfile’ parameter. Using the IRAF syntax:

```
pyraf> buildasn mymosaic suffix='flt' shiftfile='shift.txt'
```

This option allows users to fine-tune the final image combination by providing corrections to the header WCS information. For more information on the format of user-defined shifts, see Section 4.4.3. For an example illustrating the steps required to apply a shiftfile to an existing association, see Example 3 in Section 4.4.4.

4.4.3 Applying User-defined Shifts

The ability of **PyDrizzle** to properly align dithered observations relies on the accuracy of the header WCS information. Unfortunately, there are times when those WCS values may be slightly inaccurate, resulting in a mis-alignment of the final drizzled product. For example, one might wish to combine observations taken in different visits. If the visits were separated by a substantial time interval, then the telescope probably used different guide stars. Due to limitations in the intrinsic accuracy of the Guide Star Catalog, this can lead to WCS coordinates that differ by up to 1-2" between visits. Also when all exposures that one wishes to combine were taken in a single visit, it might still be important to correct for relative shifts between images. Even in two-star fine-guiding mode, thermal effects on the telescope can cause slow drifts that can build up to ~50 mas over a few orbits (or in rarer occasions within a single orbit). This corresponds approximately to 1 WFC pixel or 2 HRC pixels. Such shifts between exposures can degrade image quality and corrupt cosmic-ray rejection when left uncorrected.

By deriving shifts based on the position of objects in the data, the user may refine the shifts computed from the WCS header information to create a precisely-aligned drizzled product. Many observers have developed their own methods of comparing images and computing offsets. The conventions described here should support the majority of users, making it a simple matter to incorporate shift computations into **MultiDrizzle** or **PyDrizzle**.

Delta Shifts

A *delta shift* is defined as the residual shift required to align sources, *after* applying the offsets implied by the header WCS. Delta shifts may be determined by separately drizzling images onto a common WCS frame with the same central RA/Dec. This is performed in the ‘*driz_separate*’ step of **MultiDrizzle** (see section 4.5 for more information) or by setting the **PyDrizzle** parameter ‘single=yes’. Object lists derived for each drizzled image may then be matched and fit to derive a single delta shift for each image.

Delta shifts are defined to be in the ‘input’ frame of reference when the distortion-corrected, drizzled image has the same orientation and scale as the distorted image. Shifts will be in the ‘output’ frame of reference if any rotation or scaling was applied while drizzling. When the shifts are given in the ‘output’ frame of reference, **PyDrizzle** requires the specification of the reference image to properly account for any orientation and scale changes.

Absolute Shifts

An *absolute shift*, on the other hand, is the total shift between two images and includes offsets implied by the header WCS. Absolute shifts may be determined by separately drizzling images to their own unique WCS frame. **PyDrizzle** will use the header information to optimally place the drizzled image in the frame.

To derive absolute shifts in the ‘input’ frame of reference, each distorted image is drizzled separately, and **PyDrizzle** uses the unique WCS information from each frame to choose the central RA/Dec, the image orientation and final image size. No additional rotation or scaling is applied while drizzling. (Alternately, the user may catalogue sources in each distorted image and then apply the distortion model to the X/Y positions. However, this requires tasks to transform coordinates from distorted to undistorted space. These tasks are still being developed for release in the STSDAS dither package.)

Absolute shifts in the ‘output’ frame of reference may be computed by separately drizzling each distorted image, with the same rotation and scaling, onto a unique WCS frame. Target lists in each drizzled image may then be matched to derive the absolute shifts.

Figure 4.7: Delta Shifts in the ‘input’ and ‘output’ frame of reference.

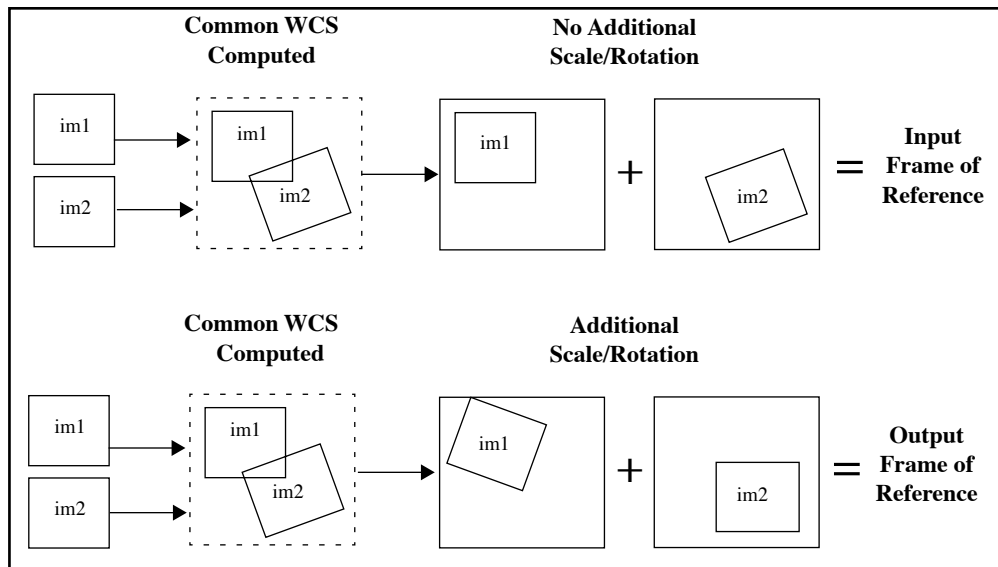
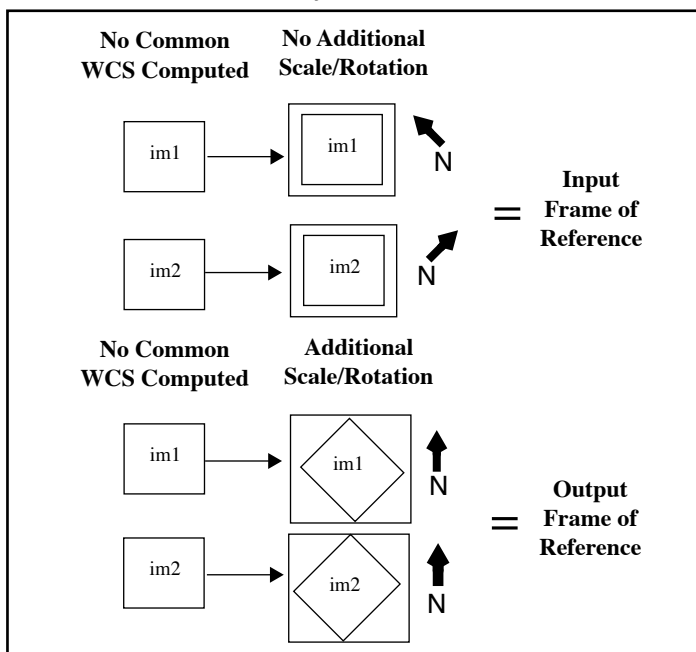


Figure 4.8: Absolute Shifts in the ‘input’ and ‘output’ frame of reference. The direction of north is indicated by the bold arrows.



Shift Units

PyDrizzle is capable of interpreting shifts in units of *pixels* or *arcseconds*. A *pixel shift* is the difference in the X/Y pixel position of a source in a given image compared to the reference image. The shifts are interpreted with respect to each image’s X/Y pixel reference position and require the specification of ‘input’ or ‘output’ frame of reference.

An *arcsecond shift* is the difference in the RA/Dec coordinates of a source in a given image compared to the reference image. Object coordinates must be derived *after* correcting for distortion. These shifts are *not* angular offsets on the sky, rather they are changes in the RA/Dec of the reference pixel and do not require any ‘cos(Dec)’ adjustment. Arcsecond shifts are a natural result of using the RA/Dec coordinates of image sources. They eliminate any ambiguity with the chosen reference frame since RA/Dec coordinates are fixed on the sky. Arcsecond shifts are interpreted as offsets in the direction of increasing RA and Dec.

Sign Conventions

The shifts are defined in terms of how the targets in the image need to move, rather than on how the telescope would need to move, to allow precise registration. **PyDrizzle** assumes that the shifts were computed as ‘image - reference’. For example, when using **geomap** to cross-correlate two source lists, the input image coordinates will be in the first two columns and the reference image coordinates will be in the last two columns.

Use of Shifts in Association Tables

The *shift file* is an ASCII file containing the user-computed shifts for a set of images and is used for updating the offsets in the association table. Observers may use either the ASN table delivered by the pipeline or create/modify their own table as described in Section 4.4.2. Both **PyDrizzle** and **MultiDrizzle** use the **buildasn** task to incorporate information from the shift file into the association table. The *shift file* uses the following format:

```
# units: pixels           Values: pixels (default), arcseconds
# frame: input           Values: input (default), output
# form: absolute         Values: absolute (default), delta
# reference: <filename> Only required when 'frame = output'
filename1  xshift1 yshift1 [rotation1]
filename2  xshift2 yshift2 [rotation2]
```

The first three lines specify the units, the frame of reference, and the form of the shifts, and only those lines with non-default values are necessary. While the '#' symbol is required, the 'Values:' comments are not. When shifts are given in the 'output' frame of reference, it is necessary to specify the name of the reference image used.

The following is an example of a shift file in units of pixels, measured in the 'input' frame of reference, with the form specified as 'delta' shifts. An x and y shift and a rotation (in degrees) is specified for each dataset. If only a simple shift is required, then the rotation column need not be specified.

```
#units: pixels
#frame: input
#form: delta
j8c0b1skqflt -2.4 -1.2 -0.002
j8c0b1smqflt -4.3 -2.3 0.001
j8c0b1sqqflt -6.9 -3.5 0.003
```

Using this shiftfile as input, the **buildasn** task will update the ASN table as indicated in Table 4.3.

Table 4.3: Association table updated with user-defined shifts.

MEMNAME	MEMTYPE	XOFF- SET	YOFF- SET	XDELTA	YDELTA	ROTATION
j8c0b1skq	EXP-DTH	0	0	-2.4	-1.2	-0.002
j8c0b1smq	EXP-DTH	0	0	-4.3	-2.3	0.001
j8c0b1sqq	EXP-DTH	0	0	-6.9	-3.5	0.003

The shift file can contain shifts for files other than those contained in the ASN table. In that case, only those entries found in the ASN table will be used to update the table. Thus, a single shift file can be generated for a whole set of observations which are represented by multiple association tables.

Association (ASN) tables can support either delta shifts or absolute shifts in units of pixels or arcseconds. Absolute shifts are stored in columns ‘XOFFSET’ and ‘YOFFSET’, while delta shifts are stored in columns ‘XDELTA’ and ‘YDELTA’. The ‘ROTATION’ column is also available if a simple shift does not produce ideal registration and indicates how much should be added to the image’s ORIENTAT to get proper alignment.

If *absolute offsets* are provided in the shift file, **PyDrizzle** will populate the OFFSET columns in the association and fill the DELTA values with zeroes. Similarly, if *delta shifts* are provided, those values will be used to populate the DELTA columns and the OFFSET columns will be zero. If shifts are given in both the OFFSET and DELTA columns, the OFFSET column will be applied and the DELTA columns will be ignored. This convention used by **buildasn** eliminates any ambiguity as to which values are applied to the data. Editing the ASN table directly will then allow the user to further update the offsets using either *delta* or *absolute* shifts.

4.4.4 PyDrizzle Examples

These examples further describe the steps required for pipeline reprocessing and serve as a continuation of the **CALACS** examples described in Section 3.5.2. We advise the reader to work through the **CALACS** examples first to develop a solid understanding of the calibrated data products and their potential limitations.

Example 1: Drizzling a Single Exposure

The following is a continuation of Example 1 from Section 3.5.2 where manual recalibration of a single raw exposure with **CALACS** produced a single image with the FLT suffix.

1. In the Unix environment, the ‘jref’ environment must be established prior to loading PyRAF. This directory indicates the location of the geometric distortion reference file (IDCTAB) specified in the image header. For more information, see “Setting up jref” in Section 3.5.1. If running IRAF and PyRAF simultaneously, ‘jref’ will need to be defined in each xterm window before loading the software. Alternatively, this path may be initialized in the user’s .setenv file. PyRAF

should then be started in its own xterm window and the **stdas**, **analysis**, and **dither** packages loaded.

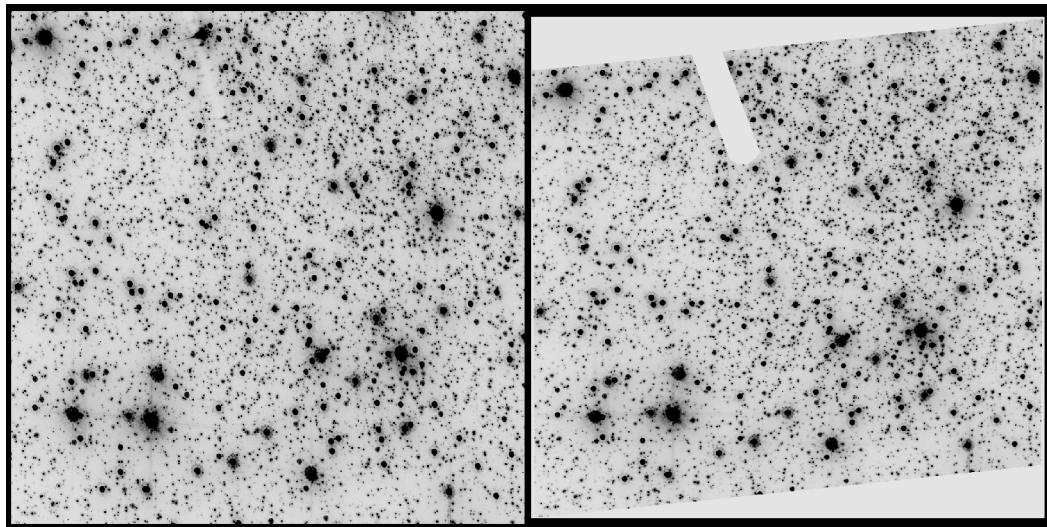
```
unix> setenv jref /mydisk/myjref/
unix> pyraf
pyraf> stdas
pyraf> analysis
pyraf> dither
```

- Next, we run **PyDrizzle** on the calibrated FLT image (j8bt07oyq_flt.fits). By default, the drizzled output product will be named j8bt07oyq_drz.fits. To specify an output file which is different than the default, the user may set the ‘outfile’ parameter to ‘myfile_drz.fits’, for example. Section 4.4.1 defined the ‘bits’ parameter; here we adopt the recommended value of 8578. The reader may also wish to add 8 to this value to see the effect of not dropping pixels behind the occulting finger. Prior to running, we recommended clearing any preset values using the ‘unlearn’ command.

```
pyraf> unlearn pydrizzle
pyraf> pydrizzle j8bt07oyq_flt.fits bits=8578
```

The distortion corrected HRC science image is shown in Figure 4.9. The image on the left is the flat fielded FLT product from **CALACS** and still contains distortion. The image on the right is the calibrated, distortion-corrected DRZ product from **PyDrizzle** using ‘bits=8578’.

Figure 4.9: The effect of applying the distortion correction to a single HRC image. The image on the left is the flat-fielded FLT product from **CALACS**. The image on the right is the distortion-corrected DRZ product from **PyDrizzle**.



Example 2: Drizzling Images taken as part of a Dither Pattern

The following is a continuation of Example 5 from Section 3.5.2 where manual recalibration of a 2-point line dither pattern produced two FLT images (j8e654c0qflt.fits, j8e654c4qflt.fits).

1. Using **tprint**, we can view the contents of the image association, including the rootnames of the individual dithered images and the default rootname of the drizzled product.

```
pyraf> tprint j8e654010_asn.fits
# MEMNAME      MEMTYPE      MEMPRSNT
j8e654c0q      EXP-DTH      yes
j8e654c4q      EXP-DTH      yes
j8e654011      PROD-DTH     yes
```

2. Next we run **PyDrizzle** on the image association. The two FLT images are combined to create a single DRZ image called 'j8e654011_drz.fits'.

```
pyraf> pydrizzle j8e654010_asn.fits bits=8578
```

Note that **PyDrizzle** is not designed to remove cosmic rays from these images. This step requires the new software **MultiDrizzle**, which is described in Section 4.5.

Example 3a: Improving the Image Registration - 'Delta Shifts' in the 'Input Frame'

For observations with large dithers or for images which were taken at nominally the same pointing but during different visits, users may wish to fine-tune the image registration, improving on the world coordinate system in the image header. In this case, the images must be separately drizzled and the position of stars cross-correlated. The derived shifts may then be used to update the image association prior to drizzling the final product. More generally, if the drizzle combined product of multiple exposures have PSFs that are not round or as sharp as expected, a likely culprit is the need for improved shifts.

The following example uses WFC observations from separate visits in program 9018: visit 61, exposure 02 (j8c061020_asn.fits) on 19April 2002 and visit C1, exposure 01 (j8c0c1010_asn.fits) on 09May2002. Each observation is CR-SPLIT with an exposure time of 45 seconds in F606W. The images are at the same pointing, use the same guide stars, and are at the same ORIENT. We describe the steps required to register the calibrated CRJ products from separate visits. However, the individual FLT files could be alternatively used if improved registration of each CRJ image is desired, although use of FLT images would result in cosmic-ray retention.

1. First, we use the CRJ images from visits 61 and c1 to create a new association table with the suffix ‘v61c1’. We advise using the **tprint** task to verify the contents of the association.

```
pyraf> import buildasn
pyraf> buildasn f606w_v61c1 suffix='crj.fits'
pyraf> tprint f606w_v61c1_asn.fits
# MEMNAME      MEMTYPE  MEMPRSNT  XOFF  YOFF  XDELTA  YDELTA  ROTATION
j8c061021     EXP-DTH   yes       0.    0.    0.      0.      0.
j8c0c1011     EXP-DTH   yes       0.    0.    0.      0.      0.
f606w_v61c1   PROD-DTH  no        0.    0.    0.      0.      0.
```

2. Next, we separately drizzle the images onto a common output WCS. This is achieved by setting the **PyDrizzle** parameter ‘single’=yes. Any differences in the aperture RA and Dec, any POS-TARG offsets, or any orientation differences will be automatically accounted for by **PyDrizzle** which uses the header WCS to transform the images. The drizzled CRJ products are appended with the suffix ‘_single_sci.fits’.

```
pyraf> pydrizzle f606w_v61c1_asn.fits bits=8578 single=yes
```

3. Using point sources in each image, we create and cross correlate coordinate lists from each visit. For example, we used **daofind** to create star lists for each DRZ product separately. Then we used **xyxy-match** to match the coordinate lists within a user specified tolerance. Finally, we use **geomap** with the matched coordinate list to compute the transformation required to map the reference coordinate system to the input coordinate system, allowing for a shift and/or rotation. We create the following “shiftfile”, a simple ascii table called ‘shift.txt’, using the derived offsets. The shifts were derived in units of *pixels* and are in the ‘input’ frame of reference since no additional rotation or scale terms were introduced while drizzling. The derived shifts using the above methods are *delta* shifts because the images were drizzled onto a common WCS.

```
#units: pixels
#frame: input
#form: delta
j8c061021_crj    0.00    0.00  0.000
j8c0c1011_crj   24.46   -56.54  0.001
```

4. Next, we delete and rebuild the previous association table, this time specifying the derived offsets via the shiftfile. To verify that the association has been correctly populated, we use the **tprint** command.

Note that the *delta* shifts appear in the DELTA columns in the association table.

```
pyraf> delete f606w_v61c1_asn.fits
pyraf> buildasn f606w_v61c1 suffix='crj.fits'
shiftfile='shift.txt'
pyraf> tprint f606w_v61c1_asn.fits
```

#	MEMNAME	MEMTYPE	MEMPRSNT	XOFF	YOFF	XDELTA	YDELTA	ROTATION
j8c061021	EXP-DTH	EXP-DTH	yes	0.	0.	0.	0.	0.
j8c0c1011	EXP-DTH	EXP-DTH	yes	0.	0.	24.46	-56.54	0.001
f606w_v61c1	PROD-DTH	PROD-DTH	no	0.	0.	0.	0.	0.

5. Finally we run **PyDrizzle** on the new image association. The two CRJ images are combined to create a single DRZ image called 'f606w_v61c1_drz.fits'.

```
pyraf> pydrizzle f606w_v61c1_asn.fits bits=8578 single=no
```

Example 3b: Improving the Image Registration - 'Absolute Shifts' in the 'Output Frame'

As a continuation of the previous example, we illustrate how to specify *absolute* shifts in the 'output' frame of reference.

1. First, the CRJ images from each visit are drizzled to separate output images. Absolute shifts are determined by separately drizzling images onto their own unique WCS frame. In this example, an additional rotation is applied to the drizzled images so that north is oriented up.

```
pyraf> pydrizzle j8c061021_crj.fits output=j8c06102X_drz.fits
bits=8578 rotate+ orient=0
pyraf> pydrizzle j8c0c1011_crj.fits output=j8c0c101X_drz.fits
bits=8578 rotate+ orient=0
```

2. As in the previous example, we create and cross correlate coordinates lists from each visit using point sources in each image. We create the following "shiftfile", a simple ascii table called 'shift1.txt', using the

derived offsets. Note that because we are using absolute shifts, we must specify the name of the reference image.

```
#units: pixels
#frame: output
#form: absolute
#reference: j8c06102X_drz.fits
j8c061021_crj    0.00  0.00  0.000
j8c0c1011_crj  -41.64  45.51  0.001
```

3. Next, we build a new image association using the two CRJ files and the derived shift file.

```
pyraf> import buildasn
pyraf> buildasn f606w_v61c1rot suffix='crj.fits'
shiftfile='shift1.txt'
```

4. To verify that the association has been correctly populated, we use the **tprint** command. Notice that the *absolute* shifts have been correctly added to the OFFSET columns, not the DELTA columns.

```
pyraf> tprint f606w_v61c1rot_asn.fits
# MEMNAME      MEMTYPE  MEMPRSNT  XOFF  YOFF  XDELTA  YDELTA  ROTATION
j8c061021      EXP-DTH   yes       0.    0.    0.    0.    0.
j8c0c1011      EXP-DTH   yes      -41.64  45.51  0.    0.    0.001
f606w_v61c1rot PROD-DTH  no        0.    0.    0.    0.    0.
```

5. Finally we run **PyDrizzle** on the new image association, specifying a rotation such that north is up. The two CRJ images are combined using the updated shift information to create a single DRZ image called 'f606w_v61c1rot_drz.fits'.

```
pyraf> pydrizzle f606w_v61c1rot_asn.fits bits=8578 rotate+
orient=0
```

4.5 MultiDrizzle

MultiDrizzle, developed in its original form by A. Koekemoer (Koekemoer et al. 2002, HST Calibration Workshop 2002, p.381), is a general purpose tool for drizzle combination of images. It provides a single-step interface to the complex suite of tasks in the STSDAS **dither** package. These tasks include, in order: initial image registration, creation of a cleaned median image, transformation back to the input image plane, creation of cosmic ray masks, and final drizzling. (For a general introduction, refer to the [HST Dither Handbook](#).) The goal of **MultiDrizzle** is to provide a high-level task which has an extensive suite of user-adjustable parameters which, if left at their default values, will allow the task to perform all steps in a single operation with no user intervention. At the same time, the parameters allow the user a large amount of flexibility in controlling the relevant aspects of these steps, in case the default parameters are not sufficient for specific scientific applications.

The current version of **MultiDrizzle** uses the image header world coordinate system (WCS) to deduce the image-to-image offsets. However, user-supplied offsets may also be applied in a flexible manner. In the future, tasks to ease the determination of such shifts will be made available. **MultiDrizzle** processing with default parameters may eventually be part of pipeline processing, if a detailed assessment shows that this is feasible.

MultiDrizzle and **PyDrizzle** are both Python scripts which can only be run within the PyRAF environment. **MultiDrizzle** calls **PyDrizzle** to manage the images, to compute the **drizzle** parameters, and to control the operation of both the **drizzle** and **blot** tasks. While the latest release of STSDAS includes **PyDrizzle**, **MultiDrizzle** must be obtained separately (see below). It is recommended that the complete current STSDAS package is installed to avoid potential missing components or mismatched versions. STSDAS and PyRAF are available from:

http://www.stsci.edu/resources/software_hardware.

Because many of these components are evolving rapidly, subsequent updates will be available from the same site.

While the software has been well tested and is fairly stable, it is still periodically being improved. We therefore refer readers to the **MultiDrizzle** web site for the most up-to-date software:

<http://stsdas.stsci.edu/pydrizzle/multidrizzle>

and to the ACS Drizzling web site for additional documentation:

<http://www.stsci.edu/hst/acs/analysis/drizzle>.

The following sections describe in detail the tasks performed by **MultiDrizzle** and give an example of its use. A help document describing the numerous **MultiDrizzle** parameters can be accessed by typing 'help

multidrizzle' from within PyRAF. HST Dither Handbook is currently under revision to include documentation for **MultiDrizzle** and several detailed ACS drizzling examples. Version 3.0 is expected to be released by mid-2004.

4.5.1 Reprocessing with MultiDrizzle

In Section 4.4, we described several occasions when off-line interactive processing with **PyDrizzle** is required, for example, when images are taken across multiple visits, when fine-tuned image registration is required, or when the user wishes to specify which DQ flags should be considered *good*. **MultiDrizzle** will address these same issues, while at the same time providing additional capabilities such as cosmic ray rejection and sky subtraction. Because it allows access to an extensive suite of user-adjustable parameters, **MultiDrizzle** gives users more flexibility in selecting the parameters that best suit their observations. This is all made possible within a single, easy-to-use interface.

MultiDrizzle creates many large files and does a considerable amount of processing. It is hence recommended that a modern fast machine with several GBytes of free space be used if possible. In the example to follow, each of the input FLT files is ~168MB and the final combined output product is 310MB. Many large intermediate files are also created but may be cleaned up if required.

The only calibration reference file required by **MultiDrizzle** is the distortion table (IDCTAB) discussed earlier in this chapter. This table describes the ACS distortion in terms of polynomial coefficients. The appropriate reference file is recorded in the image header keyword IDCTAB. This file should be obtained from the STScI archive (via Starview) and placed in some local directory. Before starting **MultiDrizzle** the environment variable "jref" must be set appropriately to point to this directory (see Example 1 in Section 4.4.4 for more information). Once the FLT files have been collected and the 'jref' environment established, PyRAF should be started and the **stsdas**, **analysis**, and **dither** packages loaded.



The IDCTAB reference file in the image header must NOT be modified prior to running MultiDrizzle or PyDrizzle. To make use of an updated distortion solution, users are advised to re-retrieve their data from the archive such that the CD-matrix keywords in the image header are correctly populated.

To avoid potential problems, it is recommended that the parameters for both **MultiDrizzle** and **PyDrizzle** be reset to the default values:

```
pyraf> unlearn pydrizzle
pyraf> unlearn multidrizzle
```

4.5.2 Tasks Performed by MultiDrizzle

MultiDrizzle starts with the flat fielded, but still distorted, FLT files created by **CALACS** in the pipeline. While these are the only mandatory input, there are many additional optional inputs including lists of improved registration information. In default mode **MultiDrizzle** performs the following steps in order, although the user has full control via the parameter file, of which steps are to be performed or omitted. Each step is described in detail in the example which follows in Section 4.6.

1. **StaticMask**: Create a mask of bad pixels, using information from the images themselves and associated data quality files,
2. **SkySub**: Carry out sky subtraction on each individual input image,
3. **Driz_Separate**: Calculate relative image shifts based on header information in the world coordinate system keywords (optionally supplemented by user-supplied shift information) and drizzle the input images onto a series of separate output images,
4. **Median**: Combine these images to create a clean median image,
5. **Blot**: Transform, or blot, the median image back to the frame of the input images,
6. **Driz_cr**: Create a cosmic ray mask for each input image based on comparison between the blotted median and its derivative and the original input image,
7. **Driz_Combine**: Perform the final **drizzle** combination step, applying the cosmic ray masks.



A warning to users: MultiDrizzle modifies the original calibrated images, for example, when updating the image DQ array with the static pixel mask and when performing the sky subtraction. When reprocessing, we advise retaining a copy of the original files in a separate directory.

4.5.3 Specifying Shifts Between Images

By default, **MultiDrizzle** uses the world coordinate system (WCS) information in the image headers to align the images. While dithers applied to a target within a single visit of HST are usually accurately reflected by the WCS, this is not the case for multiple visits which normally require guide star re-acquisitions and may utilize different guide stars. Even for nominally back-to-back exposures that are part of a CR-SPLIT association, offsets large enough to degrade final combinations do sometimes occur. As a result, it is essential to accurately determine image-to-image shifts, and possibly rotations, before running **MultiDrizzle**.

These (delta) shifts may be determined by separately drizzling each image onto a common WCS frame. This is performed by the ‘driz_separate’ task in step 3. Object lists derived for each drizzled image may then be matched and fit to derive a single shift for each image. More details on this topic and a description of the shift file formats are given in Section 4.4.3.

Once the shifts have been determined, **MultiDrizzle** must be rerun from the beginning with the shift file specified in the ‘shiftfile’ parameter. **MultiDrizzle** will update the association table to reflect these ‘delta’ offsets and will now use both the header WCS information plus the shift file to appropriately align images.

4.6 MultiDrizzle Example



The MultiDrizzle software is under continuing development. The following example was written in December 2003 should be used as a guide to facilitate reprocessing. While the algorithms and the overall processing steps will remain stable, specific details of the software may change.

This example describes the combination of six ACS/WFC images of the “Tadpole Galaxy” UGC 10214 which formed part of the ACS Early Release Observations (EROs) from program 8992. These images were acquired in two visits, with a significant shift between visits, as noted by the target RA and Dec, and with small dithers within each visit. The images were taken with the F606W filter and have the exposure times listed in Table 4.4. Prior to attempting this example, we assume that users have worked through the **PyDrizzle** examples in Section 4.4.4 and are familiar with using association tables and shift files.

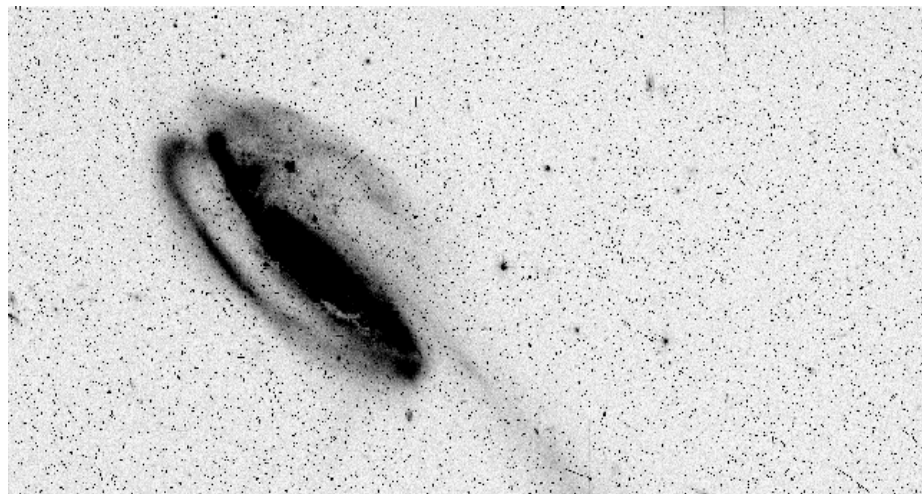
The ERO images are available from the HST archive for anyone wishing to repeat this example. A new capability has been developed in the StarView archive interface, allowing the user to request FLT files only. The FLT files are the output from **CALACS** and have been corrected for bias, dark current and flat fielding. They are not corrected for distortion and still contain numerous cosmic rays. They form the input for **MultiDrizzle**.

Figure 4.10 shows chip 1 for one of these FLT files, `j8cw54ovq_flt[sci,2]`, which still contains numerous cosmic ray events, hot pixels, and other artifacts. The rectangular, rather than ‘rhombus’ shaped image is a clear indicator that the geometric distortion has not yet been corrected.

Table 4.4: ERO datasets used in the accompanying **MultiDrizzle** example. The two sets of images were taken at different epochs and have a significant shift between epochs. Within a given epoch, small POS-TARG offsets are specified in accordance with the ACS-WFC-DITHER-LINE pattern which spans the WFC interchip gap.

Dataset	Date-Obs	RA (deg), Dec (deg)	POS-TARG1	POS-TARG2	Exposure Time (sec)
<code>j8cw04abq_flt</code>	01 Apr 2002	241.553625,+55.431094	0.000	0.000	150.0
<code>j8cw04c2q_flt</code>	01 Apr 2002		0.248	3.001	580.0
<code>j8cw04c7q_flt</code>	01 Apr 2002		0.248	3.001	840.0
<code>j8cw54oiq_flt</code>	09 Apr 2002	241.537833,+55.427594	0.000	0.000	150.0
<code>j8cw54orq_flt</code>	09 Apr 2002		0.248	3.001	580.0
<code>j8cw54ovq_flt</code>	09 Apr 2002		0.248	3.001	840.0

Figure 4.10: A single, calibrated FLT file, `j8cw54orq_flt.fits[sci,2]`, which still contains numerous cosmic rays, hot pixels, and other artifacts.



4.6.1 Detailed Processing Steps & Parameters

First, the appropriate calibrated FLT files and the geometric distortion reference table (IDCTAB) should be placed on the user's local disk. Next the 'jref' directory should be defined, PyRAF started, and the **dither** package loaded. To run **MultiDrizzle**, the parameters may be edited in the standard way using the "epar" facility.

The **MultiDrizzle** software has an extensive set of parameters, but the default values should allow the task to process nearly any set of images for an initial review. The parameters are separated according to the processing step they control, making it easier to interpret them.

In this example, we describe the parameters for each step in succession, though in practice, the user would set all relevant parameters at once. While the majority of relevant parameters are discussed here, a help document describing all parameters and tasks can be accessed by typing 'help multidrizzle' from within PyRAF.

Initial Setup

Parameters	Default	Description
output		Rootname for output drizzled products
suffix	flt	Suffix of input files in current directory
filelist		List of input files
refimage		Reference image with desired output WCS
runfile	multi.run	File for logging the script commands
restart		Parameter allowing processing to resume at a given step
coeffs	header	Use header-based distortion coefficients? (header, none)
context	no	Create context image during final drizzle?
clean	no	Remove temporary files?
section		Extension or group to be drizzled
bits	0	Integer mask bit values considered good
ra		right ascension output frame center
dec		declination output frame center
build	yes	Create multi-extension output file?
shiftfile		Shift file for improving WCS registration

The only required parameter is the rootname for the 'output' drizzled product. By default, **MultiDrizzle** will look for all files in the working directory with the FLT extension. The user may modify the 'suffix' parameter to include some other extension or may specify a subset of

images via the ‘filelist’ parameter. Using the images specified, **MultiDrizzle** calls **PyDrizzle** which uses the ‘buildAsn’ task (see Section 4.4.2) to create an association table named ‘output_asn.fits’. This association will be used to define the data set. The header WCS information from the entire set is used to define a common WCS output frame, and **MultiDrizzle** sets the **drizzle** parameters appropriately. If user defined shifts are available, these may be specified in the ‘shiftfile’ parameter, and the association table will be updated accordingly. However, a shift file is not usually available until after step 3, separately drizzling the images onto a common WCS, has been performed and objects matched.

A reference image which has the desired output WCS may be specified, and the input images will be drizzled to match the WCS of this image. Alternately, the central RA and Dec (ra, dec) of the reference pixel and the dimensions of the output frame (outnx, outny) may be specified, if desired, though reasonable values will be automatically determined from the images’ WCS if these parameters are left blank. While the central RA and Dec are specified in the initial setup parameters listed above, the output image dimensions are specified in both the ‘driz_separate’ and ‘driz_combine’ parameters in steps 3 and 7, respectively.

The ‘bits’ parameter is defined as the integer sum of all bit values from the input images’ DQ array that should be considered ‘good’ when building the weighting mask. Because **MultiDrizzle** was designed for use with multiple instruments, the default value is set to zero. For ACS data, the recommended default value is 8578. (For more information on selecting the appropriate bits for your data, refer to Section 4.4.1.) Information from the DQ array for each chip is used in combination with the ‘bits’ parameter to create temporary mask files called ‘*_inmask?.fits’. Pixels which were flagged in the DQ array *and* which were not specified as good via the bits parameter are assigned the value 0. All other pixels are set to 1 in the mask.

The distortion reference file is read from the image header via the IDCTAB keyword which specifies the name and location of the appropriate file. The distortion coefficients for each chip are written to temporary ascii files named ‘*_coeffs?.dat’. If the user wishes to retain these files, the parameter ‘clean’ should be set to ‘no’.

In general, the default **MultiDrizzle** parameters will work well for most data. When setting up the Tadpole images, for example, we have specified *only* the following non-default parameters: output=‘example’, ‘bits=8578’, ‘combine_nhigh=2’, and ‘driz_cr_snr=4.0 3.0’. The choice of these last two parameters is explained in the sections that follow. In default mode, **MultiDrizzle** performs each of its 7 steps in order. In this example, however, we perform some of the steps and examine the intermediate products before final drizzle combination is performed. Approximately 5GB of free disk space is required for this example when intermediate products are not removed. An outline of the entire process is described below:

1. Run only steps 1 through 3 of Section 4.5.2 to create sky-subtracted, separately drizzled images which are based on a common WCS.

```
pyraf> multidrizzle output='example' bits=8578 static+ skysub+
driz_separate+ median- blot- driz_cr- driz_comb-
```

2. Measure the positions of stars in the separately drizzled images and derive a shift file which defines the residual offsets.
3. Rerun step 3 using the derived 'shiftfile' to create new separately drizzled images. Also turn on step 4 to create a well-aligned median image.

```
pyraf> multidrizzle output='example' bits=8578 shiftfile='shifts'
static- skysub- driz_separate+ median+ combine_nhigh=2 blot-
driz_cr- driz_comb-
```

4. Examine the median image to ensure that the PSF is 'round' and 'narrow' and that cosmic-rays and other artifacts are appropriately rejected.
5. Run steps 5 through 7 to transform the median image back to the reference frame of each of the original input images and to derive cosmic ray masks. Using these new masks, perform the final drizzle combination.

```
pyraf> multidrizzle output='example' bits=8578 shiftfile='shifts'
static- skysub- driz_sep- median- blot+ driz_cr+ driz_cr_snr='4.0
3.0' driz_comb+
```

Once the optimal set of parameters and the optional shift file is derived, as described in the outline above, **MultiDrizzle** may be executed in a single pass by turning all steps on and by specifying any desired non-default parameters:

```
pyraf> multidrizzle output='example' bits=8578 shiftfile='shifts'
static+ skysub+ driz_sep+ median+ combine_nhigh=2 blot+ driz_cr+
driz_cr_snr='4.0 3.0' driz_comb+
```

While **MultiDrizzle** may be executed from the command line, as shown in the above examples, it may also be run from the 'epar' facility which allows the user to see all parameters at once and to turn particular steps on and off. We recommend that beginners use the 'epar facility' to become familiar with all steps and parameters before running **MultiDrizzle** from the command line.

1. Creating the Static Mask

Parameters	Default	Description
static	yes	Create static bad-pixel mask from data?
staticfile		Name of (optional) input static bad-pixel mask
static_goodval	1.0	Value of good pixels in the input static mask
Output Files	Modified DQ arrays in the original input files	

When ‘static=yes’, this step goes through each of the input images, calculates the rms value for each chip, and identifies pixels that are below the median value by more than 5 times the rms. It is aimed at identifying pixels that may have high values in the dark frame, which is subtracted during calibration, but may not necessarily have high values in the images, and thus subtraction gives them strongly negative values. Such pixels are not always flagged in the DQ file, and this step allows them to be identified. Sometimes such pixels fall on bright objects so instead of being negative, they would be positive but lower than surrounding pixels. If the images are dithered, then they should land on blank sky at least some of the time, in which case they will appear negative and will be flagged.

For the Tadpole Example, we have left the ‘StaticMask’ parameters to their default values. After this step is performed, the image DQ arrays are updated with the static mask, and new flagged pixels are set with bit 64. The ‘*_inmask?.fits’ files are subsequently updated during further processing.

2. Performing Sky Subtraction

Parameters	Default	Description
skysub	yes	Perform sky subtraction?
skytype	“single”	Type of sky subtraction (single, quadrants)
skysum	“SKYSUM”	Header keyword containing sky value (SKYSUM = ACS, BACKGRND = WFPC2)
skywidth	50.0	Interval width for sky statistics
skystat	“median”	Sky correction statistics parameter (median, mode, mean)
skylo	-50.0	Lower limit of usable data for sky (always in DN)
skyhi	200.0	Upper limit of usable data for sky (always in DN)

Output Files Modified science array in the original input files

When ‘skysub=yes’, this task will subtract the sky from each chip (‘skytype=single’) or from each of the four individual amplifiers on the WFC chips separately (‘skytype=quadrants’). The other parameters correspond directly to those in the **sky** task in the **dither** package, and are passed to it exactly as they are specified here.

The main parameter that users must set is the histogram width, though the default value is good for most cases. The width determines the region of the pixel histogram used to determine the image statistics. It should be set to include most pixels in the sky (so substantially more than the FWHM of the sky distribution) but not so large as to include a substantial amount of power from objects or cosmic rays.

The ‘SkySub’ task will update the header keyword defined by the parameter ‘skysum’ with the derived sky value for each chip and will subtract the sky from the original FLT images. Sky subtraction is recommended for effective cosmic-ray flagging and removal, but only if sufficient blank sky is available to perform an accurate determination.

Great care must be taken when choosing to implement sky subtraction, because the original calibrated images will be modified, and because if sufficiently blank sky is not available, sky subtraction will produce erroneous results. In the case of the Tadpole images, adequate blank sky is available for each WFC chip to allow an accurate sky determination when ‘skytype’ is set to ‘single’. Choosing ‘skytype=quadrants’, on the other hand, results in an over-subtraction of the ‘A’ amplifier (upper-left quadrant) since the galaxy light dominates the background in this region of the detector.

3. Drizzling to Separate Output Images

Parameters	Default	Description
driz_separate	yes	Drizzle to separate output images?
driz_sep_outnx		Size of x-axis for separate output frame
driz_sep_outny		Size of y-axis for separate output frame
driz_sep_kernel	“turbo”	Shape of kernel function (square, point, gaussian, turbo, tophat, lanczos3)
driz_sep_scale	1.0	Linear size of output pixels (relative to input)
driz_sep_pixfrac	1.0	Linear size of drop in input pixels
driz_sep_rot	0.	Rotation of input image to be applied (degrees anti-clockwise)
driz_sep_fillval	INDEF	Value assigned to undefined output pixels
Output Files	Images= ‘*_single_sci.fits’, ‘*_single_wht.fits’	

When ‘driz_separate=yes’, the input images are corrected for geometric distortion and drizzled onto separate output frames which have a common WCS. Any shifts, rotations, or scale changes are calculated from the image headers by **PyDrizzle**. The output image dimensions are calculated on-the-fly and the pixel scale is taken from the column ‘scale’ from the IDCTAB, where the default values are 0.05 *arcsec/pix* for the WFC and 0.025 *arcsec/pix* for the HRC. The drizzled images are in units of *electrons/sec*.

By default, the ‘driz_separate’ task uses the ‘turbo’ drizzle kernel and **drizzle** parameters ‘pixfrac=1’ and ‘scale=1’. For more information on setting these parameters, refer to the [HST Dither Handbook](#). These parameters can be changed; for example, masks can be substantially improved by specifying a smaller value of scale (e.g., 0.5 or 0.66), with the primary trade-off being much larger images (their size increases as the inverse square of the value of ‘scale’) and increased computation time.

In the Tadpole example, we have left the ‘Driz_Separate’ parameters to their default values. While **PyDrizzle** produces a final drizzled image containing 3 extensions (the science, weight, and context images), the ‘driz_separate’ products are separate science and weight images named ‘*_single_sci.fits’ and ‘*_single_wht.fits’. No context image is created.

One of the singly drizzled FLT images, ‘j8cw54ovq_ft_single_sci.fits’, is shown in Figure 4.11. This image still contains numerous cosmic ray events, hot pixels, and other artifacts. The ‘rhombus’ shape is a result of correcting the geometric distortion. The corresponding weight image, ‘j8cw54ovq_ft_single_wht.fits’, is shown in Figure 4.12, where white indicates pixels with zero weight. Due to the effects of distortion and

varying pixel area in the FLT images, the weight image changes gradually across the detector. Because the association table was used to define a common WCS for all images, the drizzled image is ‘padded’ with zeros outside the boundary of the original array. The weight image is set to zero in these regions, allowing these pixels to be rejected during median combination.

The separately drizzled science images may be used to improve the image registration prior to final drizzle combination. In this example, the images form two groups of three. While the WCS information for images within a single group (visit) are adequate to align them, there is a small residual offset between visits. Shifts which are determined from separately drizzling images onto a common WCS are by definition ‘delta’ shifts (see Section 4.4.3 for details), and will be applied in addition to any offsets from the WCS when a ‘shiftfile’ is provided. Because no additional scale or rotation was applied to the singly drizzled images, the shifts are in the ‘input’ frame of reference.

In this example, unsaturated stars in the short exposures (j8cw04abq and j8cw54oiq) were used to derive the offsets between the two groups of images. Shifts were measured to the nearest 0.1 pixel and are listed below in the form of a shift file. Refined shifts within a given visit may also be determined, but these are typically less than a tenth of a pixel, and accuracy at this level is not as critical for extended sources as it may be for point sources.

```
#units: pixels
#frame: input
#form: delta
j8cw04abq_flt 0.0 0.0
j8cw04c2q_flt 0.0 0.0
j8cw04c7q_flt 0.0 0.0
j8cw54oiq_flt -5.6 -0.3
j8cw54orq_flt -5.6 -0.3
j8cw54ovq_flt -5.6 -0.3
```

When the ‘driz_separate’ step is run for the second time, but with a ‘shiftfile’ specified, the association table for the data set will be automatically updated. To confirm that the new separately drizzled images are appropriately registered, the position of stars should again be examined. It is also useful to create a median image and examine the width and shape of the PSF over the entire FOV to look for any effects of mis-registration. Median combination is performed in Step 4 of **MultiDrizzle**.

Figure 4.11: The singly drizzled FLT image 'j8cw54ovq_ft_single_sci.fits' from the **MultiDrizzle** example.

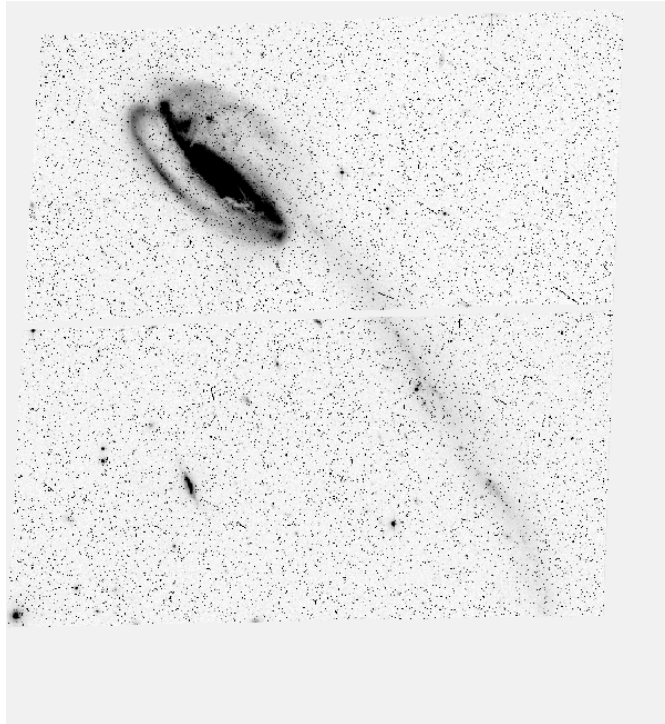
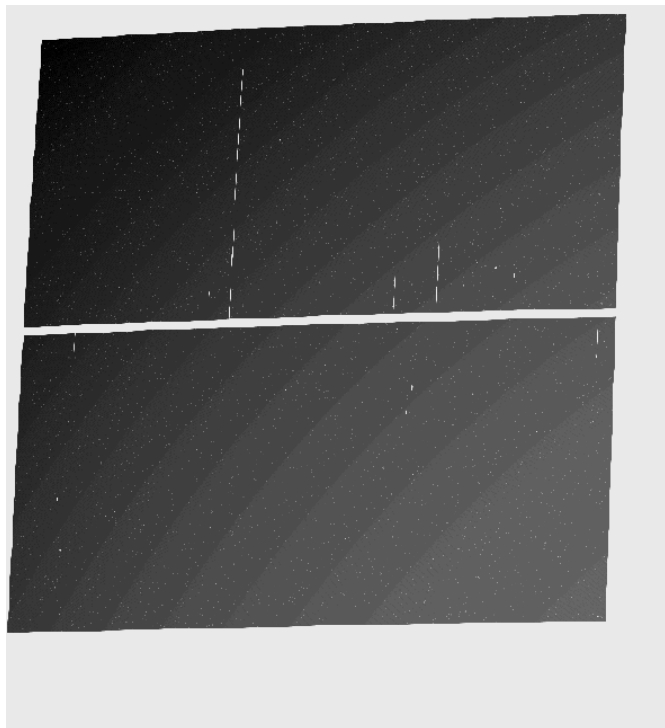


Figure 4.12: The weight image 'j8cw54ovq_ft_single_wht.fits' corresponding to the singly drizzled image in Figure 4.11 where white indicates zero weight.



4. Creating the Median Image

Parameters	Default	Description
median	yes	Create median image?
median_newmasks	yes	Create new masks when doing the median?
median_preclean	no	Remove intermediate files prior to doing median?
combine_type	“median”	Type of combine operation (average, median, sum, minmed)
combine_reject	“minmax”	Type of rejection (none, minmax, ccclip, crreject, sigclip, avsigclip, pclip)
combine_nsigma	6 3	Significance for accepting min instead of median
combine_nlow	0	minmax: Number of low pixels to reject
combine_nhigh	1	minmax: Number of high pixels to reject
combine_lthresh	INDEF	Lower threshold for clipping input pixels
combine_hthresh	INDEF	Upper threshold for clipping input pixels
combine_grow	1.0	Radius (pixels) for neighbor rejection
Output Files	Images= ‘output_med.fits’, ‘*_single_wht_maskhead.pl’	

When ‘median=yes’, this step creates a median image from the separate drizzled input images, allowing a variety of combination and rejection schemes. If ‘combine_type’ is set to ‘median’ or ‘average’, then the routine calls the IRAF task **imcombine**, passing to it the values of ‘combine_reject’ (the rejection algorithm chosen, usually expected to be ‘minmax’), ‘combine_nlow’ and ‘combine_nhigh’ (the number of low and high pixels to reject), and ‘combine_grow’ (the amount by which flagged pixels can grow). All **imcombine** parameters other than those specified above are reset to their default values. While the ‘median’ is recommended, a slightly more sophisticated algorithm than those available in **imcombine** will be used when ‘combine_type=minmed’. This algorithm is described at the end of this section.

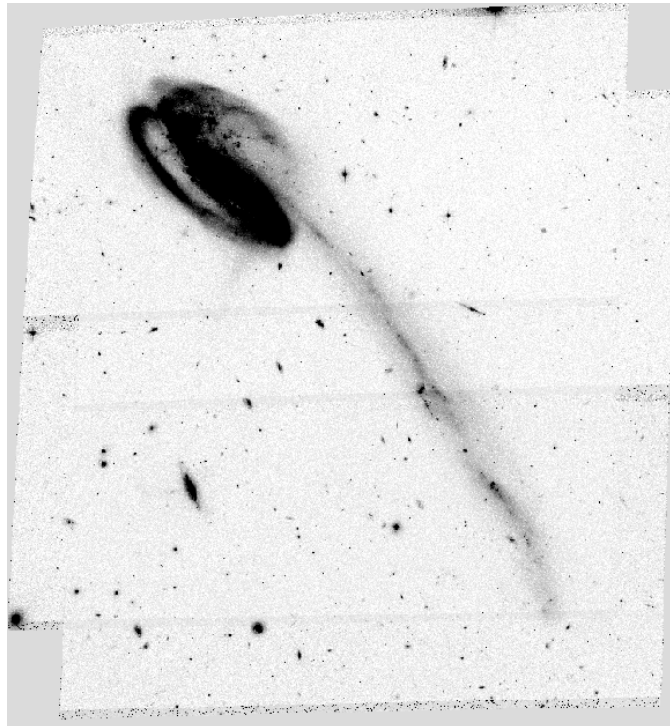
If ‘median_newmasks=yes’, then the singly drizzled weight maps (‘*_single_wht.fits’) are used to create pixel masks for each image (with values 0 and 1) which are named ‘*_single_wht_maskhead.pl’. The IRAF task **mask_head** prepares the singly drizzled images for use with **imcombine** by populating the header bad pixel mask keyword ‘BPM’ for each image. These masks will be used by **imcombine** when combining images, where the assumed mask parameters are ‘masktype=goodvalue’ and ‘maskvalue=1’, indicating that pixels assigned a value of 1 are considered good.

If 'median_newmasks=no', this task will use whatever masks are specified by the user (and which are created offline) in the 'BPM' header keyword of each image. In general, however, it is recommended that the pixel masks which are generated by default are used instead.

Selecting the best parameters for the median step can be an iterative process and should always involve examination of the clean, combined product to verify that the majority of cosmic-rays and other artifacts are successfully removed. The rejection algorithm which is ultimately chosen depends largely on the number of datasets being combined and the amount of overlap between dithered images.

In this example, we have chosen the default parameters 'combine_type=median', 'combine_reject=minmax', and 'combine_nlow=0'. Instead of the default value, we have set 'combine_nhigh=2' so that hot pixels are rejected in the outer portions of the image, which have only 3 input images contributing to the median calculation, and for which 2 of the 3 images are at the same dither position. Thus, the **imcombine** parameter 'nkeep' (the minimum number of pixels retained) is 1 in the outer portion of the median image. In the central regions of the image, six images have contributed to the median, but two high images are rejected, so 'nkeep' is 4. The six separately drizzled images are combined using the bad pixel masks and the rejection parameters specified above to create a single clean median image named 'example_med.fits'. This median image is shown in Figure 4.13.

Figure 4.13: The cleaned median image created using the 6 separately drizzled Tadpole images and their bad pixel masks.



When ‘combine_type=minmed’, a slightly more sophisticated algorithm will be used to combine images. This algorithm requires significantly more computation time and disk space (>1GB) compared to the other available algorithms. The basic concept is that each pixel in the output combined image will be either the median or the minimum of the input pixel values, depending on whether the median is above the minimum by more than n times sigma. An estimate of the “true” counts is obtained from the median image (after rejecting the highest-valued pixel), while the minimum is actually the minimum unmasked (“good”) pixel. This algorithm is designed to perform optimally in the case of combining only a few images (3 or 4), where triple-incidence cosmic rays often pose a serious problem for more simplified median combination strategies. The algorithm performs the following steps:

1. Create median image, rejecting the highest pixel and applying masks.
2. Use this median to estimate the true counts, and thus derive an rms.
3. If the median is above the lowest pixel value by less than the first value in ‘combine_nsigma’, then use the median value, otherwise use the lowest value.
4. If ‘combine_grow’ > 0, repeat the above 3 steps for all pixels around those that have already been chosen as the minimum, this time using a lower significance threshold specified as the second value in ‘combine_nsigma’.

The last step is very successful at flagging the lower signal-to-noise “halos” around bright cosmic rays which were flagged in the first pass.

5. Blotting Back the Median Image

Parameters	Default	Description
blot	yes	Blot the median back to the input frame?
Output Files	Images= ‘*_sci?_blt.fits’	

When ‘blot=yes’, this task takes the median image and uses the **dither** package **blot** task to apply the geometric distortion and to transform (‘reverse drizzle’) it back to the reference frame of each of the original individual input images. This involves reversing the shifts and reapplying the geometric distortion that had been removed in step 3. In addition, the median image is resampled to the pixel scale of the original images and is trimmed to match the dimensions of each input image. This step is done in preparation for subsequent cosmic-ray rejection in step 6. The blotted frames are named ‘*_sci?_blt.fits’.

If desired, the user may wish to display the input images and blink them with their ‘blotted’ counterparts. The ‘blotted’ images should align perfectly with their respective input images and should be reasonably similar in appearance, except for the fact that they should be cleaned of cosmic rays and other defects.

6. Creating Cosmic Ray Masks

Parameters	Default	Description
driz_cr	yes	Perform CR rejection with deriv and driz_cr?
driz_cr_snr	“3.0 2.5”	Driz_cr.SNR parameter
driz_cr_scale	“1.2 0.7”	Driz_cr.scale parameter
Output Files	Images= ‘*_sci?_blt_deriv.fits’, ‘*sci?.fits’, ‘*_sci?_crderiv.pl’, ‘*_sci?_cor.fits’, ‘*_sci?_mask.pl’	

When ‘driz_cr=yes’, this step uses the original input images, the blotted median images, and the derivative of the blotted images (which it creates using the **deriv** task) to create a cosmic ray mask for each input image (using the **driz_cr** task).

First, the **deriv** task uses the blotted median images (‘*_sci?_blt.fits’) from step 5 to calculate the absolute value of the difference between each pixel and its four surrounding neighbors. For each pixel, the largest of these four values is saved in an output image, ‘*_sci?_blt_deriv.fits’, which represents an effective gradient or spatial derivative.

These derivative images are used by the task **driz_cr** when comparing the original and blotted images. First, the original FLT images for each chip are copied to files named ‘*_sci?.fits’ which are the required input for the **driz_cr** task. These images are compared with the corresponding blotted median image ‘*_sci?_blt.fits’ and its absolute derivative ‘*_sci?_blt_deriv.fits’ to create a mask of cosmic rays (and other blemishes, like satellite trails). Where the difference is larger than can be explained by noise statistics, or the flattening effect of taking the median, or perhaps an error in the shift (the latter two effects are estimated using the image derivative), the suspect pixel is masked. Cosmic rays are flagged using the following rule:

$$|data_image - blotted_image| > scale * deriv_image + SNR * noise$$

where ‘scale’ is the user supplied **driz_cr** parameter listed above and is defined as the multiplication factor applied to the derivative before determining if the difference between the data image and the blotted image is sufficiently great to require masking. ‘Noise’ is calculated using a combination of the detector read noise and the poisson noise of the blotted median image plus the sky background.

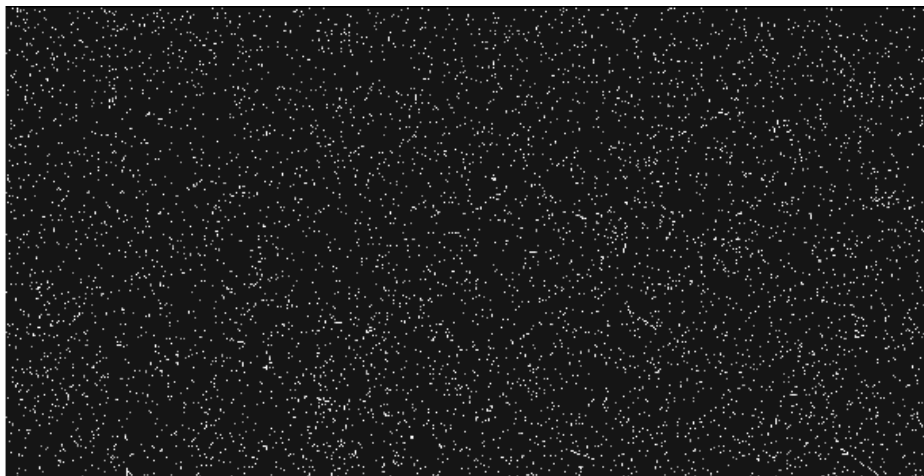
The user must specify a cut-off signal-to-noise (SNR) value for determining whether a pixel should be masked. Actually, two cut-off signal-to-noise ratios are needed, one for detecting the primary cosmic ray, and a second for masking lower-level bad pixels adjacent to those found in

the first pass. After the first pass through the image, the procedure is thus repeated on pixels that are adjacent to previously masked pixels using a lower SNR threshold, since cosmic rays often extend across several pixels.

The final output is a cosmic-ray mask file named `*_sci?_crderiv.pl`. One of the resulting masks for chip 1 is shown in Figure 4.14 and should be blinked with the original image `j8cw54orq_ft[sci,2]` from Figure 4.10 (or the equivalent `j8cw54orq_ft_sci2.fits` file) to visually ascertain that all cosmic rays were flagged. If it appears that the central pixels of some stars were unnecessarily masked, the `'driz_cr_scale'` parameter should be increased. If not enough cosmic rays were masked out, this parameter should be decreased. In this example, the default `'driz_cr_snr'` values "3.0 2.5" were too stringent and resulted in flagging the centers of stars and the core of the Tadpole galaxy. Instead, we have increased the default SNR values to "4.0 3.0" to create ideal cosmic ray masks for this data set.

The **driz_cr** task also creates a `*_sci?_cor.fits` image, where flagged pixels are replaced with pixels from the blotted median image. The cosmic ray mask files are then multiplied by the bad pixel masks (which are a combination of the image DQ array and the static masks) to create a final mask file for each input image, `*_sci?_mask.pl`, which will be used during final drizzle combination.

Figure 4.14: A single cosmic ray mask `'j8cw54orq_ft_sci2_crderiv.pl'`. This mask should be blinked with the original image `'j8cw54orq_ft[sci,2]'` from Figure 4.10 or the equivalent `'j8cw54orq_ft_sci2.fits'` output file to assure that optimal parameters were chosen in the **driz_cr** task.



7. Performing the Final Drizzle Combination

Parameters	Default	Description
driz_combine	yes	Perform final drizzle image combination?
final_outnx		Size of FINAL output frame x-axis
final_outny		Size of FINAL output frame y-axis
final_kernel	“square”	Shape of kernel function (square, point, gaussian, turbo, tophat, lanczos3)
final_scale	1.0	Linear size of output pixels (relative to input)
final_pixfrac	1.0	Linear size of drop in input pixels
final_rot	0.	Rotation of input image to be applied (degrees anti-clockwise)
final_fillval	INDEF	Value given to undefined output pixels
Output Files	Images= output_sci.fits, output_wht.fits	

When ‘driz_combine=yes’, this step takes the original input images, together with the final cosmic ray masks, and drizzles them all onto a single output image. The standard **drizzle** parameters kernel, scale, pixfrac, and rot can be specified by the user, if desired. By default the scale of the output image is 1.0, but the user is encouraged to experiment with other options (e.g. scale=0.5 and pixfrac=0.7 yields a sharper output PSF).

When the following initial setup parameters are set: ‘build=yes’ (default) and ‘context=yes’ (non-default), the final **MultiDrizzle** output image will be a single multi-extension FITS file named ‘example_drz.fits’. The format of this file is identical to the DRZ product from **PyDrizzle** which is delivered from the archive (see Section 4.3.3) and which contains the science image in extension 1, the weight image in extension 2, and the context image in extension 3. When ‘build=no’, these files will be written to separate output files. When the default value ‘context=no’ is used, no context image is created.

The first extension of the drizzled product contains the science (SCI) image which is corrected for distortion and which is dither-combined (or mosaiced), if applicable. The drizzled SCI image derived from the Tadpole example is presented in Figure 4.15 and is in units of *electrons/sec*. All pixels have equal area on the sky and equal photometric normalization across the field of view, giving an image which is both photometrically and astrometrically accurate for both point and extended sources. The dimensions of the output image are computed on-the-fly by **MultiDrizzle** and the default output plate scale is read from the ‘scale’ column in the IDCTAB. These parameters, however, may be chosen by the user to best suit the actual data.

The second extension of the output image contains the weight (WHT) image. This image gives the relative weight of the output pixels and, in standard processing using the **MultiDrizzle** defaults, it can be considered an effective exposure time map. The weight image from the example is shown in Figure 4.16, where darker areas have higher weight. The chip edges and gaps are clearly visible, as are column defects and cosmic ray features. The bulk of the image is “dark gray” corresponding to the overlap of all six inputs. In this area the weight value is ~ 3140 , equal to the sum of the exposure times of the six images which contribute. There is also a smooth variation across the image due to the variation of the pixel area on the sky caused by the distortion.

The third extension of the **MultiDrizzle** output image contains the context (CTX) image which encodes information about which input image contributes to a specific output pixel. This is done using a bitmask for each output pixel, where ‘bit set’ means that the image, in the order it was combined, contributed with non-zero weight to that output pixel. The context image starts as a single 32-bit integer image but is extended as a cube with additional 32-bit deep planes as required to handle all the input images. The context image for the Tadpole example is shown in Figure 4.17. As there are six input images, each with two chips which are treated separately, this image has 12 bit planes which may be set. The darkest area shown corresponds to the [sci,2] chip from all six inputs and hence has all the following even bits set: $2+8+32+128+512+2048=2730$. Cosmic ray hits or other defective pixels contribute to the appropriate bit plane with zero weight and hence appear as lighter spots.

Figure 4.15: The science (SCI) extension of the drizzled product from the **MultiDrizzle** example. This image has been corrected for distortion and drizzled onto a single mosaic using the six images in the dither pattern.

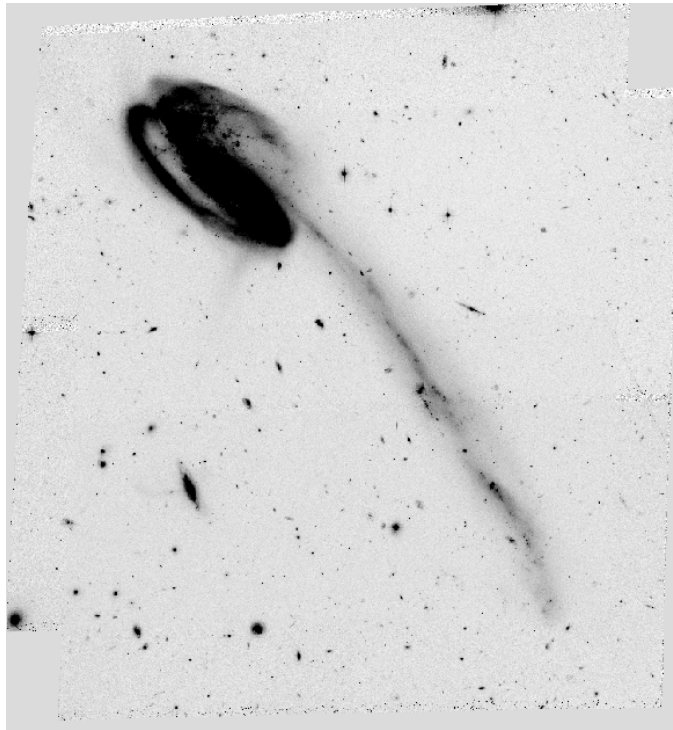


Figure 4.16: The corresponding weight (WHT) extension of the drizzled product from the example.

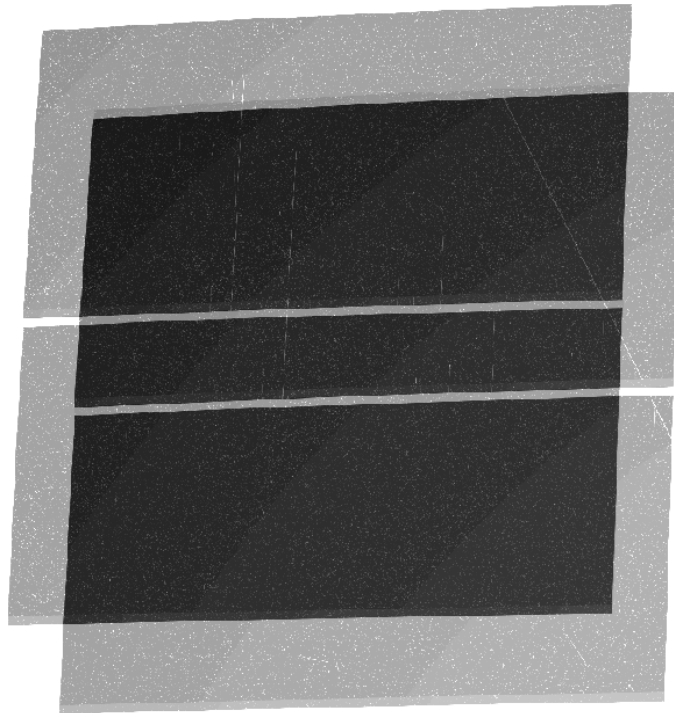
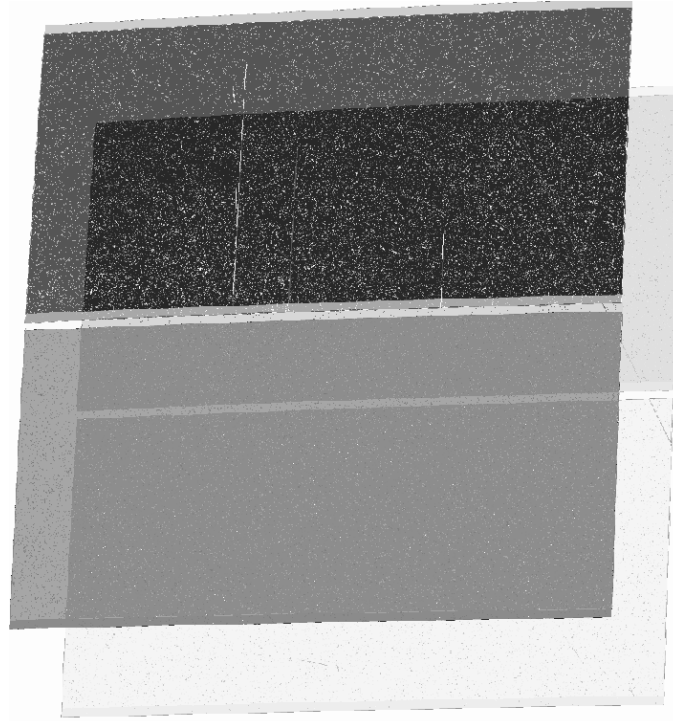


Figure 4.17: The corresponding context (CTX) extension of the drizzled product from the example.



ACS Error Sources

In this chapter. . .

5.1 Readout Noise and A-to-D Conversion / 5-1
5.2 Bias Subtraction / 5-2
5.3 Dark Current, Hot Pixels, and Cosmic Rays / 5-6
5.4 Flat Fields / 5-17
5.5 Image Anomalies / 5-25
5.6 Generic Detector and Camera Properties / 5-37

This chapter describes the sources of uncertainty in ACS data, and provides pointers to resources beyond this manual for the latest descriptions of our evolving understanding of these issues.

5.1 Readout Noise and A-to-D Conversion

5.1.1 Readout Noise

The readout noise is an irreducible contribution to the total error budget. It is linked to the readout process and there are no reduction steps that can minimize or remove it. The effect of the read noise can however be minimized by keeping the number of pixels where signal is measured small and by minimizing the number of CCD readouts (while still allowing for good cosmic ray and hot pixel removal). The read noise is independent of position on the CCD. Table 5.1 lists the read noise (R) expressed in electrons (rms), which is associated with the readout from each amplifier for the main gain settings.

Table 5.1: Readout Noise (R) and Total Noise (Readout+Quantization, TRN) for ACS CCDs

Chip	Amp	Gain 1			Gain 2		
		Gain	R	TRN	Gain	R	TRN
WFC1	A	1.00	4.99	5.00	2.002	5.24	5.27
WFC1	B	0.972	4.89	4.90	1.945	5.08	5.11
WFC2	C	1.011	5.16	5.17	2.028	5.37	5.40
WFC2	D	1.018	4.82	4.83	1.994	5.11	5.14
HRC	C	1.185	4.46	4.47	2.216	4.71	4.78

5.1.2 A-to-D Conversion

The analog information (electrons) accumulated in the CCD are converted into data numbers (DN) by the analog-to-digital converter (ADC). Both ACS CCD cameras employ 16-bit ADCs, which can produce a maximum of $2^{16} = 65,535$ DN. If the gain conversion factor (g) is expressed in (e^-/DN) the largest number of electrons representable by these ADCs is given by $g \cdot 2^{16}$. Any charge packet containing more than $g \cdot 2^{16}$ electrons will reach the digital saturation limit (see Section 5.6.4). The analog-to-digital converter produces only discrete output levels. This means that a range of analog inputs can produce the same digital output. This round-off error is called quantization noise. It can be shown (Janesick J.R., 2001, Scientific Charge-Coupled Devices, SPIE Press.) that the quantization noise QN is constant for a given gain setting when expressed in DN: $QN(DN) = 12^{-1/2}$. The quantizing noise can be converted into noise electrons as: $QN(e^-) = 0.288675 \cdot g$. The total noise (TRN) associated with CCD readout noise (R) and quantizing noise is obtained by adding the two figures in quadrature (see Table 5.1), $TRN = (R^2 + (0.288675 \cdot g)^2)^{1/2}$.

5.2 Bias Subtraction

5.2.1 Bias Calibration Issues

Bias reference frames are acquired daily for scientific calibration purposes and for monitoring the detector performance. Every week multiple bias frames are combined together into a reference superbias image. The combination removes the cosmic rays accumulated during the readout time and enhances the signal-to-noise ratio of the fixed pattern noise.

CALACS performs the bias correction in two steps (see Section 3.4.1): **doBlev** subtracts the bias level from the overscan region and **doBias** subtracts a reference bias image.

Each quadrant of the WFC focal plane (A&B for WFC1 and C&D for WFC2) has two overscan regions: a 24-pixel-wide leading physical overscan at columns 1-24 and a 20-row-wide virtual overscan at rows 2049-2068 obtained by overclocking after the readout of the last CCD row.

All four WFC amplifiers produce a horizontal ramp in the leading overscan which extends up to 18 columns toward the active area (see Sirianni, M., Martel, A.R., & Hartig, G.F., 2001, JHU-ACS internal report, available on-line at:

http://acs.pha.jhu.edu/instrument/calibration/results/by_detector/wfc/build4/overscan/).

The amplitude of the ramp is different for each amplifier, but it is generally larger in the A&D quadrants than in the B&C quadrants. The perturbation associated with the ramp fades gradually and disappears within approximately 15 columns. The bias level from the physical overscan can be safely measured using the last six columns adjacent to the active area [columns 19-24 for A&D and columns 4121-4126 for B&C].

The virtual overscan is not used to estimate the bias level. This region shows large scale structure which is quadrant and gain dependent. This structure, due to the bias fixed pattern noise, makes using the virtual overscan in an automatic routine problematic. Moreover, this region can be contaminated by deferred charge due to degradation in the parallel charge transfer efficiency.

Analysis of the active area in the bias frames shows small differences in the bias level between the leading physical overscan and the active area (Sirianni, et al., 2002, "HST Calibration Workshop" STScI, page 82). This offset varies from amplifier to amplifier and it can be as large as 3.5 DN. Ideally, no offset would be seen. In principle, if the residual offset between the imaging area and the overscan region were always of the same amplitude, a full frame bias subtraction (**doBias**) would remove any residual differences. Unfortunately, the offset is not constant but shows random variations on the order of few tenths of a DN. These variations are likely due to interference between the WFC integrated electronics module and the telescope and/or other science instruments. The accuracy of the bias level subtraction in a single quadrant is limited by this random effect.

HRC images are read out using only one amplifier. Each image has three overscan regions: the 19-pixel-wide leading and trailing physical overscans at columns 1-19 and 1044-1062, respectively, and the 20-row-wide virtual overscan at rows 1025-1044. The readout amplifier produces a horizontal ramp in the leading overscan which extends up to 10 columns toward the active area. As in the case of WFC images, the bias level is therefore measured in the last six columns of the leading physical overscan. There is not a significant difference between the bias level in the overscan area and in the active area.

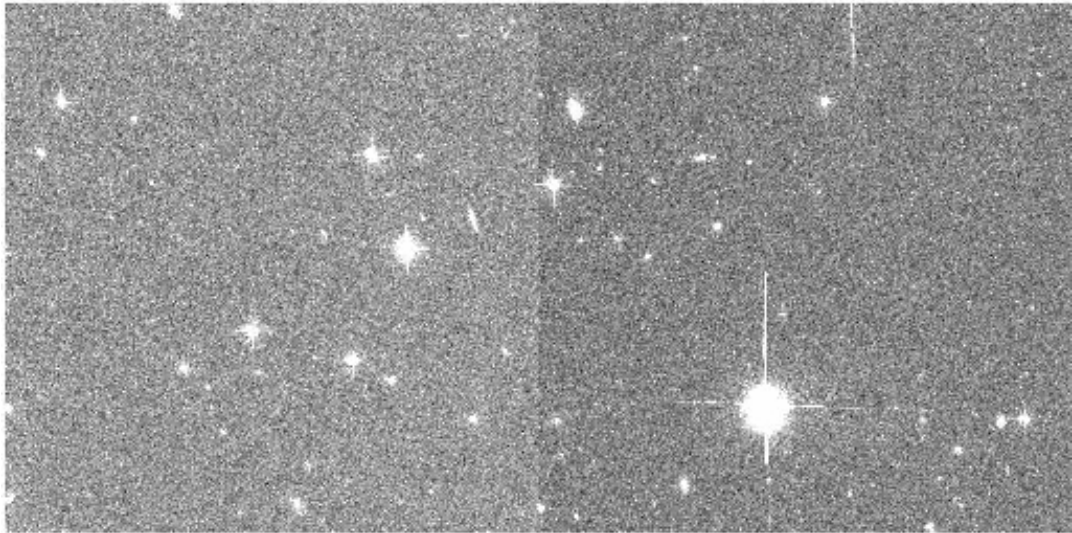
Even after the subtraction of the overscan bias level, some structure remains in the bias frames. This structure includes real bias pattern noise

and also some dark current accumulated during the time required to read out the detector. To remove this structure a superbias is subtracted from each science image.

In principle, a perfect bias subtraction would give an image, that once flat fielded and converted to electrons should not present any discontinuity at the boundaries between quadrants.

However, in the case of WFC images the amplitude of the residual offset level varies between each quadrant. This results from the bias level being measured in the leading physical overscan and then subtracted from the active area of each quadrant, causing the center of the resulting image to show a jump between the two adjacent quadrants, A&B or C&D (Sirianni, et al., 2002, "HST Calibration Workshop" STScI, page 82). This jump is not removed by the superbias subtraction. Indeed, since this offset (usually with a maximum amplitude of 0.3 DN) is present in all calibration and scientific frames, the final product of **CALACS**, after bias and dark subtraction, may show a quadrant-to-quadrant jump as large as few DN's (Figure 5.1).

Figure 5.1: Calibrated WFC1 image showing the quadrant-to-quadrant jump



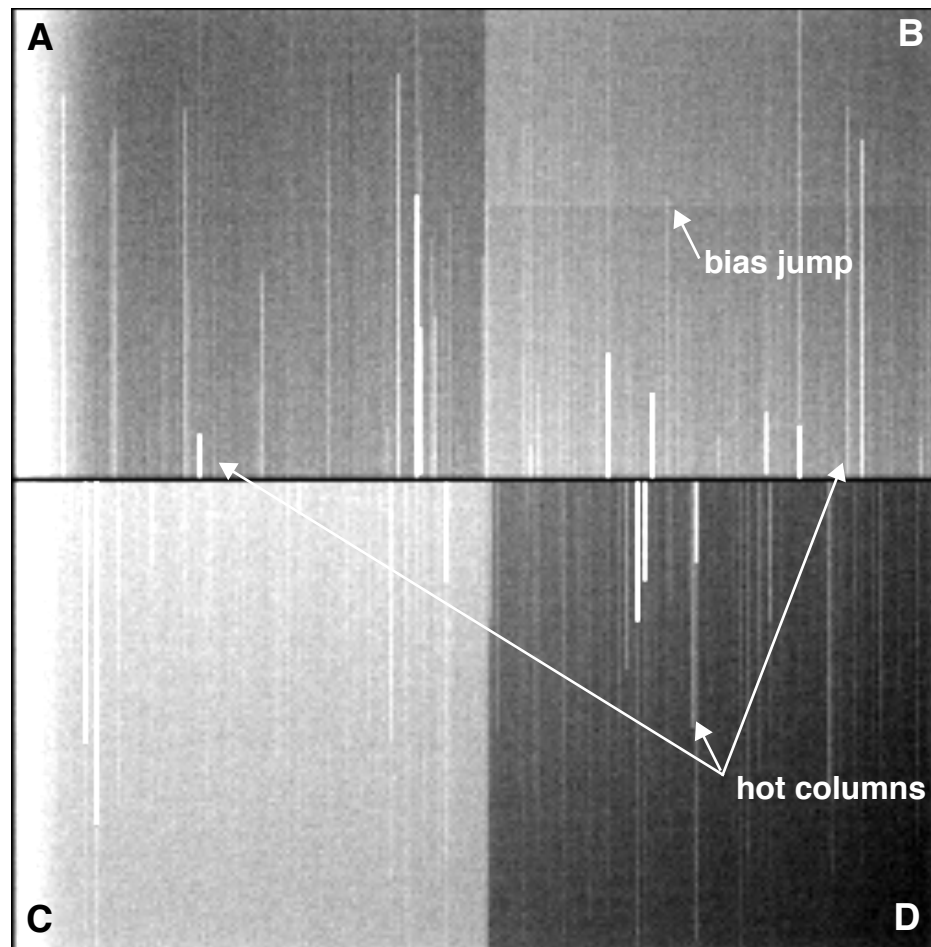
For point source photometry, the local background in an annular region is typically subtracted so that the residual offset has essentially no impact on the integrated magnitudes. However, if the object falls across the quadrant jump the photometry should be considered suspect. Similarly, extended objects may spread over two or more quadrants, so their surface brightness profile will suffer from the residual offset. More studies are in progress to better characterize this problem and to develop a correction that can be applied directly to the calibrated images. At the moment we suggest fitting the sky level in each quadrant separately.

A detailed study of the on-orbit characteristics of bias frames, the overscan regions and superbias used in the pipeline will be published in a future ISR.

5.2.2 Bias Jump

There is some evidence for intermittent variations at the sub-DN level in the WFC bias level during the readout. On a few occasions some frames show horizontal bias jumps in one or more quadrants that can run for several hundreds of rows (Figure 5.2). These effects occur at a few tenths of a DN level and they might appear in one or more quadrants simultaneously. The probable source is interference from other activities on the spacecraft and/or scientific instruments with the electronics. At the moment there is no automatic detection of bias jumps within the calibration pipeline. Though bias jumps at the sub-DN level are not important for most applications, users should be cautioned that they might exist in processed data and we recommended a careful inspection of the data for their presence.

Figure 5.2: Bias Jump in the WFC1 Quadrant B only. The vertical stripes in the data are hot columns and are unrelated to bias.



5.2.3 Bias Subarrays for the WFC and HRC

When science data are obtained with use of subarrays, dark and flat field images will be extracted from full-frame images. Bias images will also be extracted from full-frame images, if the subarray is contained in a single quadrant. Tests have shown that this does not degrade the quality of the dark, flat field or bias corrections as compared to full-frame data. It is strongly advised that subarrays not be defined such that they span amplifier quadrants. In the case that the subarray extends into the adjacent amplifier quadrant, users must obtain their own subarray bias images (with one-amp readout), otherwise the bias level difference between adjacent amplifier quadrants would be applied inappropriately.

5.3 Dark Current, Hot Pixels, and Cosmic Rays

5.3.1 Dark Current

The procedure for creating ACS dark reference files and applying dark subtraction to ACS science data is described in detail in Section 3.4.3. Because it takes a few weeks to collect enough frames to create the dark reference file, the "best" darkfile is typically not available in the pipeline until 2-3 weeks after the date of observation. Users may verify whether the darkfile most appropriate to their observations has been installed for pipeline use by checking the ACS reference file web pages:

http://www.stsci.edu/hst/acs/analysis/reference_files/drkim_age_list.html.

Using an old reference file will produce a poor dark correction: either leaving too many hot pixels uncorrected and unflagged, or creating many negative "holes" caused by the correction of hot pixels which were not actually hot in the science data (i.e., if the detectors were annealed in the interim).

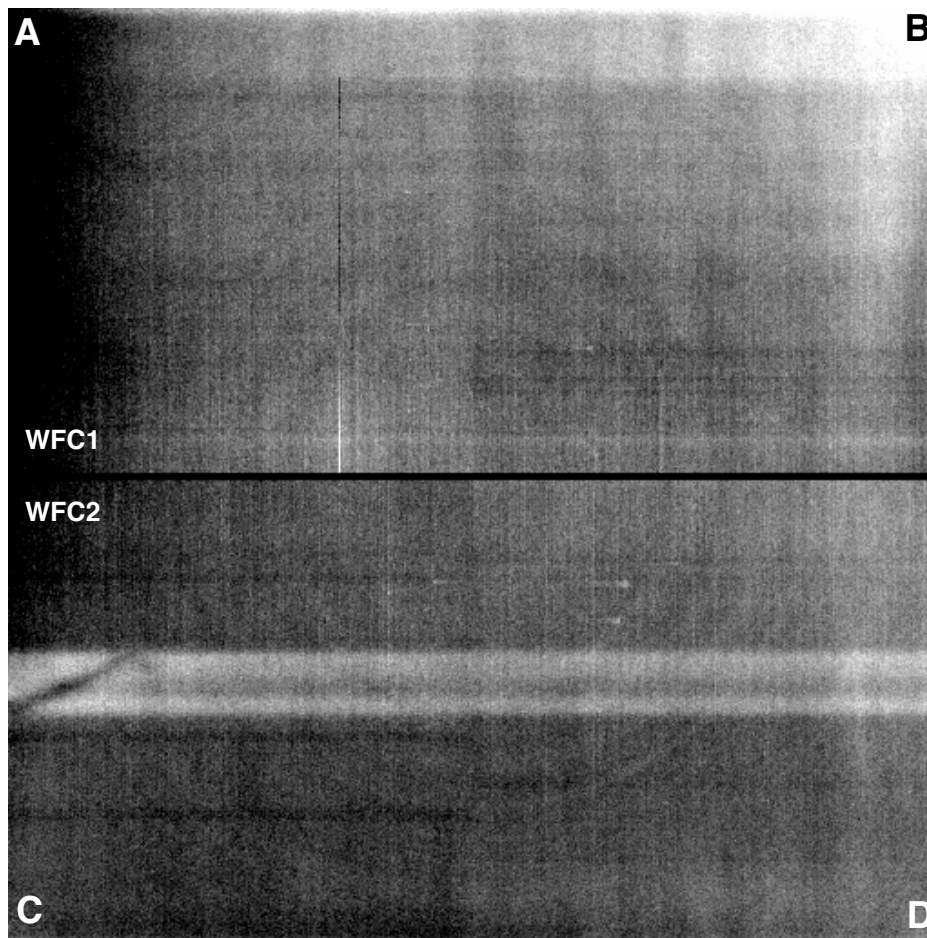
WFC/HRC

The dark current is not constant across the CCDs. Figure 5.3 shows the dark current features in the WFC1 (above) and WFC2 (below) chips. These features were observed in pre-flight tests, and have generally remained stable in orbit.

There is a gradient, most noticeable on the WFC1 chip, going from a dark edge in the amplifier A quadrant (upper left) to a bright corner in the amplifier B quadrant (upper right). There are two horizontal bright bands of elevated dark current in the center of the WFC2 chip. Many faint rings are also visible: all concentric with the center of both chips. These features are

likely intrinsic to the chips themselves: artifacts embedded in (or on) the silicon during various stages of the CCD manufacturing process.

Figure 5.3: High S/N combination of WFC dark frames illustrating dark current structure



The WFC1 and HRC histograms in Figure 5.4 and Figure 5.5 show the growth of hot pixels (for more information, refer to Section 5.3.2). A less obvious result is that the peak of the normal pixel (Poisson) distribution (i.e., the mean dark current, excluding the hot pixels) has also increased.

The increase in the mean dark current for WFC has been from 6.8 $e^-/\text{pixel}/\text{hr}$ at launch to 11.1 $e^-/\text{pixel}/\text{hr}$ (an average of 11.4 for WFC1 and 10.8 for WFC2) in April 2004. For HRC the change in dark current has been from 9.3 $e^-/\text{pixel}/\text{hr}$ at launch to 13.4 $e^-/\text{pixel}/\text{hr}$ in April 2004. The dark calibration tracks the mean dark current very closely at 2 week intervals (see Section 3.4.3).

Figure 5.4: WFC dark current histogram for chip 1. Statistics for chip 2 are nearly identical. The figures illustrate the growth of hot pixels over time.

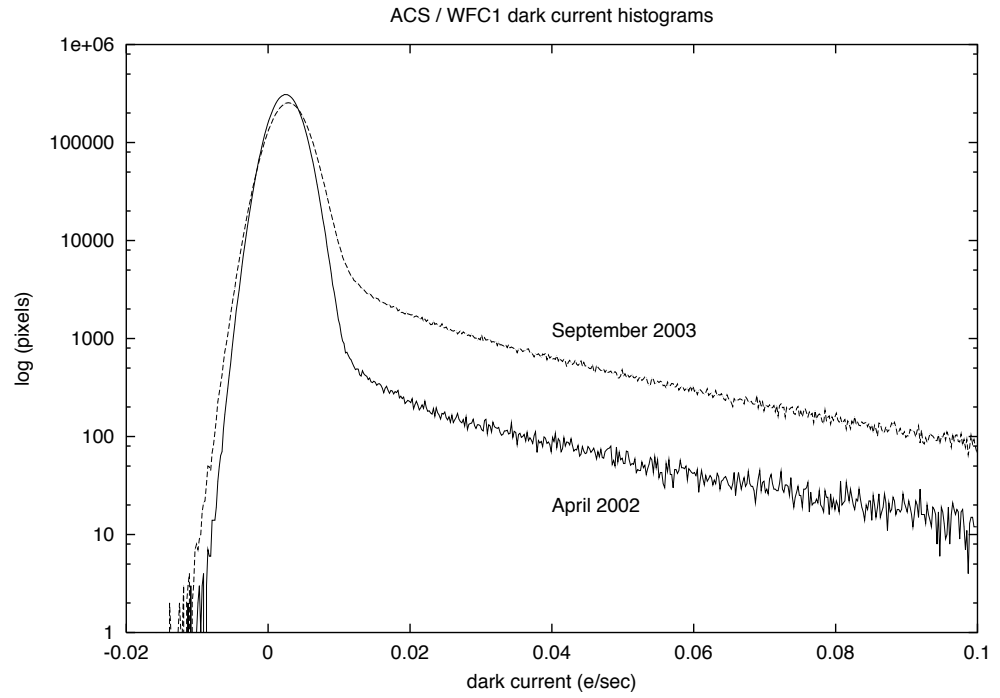
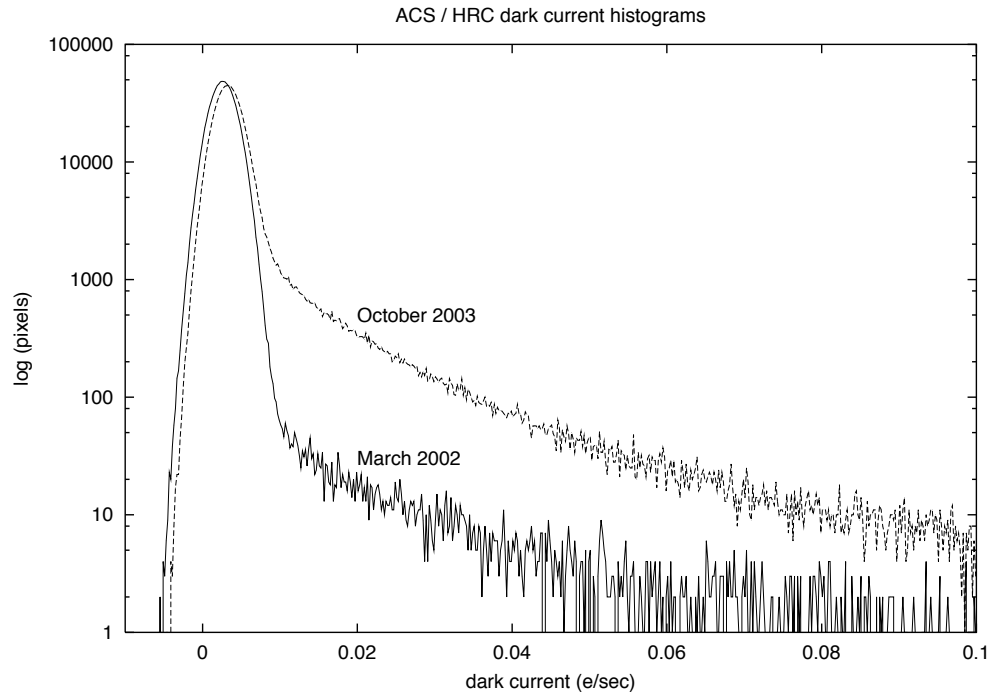


Figure 5.5: HRC dark current histogram



SBC

The SBC, or ACS MAMA detector, intrinsically has no readout noise and very low detector noise levels which normally will be negligible compared to statistical fluctuations. A dark current image was compiled during ground tests with a total exposure time of 14,400 seconds and remains as the calibration dark image. This has a mean dark count of 3.8×10^{-5} counts/pixel/second. Approximately once per month since ACS was installed, 1800 second SBC darks have been accumulated. The dark rate for these is typically 2.0×10^{-5} counts/pixel/second. The principal reason for taking these images is to provide a measure of the health of the SBC. No major fluctuation has been noted. An in-flight super-dark will be constructed from the in-flight measurements.

5.3.2 Hot Pixels

A “hot pixel” is a state of radiation damage affecting the dark current production of an individual pixel. Such pixels likely result from the close interaction of an incident heavy nucleon with Si nuclei in a pixel, creating new Si-SiO₂ interface states. Once produced, such pixels almost always remain hot before the next anneal producing a continuum of excess dark current rates more than 1 to 4 orders of magnitude greater than the mean value. Although such pixels have temporarily lost their ability to yield precise measurements of faint photon flux, the majority of such pixels are “healed” in monthly anneals when the CCD temperature is raised by ~100 degrees over operating temperature for ~16 hours.

As seen in Figure 5.4 and Figure 5.5, the dark current distribution is well described by a Gaussian. As expected from experience with earlier HST cameras, very significant tails (all pixels with dark current above the 5 sigma limit [~ 0.02 e⁻/sec]) are seen in these distributions. These tails consist of the pixels that have suffered radiation damage. The pixels with values between 0.02 and 0.08 e⁻/sec are referred to as “warm” pixels and those above 0.08 e⁻/sec as hot pixels. Figure 5.6 and Figure 5.7 show these tails for WFC and HRC for two different dates. As shown, many thousands (HRC) or tens of thousands (WFC) of pixels have greatly elevated dark current with the number markedly increasing with time.

Figure 5.6: Dark Current Distribution Tails in WFC for March 15, 2002 and March 28, 2002. All pixels with dark current above the 5 sigma limit (~ 0.02 e⁻/sec) are due to so-called "warm" and "hot" pixels which have suffered radiation damage. The number of these pixels can be seen to increase rapidly with time.

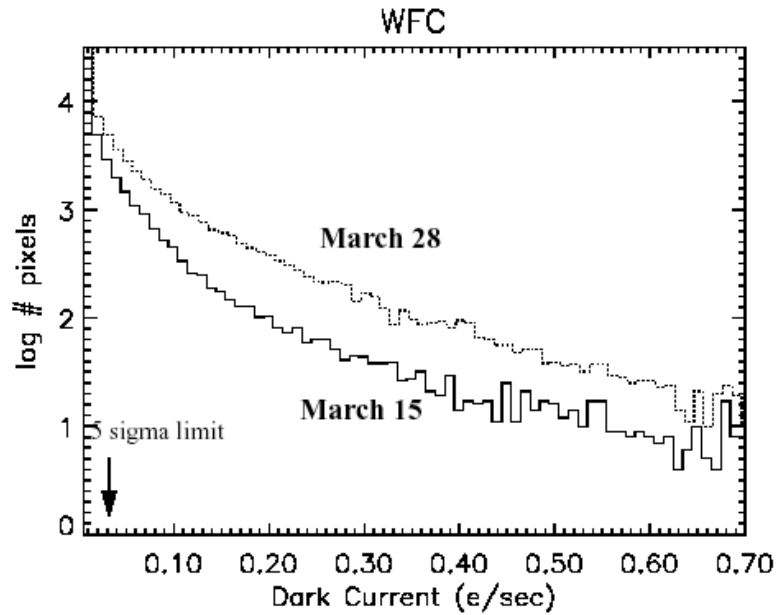
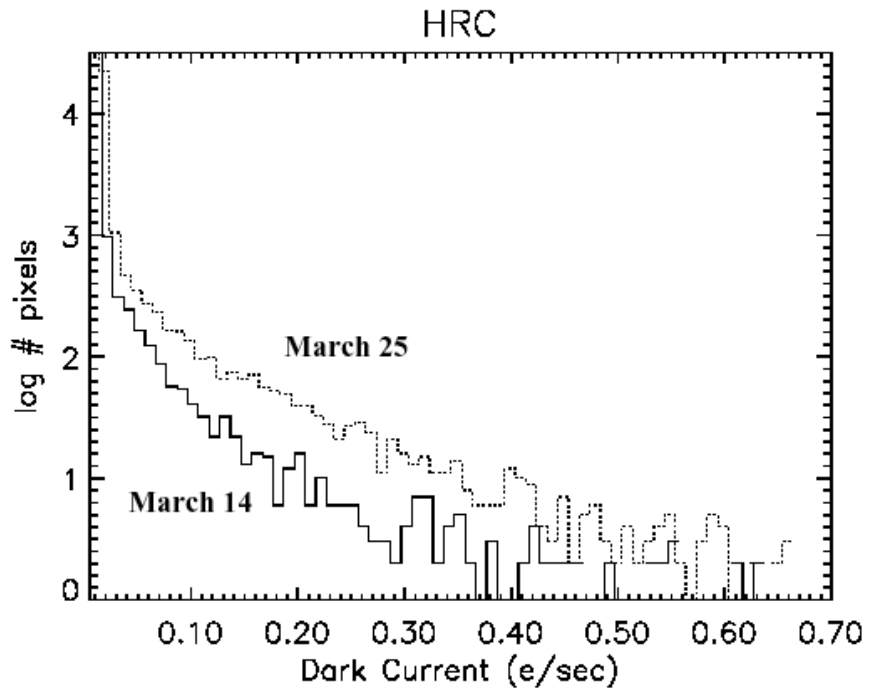


Figure 5.7: Dark Current Distribution Tails in HRC for March 15, 2002 and March 28, 2002.



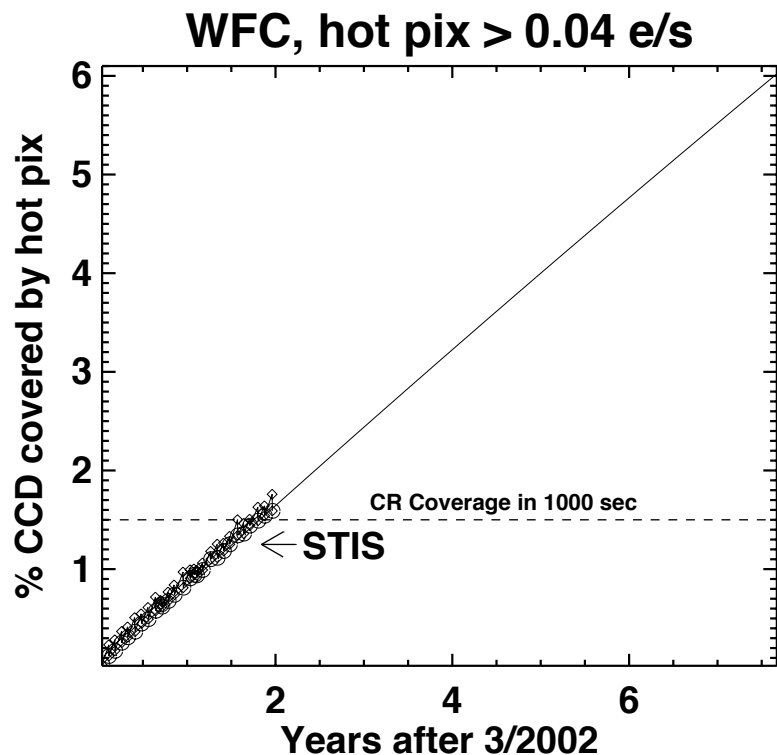
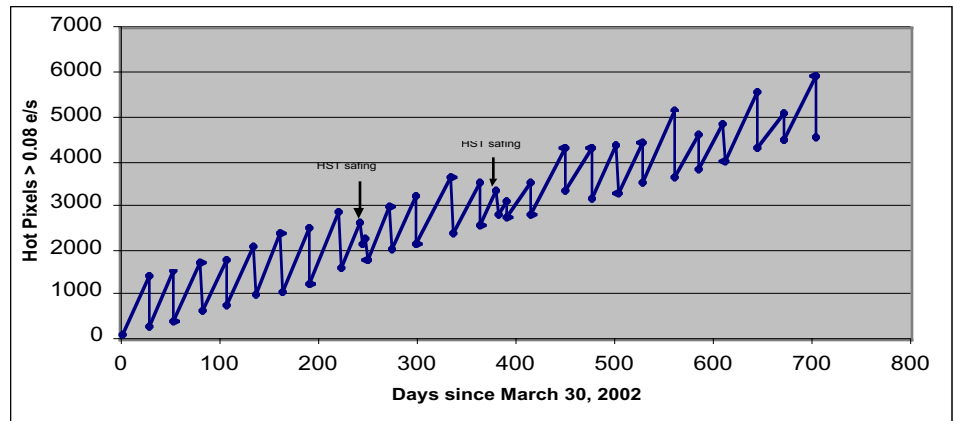
Trending

Even with 16 million pixels, the WFC would not be expected to have any pixels with dark current more than 6 standard deviations beyond the mean (i.e., with dark current greater than $0.02 \text{ e}^-/\text{sec}$) were it not for the effects of radiation damage. To analyze hot pixel trending, we have chosen a conservative limit of twice that, or $0.04 \text{ e}^-/\text{sec}$ for WFC and $0.08 \text{ e}^-/\text{sec}$ for HRC as a threshold above which we consider a pixel to be “hot” and not part of the normal distribution of pixel dark current. Note that even though trending is analyzed at these limits, only pixels above $0.08 \text{ e}^-/\text{sec}$ are flagged in the DQ array.

For WFC we find a growth rate of approximately 1200 new hot pixels per day with dark current greater than $0.04 \text{ e}^-/\text{sec}$. For HRC the number of new hot pixels per day above the threshold is approximately 90. Because the distribution of dark current in hot pixels is strongly concentrated near the threshold, the specific number of such pixels is necessarily a strong function of the chosen threshold.

Subtraction of a superdark frame from a science image can remove the dark current from hot pixels just as it does for normal pixels. However, hot pixels are often orders of magnitude noisier than normal pixels, which in many cases limits their ability to provide useful measurements of flux. For the hottest pixels, the observed noise exceeds even the Poisson noise by an order of magnitude. Observations which do not employ contemporaneous superdarks will suffer from uncorrected and unflagged hot pixels, a particular concern to programs whose aim is to search for astrophysical transients or broad-band drop-outs. The problems associated with hot pixels can be mitigated by using the best reference files (usually necessitating recalibration with OTFR several weeks after the data were obtained), flagging and discarding data from hot pixels, and dithering observations to provide additional sampling of pixel positions on the sky and allow rejection of hot pixels.

Figure 5.8: Hot pixel growth rate for HRC (top) and WFC (bottom).



The anneal rate of new hot pixels (dark current $> 0.04 \text{ e}^-/\text{sec}$) on the ACS WFC has been disappointingly low: $\sim 60\%$ in the first eight monthly anneal cycles of the instrument. This rate is significantly lower than the observed and characteristic value for other HST CCDs: 80-85% for WFPC2, STIS, and ACS HRC. The anneal rate is significantly less than 60% for pixels which are much hotter than $0.04 \text{ e}^-/\text{sec}$. The consequence of poor annealing is a greater fractional coverage of the camera by pixels with elevated dark current than was the experience of other HST CCDs. Figure 5.8 shows the observed growth of hot pixels on both HRC and WFC. The saw-tooth appearance is due to regular anneals. For WFC hot pixel coverage is more than cosmic ray coverage in a typical exposure. By the

end of the decade one in every 16 pixels is predicted to be hot on WFC. For more details see [ACS ISR 02-09](#).

5.3.3 Cosmic Rays

Like all HST cameras before it, the ACS HRC and WFC fields of view are heavily peppered by cosmic rays in even the shortest of exposures. For full orbit integrations, approximately 5% of the pixels receive significant charge from cosmic rays via direct deposition or from diffusion from nearby pixels. Great care must be taken in planning and analyzing HST ACS observations to minimize the impact of cosmic rays on science images.

Fractional Coverage

For most users of the HRC and WFC, the most important characteristic of cosmic rays is simply the fraction of pixels they impact. This number provides the basis for assessing the risk that the target(s) in any set of exposures will be compromised. For ACS the observed rate of cosmic ray impacts on an individual frame varies by a factor of two depending on the proximity of the spacecraft to the confluence of the Earth's magnetic field lines (e.g., the South Atlantic Anomaly). For a 1000 second exposure the fraction of pixels affected by cosmic rays (in non-SAA passages) varies between 1.5% and 3%. This fraction is the same for the WFC and HRC despite their factor of two difference in pixel areas, because the census of affected pixels is dominated by charge diffusion, not direct impacts. This fraction is also consistent with what is observed for WFPC2. For most science observations, a single CR-split (i.e., two exposures) is sufficient to insure that measurements of the targets are not compromised by cosmic rays.

More consideration is required for survey-type observations with WFC, a bonafide survey instrument. Observers seeking rare or serendipitous objects as well as transients may require that every single WFC pixel in at least one exposure among a set of exposures is free from cosmic ray impacts. For the cosmic ray fractions of 1.5% to 3% in 1000 sec, a single ~2400 second orbit must be broken into 4 exposures (4 CR-splits of 500 to 600 sec each) to reduce the number of uncleanable pixels to 1 or less. (It is also recommended that users dither these exposures to remove hot pixels as well.)

Cosmic Ray Events

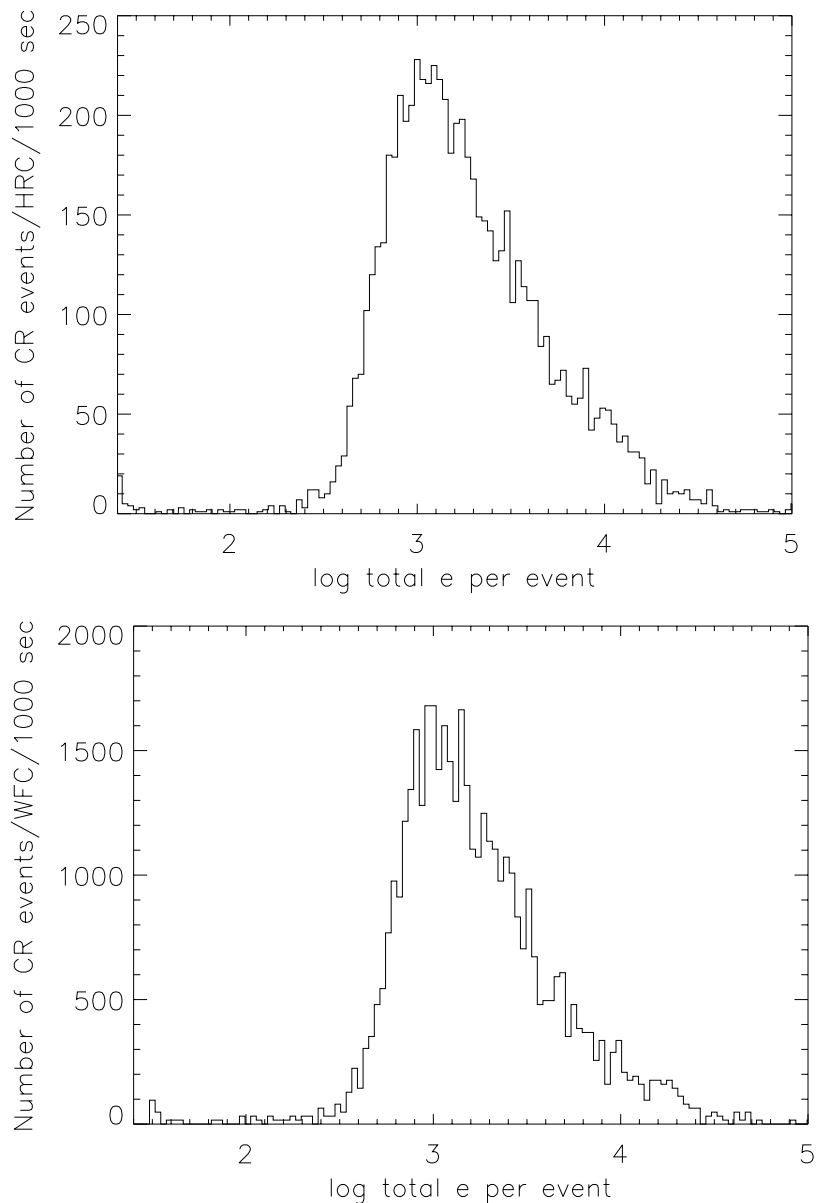
Many science observations require a careful consideration of individual cosmic rays events. To either remove cosmic rays events or distinguish them from astrophysical sources users might consider the distributions of observed cosmic ray fluxes, sizes, anisotropies, and the number of attached pixels per event.

Electron Deposition

The flux deposited on the CCD from an individual cosmic ray does not depend on the energy of the cosmic ray but rather the length it travels in the silicon substrate. As a result the deposition distribution has a well-defined minimum with few events of less than 500 electrons (where such low-electron events correspond to cosmic rays which pass through the CCD at a normal angle of incidence). As seen in Figure 5.9, the median charge deposited for WFC and HRC is about 1000 electrons, the same as for WFPC2.

A useful characteristic of the deposition distribution is its well-defined minimum; e.g., multi-pixel events which have an apparent magnitude of 25th or fainter, in a 500 second broad-band exposure, are unlikely to be caused by cosmic rays. Such information can help with the removal of false positives from searches for faint transients (e.g., high-redshift SNe).

Figure 5.9: Electron deposition by cosmic rays on HRC (top) and WFC (bottom). A minimum deposition of ~ 500 e is seen corresponding to CR's with normal incidence. The median deposition is ~ 1000 e.

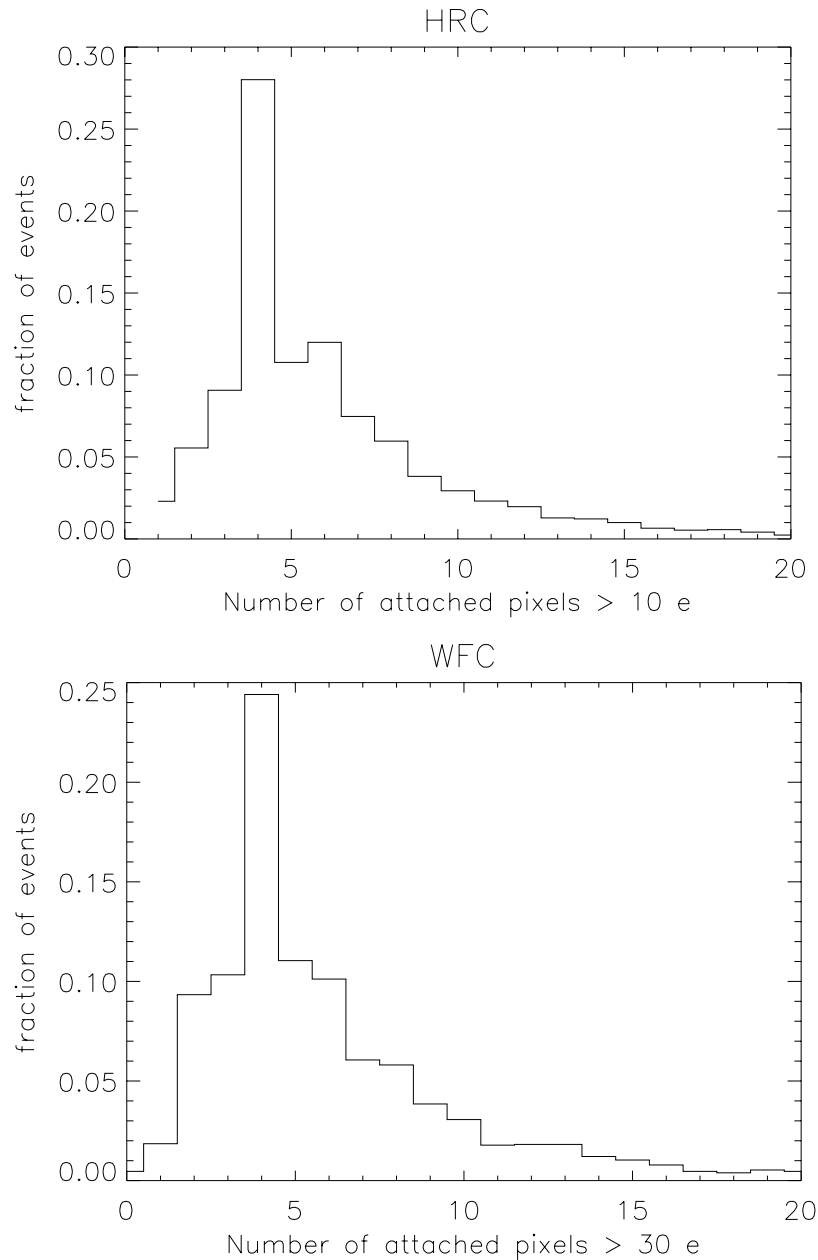


Attached Pixels

As seen in Figure 5.10 for HRC and WFC the salient features of electron deposition are a strong peak in the distribution function at 4 to 5 pixels. On the smaller side there is a sharp decline in events. Although a few events are seen which encompass only one pixel, examination of these events indicate that at least some and maybe all of these sources are actually transient hot pixels or unstable pixels which can appear hot in one exposure (with no charge diffusion) and normal in the next. Such pixels are very rare but do exist. There is a long tail in the direction towards increasing

numbers of attached pixels. Some of these are likely due to two individual events associated by their chance superposition, but more are from oblique incidence cosmic rays which skim the surface of the CCD leaving a long trail (which is wider near the surface). Unfortunately the number of attached pixels is not a very useful characteristic to distinguish cosmic rays from unresolved astrophysical sources.

Figure 5.10: Distribution of the number of pixels associated with a single CR event for the HRC (top) and WFC (bottom). Some bias exists for events >6 pixels which may be composed of two events with chance superposition.



5.4 Flat Fields

5.4.1 Ground Flats (P-flats)

In early 2001, flat field images for the ACS were produced in the laboratory using the Refractive Aberrated Simulator/Hubble Opto-Mechanical Simulator (RAS/HOMS). The RAS/HOMS is an HST simulator capable of delivering OTA-like, external, monochromatic point-source and broad-band, full field illumination above its refractive cutoff wavelength of ~ 3500 Å. Because the RAS/HOMS optics are opaque below 3500 Å, flats for the UV filters F330W and F344N (see [ACS ISR 03-02](#)) were created using inflight observations of the bright earth. (See Section 5.4.4 for more information on the ACS Earth flat program.)

The resulting flat fields include both the low frequency (L-flat) and the high frequency pixel-to-pixel (P-flat) structure. A total signal of about 100,000 electrons per pixel is required for each flat field to avoid degrading the intrinsic pixel-to-pixel rms response of $<1\%$ for the ACS CCD detectors. The flats are normalized by dividing by the average number of counts in the central 1% of the frame. In the case of full WFC frames, the chip 2 images are divided by the chip 1 central value, in order to preserve the overall sensitivity difference between the two CCD chips across the ~ 50 pixel gap that separates the two independent pieces of the WFC detector. For small filters that cover just part of one quadrant of the WFC (i.e., F892N, polarizers), flats were masked to unity below 90% of the central value of the data, so that no flat field correction is done on the scattered light outside the physical edge of the filter.

Because of geometric distortion effects, the area of the sky seen by a given pixel is not constant; therefore, observations of a constant surface brightness object will have count rates per pixel that vary over the detector, even if every pixel has the same sensitivity. In order to produce images that appear uniform for uniform illumination, the observed flat fields include the effect of the variable pixel area across the field. A consequence of dividing by these flat fields is that two stars of equal brightness do not have the same total counts after the flat fielding step. Thus, point source photometry extracted from a flat fielded image must be multiplied by the effective pixel area map (see Section 6.1.3). This correction is accounted for in pipeline processing by **PyDrizzle** (see Section 3.4.3), which uses the geometric distortion solution to correct all pixels to equal areas. In the drizzled images, photometry is correct for both point and extended sources.

More information on the HRC and WFC ground flats may be obtained from [ACS ISR 01-11](#) for the standard filters, polarizers, and coronagraph, from [ACS ISR 02-01](#) for the ramp filters, and from [ACS ISR 02-04](#) for the prism and grism. Ground flats for the SBC detector are described in [ACS ISR 99-02](#).

5.4.2 Inflight L-flat Correction

The large scale uniformity of the WFC and HRC detector response, as provided by the ACS laboratory flats, has been improved in flight by using multiple dithered pointings of stars in 47 Tucanae. By placing the same stars over different portions of the detector and measuring relative changes in brightness, low frequency spatial variations in the response of each detector have been measured. Photometric errors of up to $\pm 9\%$ corner-to-corner have been found in the original WFC and HRC laboratory flat fields (see [ACS ISR 02-08](#)). The derived L-flats are based on a 4th-order polynomial fit and are shown in Figure 5.11 and Figure 5.12 for WFC and HRC, where white indicates that the photometry produced using the laboratory flats is too faint with respect to the true stellar magnitude, and black indicates that the photometry is too bright. There is a continuous gradient in the L-flat correction along the diagonal of the detector which corresponds to the axis of maximum geometric distortion.

L-flats were determined from in flight observations using filters F435W, F555W, F606W, F775W, F814W, and F850LP for both the WFC and HRC. The HRC study included two additional filters: F475W and F625W. The L-flat correction for the remaining filters was derived by using linear interpolation as a function of wavelength. The pivot wavelength of each filter was used for the interpolation, where the resulting L-flat is equal to the weighted average of the L-flat for the two filters nearest in wavelength. For a detailed discussion of ACS L-flat corrections, refer to [ACS ISR 02-08](#). For a discussion of the mathematical algorithm used to derive the L-flats, refer to [ACS ISR 03-10](#).

To verify the interpolated L-flat corrections, additional observations of the same stellar field have been acquired and are undergoing analysis. The L-flat calibration program revisits the same target several times per year; therefore, observations are obtained for a range of orientations due to the roll of the telescope. Differential photometry of stars falling on different portions of the detector as the telescope rolls will provide a verification of the L-flats for the interpolated filters and will also provide an independent test of the absolute sensitivity dependence with time for the full suite of ACS filters. Initial testing indicates that the photometric response for a given star is now the same to $\sim 1\%$ for any position in the field of view for filters which were observed during the in flight L-flat campaign. For the interpolated L-flats, the expected accuracy is $\sim 2-3\%$. Once the analysis for the additional stellar observations is complete, new flat fields will be delivered to the pipeline and the errors are expected to be reduced to $< 1\%$.

Figure 5.11: WFC low frequency (L-flat) flat-field corrections required for the laboratory data. While the pixel-to-pixel (P-flat) structure of the laboratory flats is robust, a low frequency correction is required to achieve uniform detector response. This correction ranges from +/- 5% for the F555W filter to +/- 9% for the F850LP filter.

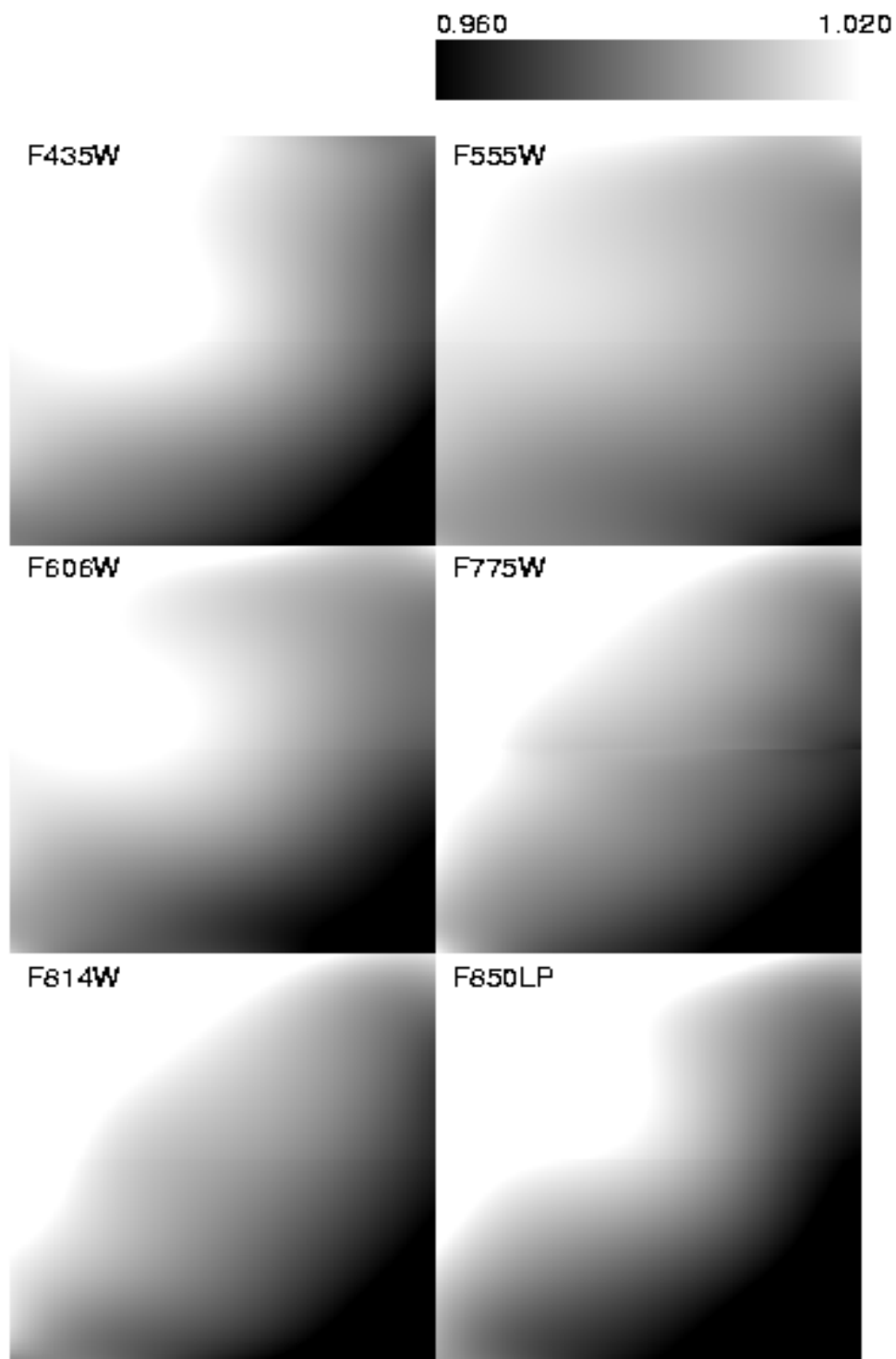
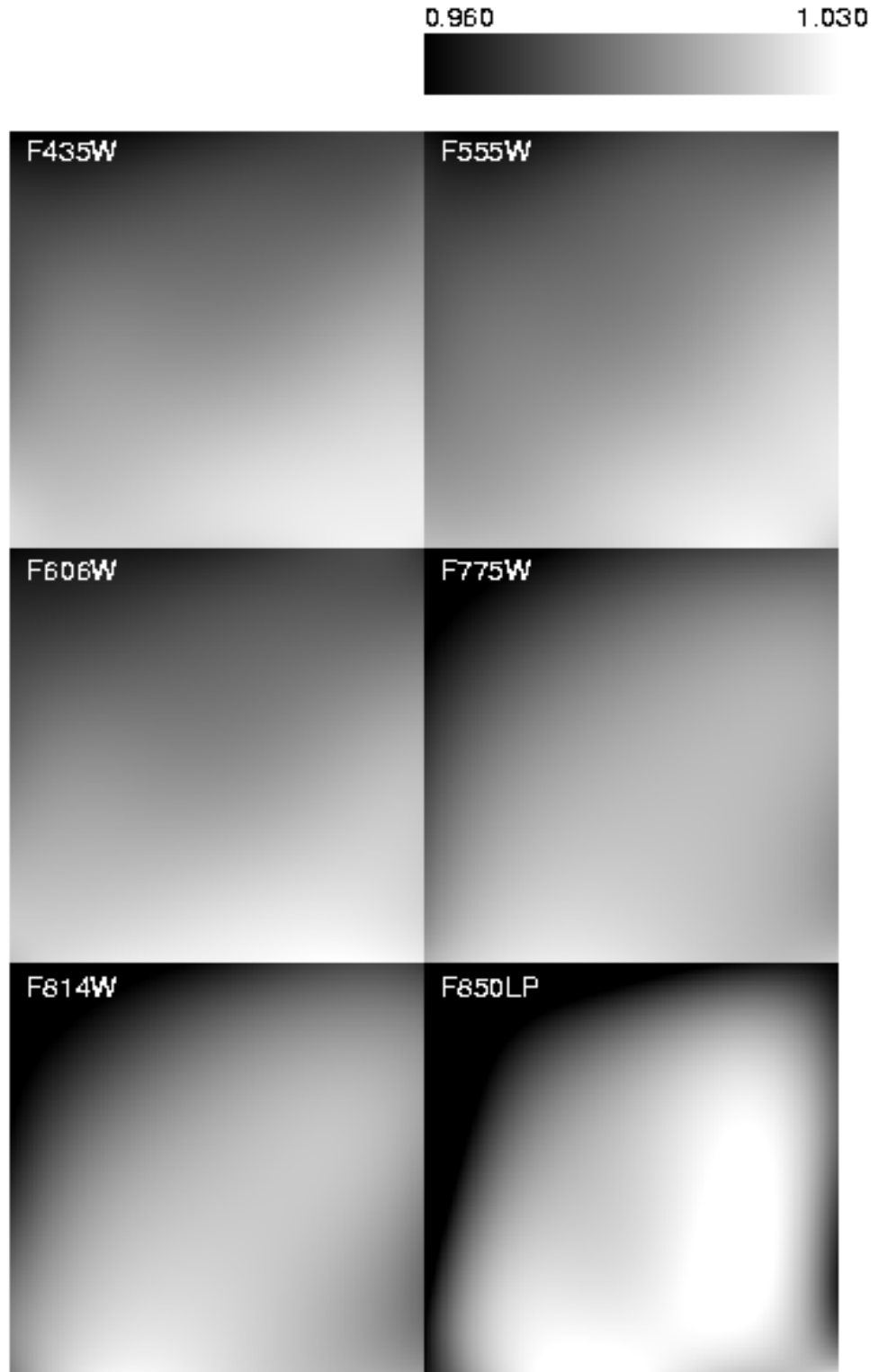


Figure 5.12: HRC low frequency (L-flat) flat-field corrections required for the laboratory flats. This correction ranges from +/- 3% for the F555W filter to +/- 6% for the F850LP filter.



5.4.3 Pipeline Flats

The inferred low-frequency (L-flat) corrections were applied to the laboratory flats, and the resulting flat fields have been in use in the calibration pipeline. Observers are encouraged to check the ACS web site for the most recent flat fields available for recalibration at http://www.stsci.edu/hst/acs/analysis/reference_files/flatimage_list.html

Figure 5.13 shows the corrected WFC laboratory flats for several broadband filters. Note that on the sky a gap of ~ 50 pixels exists between the top and bottom halves that is not shown here. The central donut-like structure is due to variations in chip thickness (see ACS ISR 03-06) and is dependent on wavelength. Pixels in the central region, for example, are less sensitive than surrounding pixels in the blue F435W filter and more sensitive in the red F850LP filter.

The HRC corrected laboratory flats for the same broadband filters are shown in Figure 5.14. The donut-like structure seen in the WFC response is not found in the HRC flats. The small dark rings are shadows of dust on the CCD window (see Section 5.5.1). The large dust mote seen in the WFC F606W flat is due to dust on the F606W filter. That portion of the filter is not "seen" by the HRC detector.

The accuracy of pipeline flats can be verified using a variety of complementary methods. In Section 5.4.2, we discussed how follow-up observations of the same stellar field can be used to verify the derived L-flat corrections. Alternately, observations of the bright earth can provide a uniform flat field source for the complete OTA optical complement and incorporate both the low frequency L-flat and the high frequency pixel-to-pixel P-flat response. Earth flats are described in Section 5.4.4. For most filters, the flats agree to within $\sim 1\%$, except for the interpolated L-flat filters which usually agree to within $\sim 2\%$.

The third method for verifying the ACS pipeline flats is with sky flats. These can be made by filtering and summing many observations of a sparse field. Sky flats have been created for a few of the most frequently used broadband WFC filters and are discussed in detail in Section 5.4.5. The sky flats are generally similar to the corrected ground flats at the 1% level, in accordance with the results of the previous two methods. While the WFC can show residuals in the central donut-region which are as large as 2%, these are most likely due to differences in the color of the spectrum of the sky from that of the bright globular cluster stars used for the L-flat determination.

To summarize, the pipeline flats were created by correcting the laboratory flats by low-frequency corrections derived from dithered stellar observations. For the most used modes, the flats are accurate to better than 1% across the detector. For a few filters, improved L-flats will be derived from additional stellar observations to improve the flat field errors to $<1\%$ for all modes. Additional verification of the pipeline flats will be provided by both earth flats and sky flats, thus confirming the derived L-flats and setting limits to their dependence on the color of the source spectrum.

Figure 5.13: WFC Flat Fields

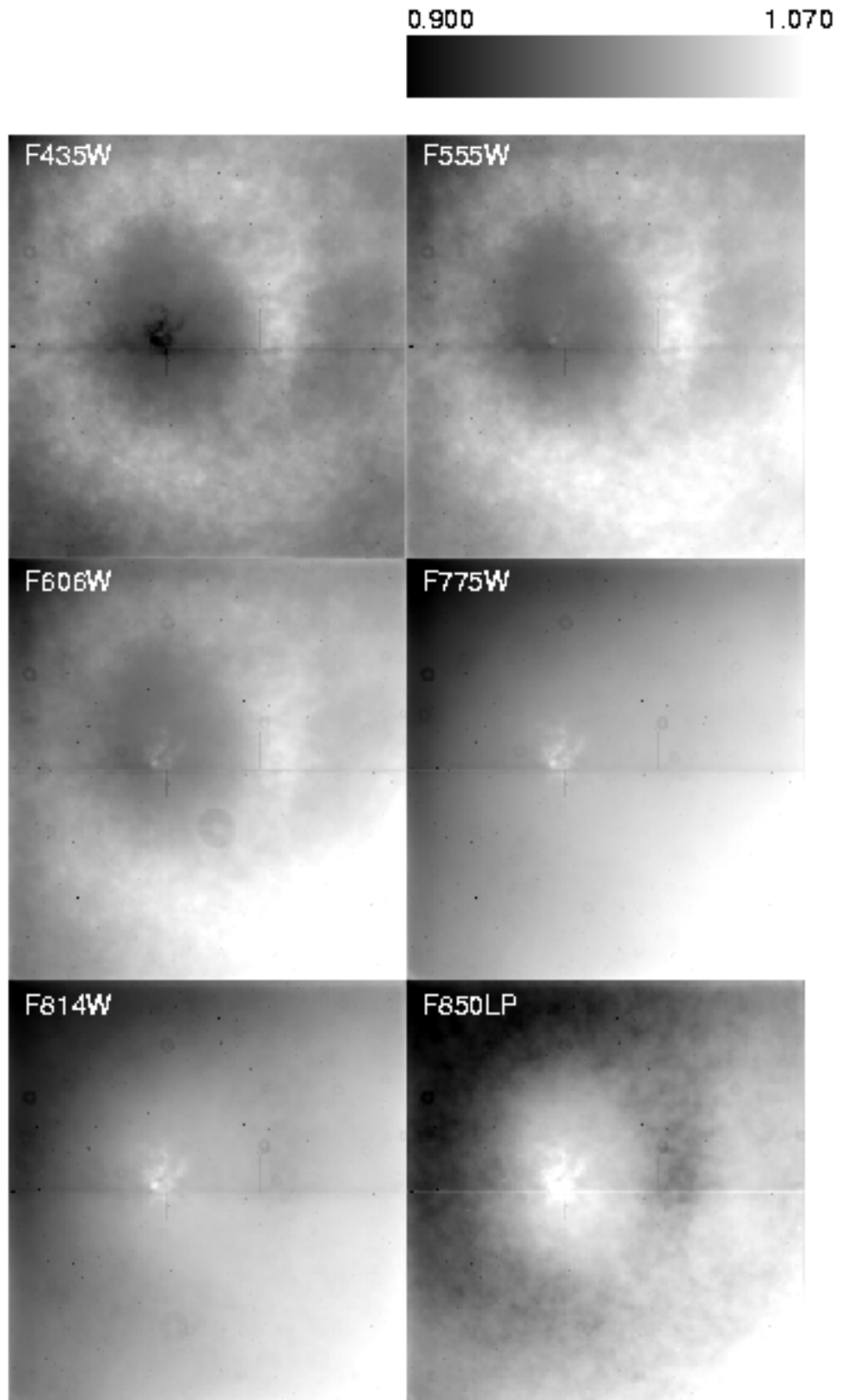
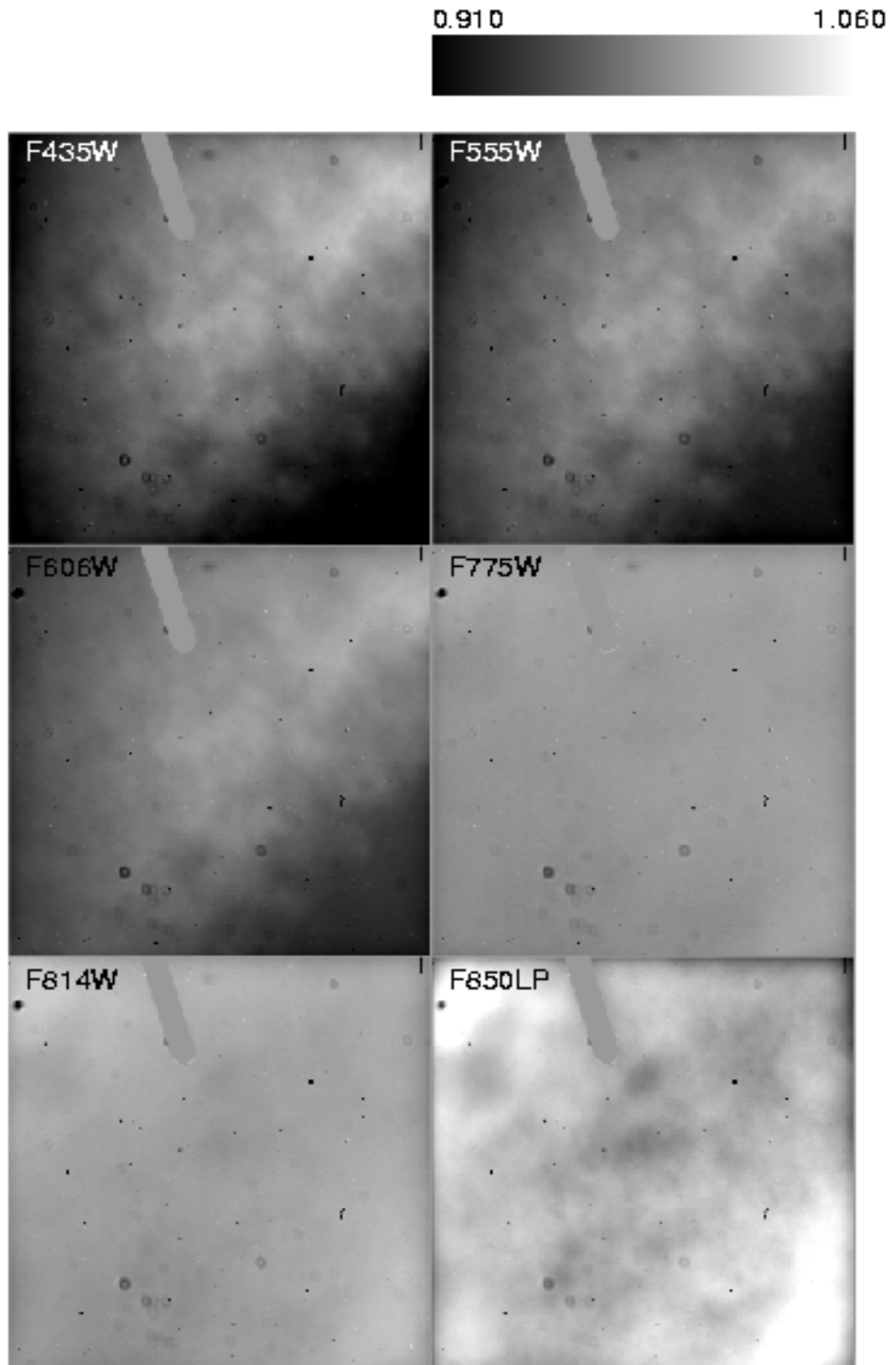


Figure 5.14: HRC Flat Fields



5.4.4 Earth Flat Verification

Because the RAS/HOMS optics are opaque below 3500 Å (see Section 5.4.1), flat fields for the UV filters were created using inflight observations of the bright Earth. The Earth is a poor flat field source at optical wavelengths because structure in the cloud cover can cause streaking in the flat. However, HRC modes that utilize the F220W, F250W, F330W, and F344N are immune to streaks because of the large optical depth down to the tropospheric cloud layers. The bright Earth then provides a uniform flat field source for the complete OTA+HRC optical complement. The required calibration flat, which incorporates both the low frequency L-flat and the high frequency pixel-to-pixel P-flat response, can be easily produced from these observations. Unfortunately, the red leaks in F220W and F250W are so large that the out of band light dominates, and the lab flats made with the deuterium lamp illumination (see [ACS ISR 01-11](#) for details) are superior to the observed earth flats for the modes that include these two filters. Because no L-flat correction has been applied, errors in the flats of +/- 2 to 4% corner-to-corner are expected for these two filters.

The HRC earth flats at wavelengths longward of 4000 Å often show streaking and non-uniformities from clouds or the terminator. However, a number of the images are free from these defects and provide a promising alternative to sky flats as well as an independent verification of the stellar L-flats. A comparison of the current pipeline flats to the new HRC earth flats shows differences of typically less than ~1% for the broadband filters and ~2% for the interpolated L-flat filters. An ISR describing these results is in preparation and will be posted on the ACS site:

<http://www.stsci.edu/hst/acs/documents/isrs>.

At the shortest exposure time of 0.5 sec, most WFC modes saturate, except for a few narrow band filters (i.e., F502N, F658N, F660N, and F892N) which are only saturated approximately half the time. Earth flats in these narrow band filters would nicely compliment the stellar L-flat program and sky flat results, which use only broadband filters. More observations in the WFC are required to produce good enough earth flats for comparison with the pipeline flats.

5.4.5 Sky Flats

The ACS sky flat program takes advantage of the many survey programs that have executed on ACS, such as the GOODS program, to build high signal-to-noise flats. These non-proprietary data consist mostly of sparsely populated images with relatively uniform sky that can be stacked to further quantify the pixel-to-pixel variation of the instrumental response.

The GOODS program consists of either 15 or 16 unique pointings over 5 epochs for both the Chandra Deep Field South (CDFSS) and the Hubble Deep Field North (HDFN) using the most commonly used F606W, F775W

and F850LP filters in the WFC. In total between 50-70 images were stacked in each filter.

The sky flats were created by median combining the pipeline reduced FLT files after removing cosmic rays and masking all of the sources. Because the GOODS data contained 2-4 dithers at each pointing, depending on the filter, masking was necessary to eliminate the high values at each pixel. In addition before combining, each image was corrected for the pedestal bias signature and inspected for scattered light.

The combined sky flat in each filter was box medianed down to a 32x32 image, in order to directly compare it to the corrected ground flat. Since the sky flats were created starting from the pipeline calibrated FLT files, it will show any flat field signatures that were not accounted for by the pipeline. The resulting sky flats show variations across the field of view of <2% for each of the 3 filters. The CDFS and HDFN fields were analyzed separately and compared. The two fields showed excellent agreement (<1%) for all 3 filters. The inferred sky flats will be made available on the ACS web site and a future ISR will describe the sky flats in more detail. Please check the ACS web site for updates on this issue.

5.5 Image Anomalies

5.5.1 Dust Motes

Several circular patterns, consisting of a dark ring with a bright center, are visible in the ACS flat fields, with typical diameters of ~30 pixels on HRC and ~100 pixels on WFC. These artifacts are shadows of dust on the CCD windows and are weaker on the f/25 WFC than on the f/68 HRC. These motes can be seen in the WFC and HRC flats in Figure 5.13 and Figure 5.14. Since the shapes and depths of these motes are almost independent of wavelength, their effects will be removed by the flats to <<1%, unless any of these particulate contaminants move to different positions on the CCD windows. In case of particulate migrations, the internal lamp flats have a lower f-ratio with a wider angular distribution and cannot be used to patch the flat fields, because they wash out the mote shadows. To correct for new motes, patches to the pipeline flats must be made using the original laboratory flats, corrected for the low-frequency flats derived in flight or, for short wavelengths, using observations of the bright earth.

Larger motes are sometimes present due to dust and blemishes on several ACS filters, including F606W, POL0V, and POL60V. Because the filters are located farther from the detector windows in a converging light beam, imperfections on the filters produce an out of focus image at the

detector, where typical mote diameters are ~ 350 - 400 pixels on the WFC and ~ 250 - 300 pixels on the HRC. One of these large motes can be seen on chip 2 of the WFC F606W flat in Figure 5.13.

Until April 2004, the positioning accuracy of the filter wheels has been within \pm one motor step of the nominal position. This delta corresponds to a distance on the detectors of ~ 18 HRC pixels and ~ 20 WFC pixels. Features with sharp transmission gradients at the filter wheels cause a corresponding flat field instability, where errors are 1-3% for a few pixels near the blemishes. If the filter wheel lands in a different place the dust mote will move. Blemish misregistration is an error in the pixel-to-pixel high frequency component of the flat fields and is not related to the low-frequency L-flat correction, which has been applied for all standard and polarizing filters. For details on the L-flat correction, see Section 5.4.2.

This problem was recognized and addressed before launch by a laboratory calibration campaign to obtain flat fields at the nominal position and at plus and minus one step for the F606W+CLEAR on the WFC and for the two POLV filters in combination with the highest priority F475W, F606W, and F775W filters on both HRC and WFC. Since the resolver position uniquely determines the filter wheel step, the ACS pipeline data processing has been enhanced to automatically apply the proper flat for the wheel step position. The keyword FWOFFSET has been added to the ACS image headers to indicate the position of the filter wheel. In April 2004 an update was made to the ACS flight software and the filter wheel is now always positioned at its nominal position. For more information on flat fields for filter wheel offset positions, refer to [ACS ISR 03-11](#).

5.5.2 Optical Ghosts and Scattered Light

The ACS was designed with a requirement that no single straylight feature may contain more than 0.1% of the detected energy in the object producing it. This goal has generally been met, but during the extensive ground and SMOV test programs a few exceptions have been identified (Hartig et al. 2002, Proc SPIE 4854) such as the WFC elliptical haloes and the F660N ghosts. While these ghosts exceed the specified intensity, their origin and characteristics are well defined and they should have minimal impact on the ACS science program. We describe below in more detail some of the optical ghosts relevant for ACS.

- WFC elliptical haloes: These ghosts are caused by reflection from the CCD surface (which lies at an ~ 20 degree angle to the chief ray) up to the detector windows and back to the CCD. They show up as pairs of elliptical annuli, aligned along the negative diagonal of the FOV (see Figure 5.15), and are observed when bright sources are placed on the lower right (D amplifier) quadrant of the WFC detector. The surface brightness of the annuli increases and size decreases with proximity

to the corner. Two pairs of ghosts are seen, produced by reflection from the four window surfaces. The total energy fraction in each ghost may exceed 0.2% of the target signal.

- F660N ghosts: The F660N narrow band filter produces pairs of relatively bright circular annuli stationed near to (but radially outward from) the target image (see Figure 5.16). This is due to reflection from the two surfaces of the second "blocker" substrate back to the many-layer dielectric stack on the first substrate, which in turn reflects at high efficiency at the filter wavelength range. These haloes contain ~2% of the detected target energy and are always ~10 and 20 pixels in diameter.
- Annular ghosts: Large annular ghosts near their parent images are caused by reflection from the detector windows, back to the filters, then returning to the CCD. The fringes arise from interference between the HeNe laser light reflections from the two surfaces of the windows (Figure 5.15). Another type of annular ghost arises from reflections between the inner and outer window surfaces; these are much smaller in diameter, relatively low in intensity (well within the specification) and are displaced radially from the parent image by a small amount.

In general, little can be done about ghosts and blooming in the post-observation data processing phase. Instead, some judicious planning of the actual observations, particularly if bright sources are expected in the field of view, is recommended. For instance, the impact of diffraction spikes (which for ACS lie along X and Y axes) and of CCD blooming (which occurs along the Y direction) due to a bright star, can be reduced by choosing an ORIENT which prevents the source of interest from being connected to the bright star along either of these axes. Alternatively, a suitable ORIENT could move the bright star(s) into the interchip gap or off the field of view altogether. Similarly, the impact of WFC elliptical haloes can be minimized by avoiding a bright star in the quadrant associated with amplifier D.

Figure 5.15: Optical ghosts, diffraction spikes, and blooming. This image was obtained during ground calibration at Ball Aerospace using a HeNe laser (633nm) through F625W. It is a factor of $\sim 50,000$ saturated with 9 highly saturated point sources in and just off of the field of view. Note the diffraction spikes along the X and Y axes of the WFC, and the severe blooming of the charge along the Y axis. Several optical ghosts with different origins and intensities (see text for details) are visible: (a) WFC elliptical haloes show up in the lower left as pairs of elliptical annuli aligned along the negative diagonal of the FOV; (b) Large annular ghosts are seen near their parent images; (c) To the lower right, there exist smaller fainter annular ghosts which tend to be displaced radially from the parent image by a small amount.

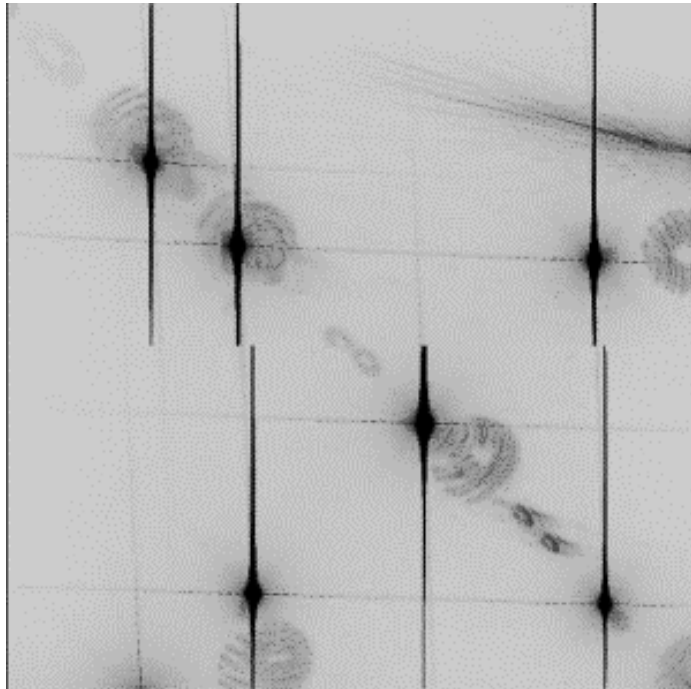
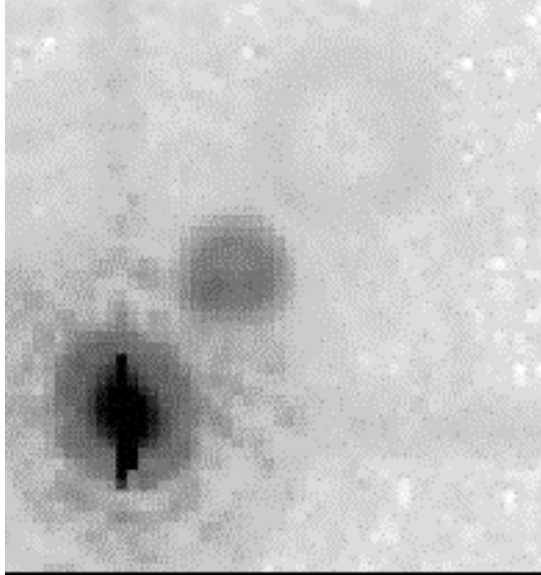


Figure 5.16: F660N optical ghosts. This 700 s F660N exposure illustrates the F660N ghosts, namely pairs of relatively bright circular annuli stationed near to (but radially outward from) the target image. These haloes contain ~2% of the detected target energy and are always ~ 10 and 20 px in diameter.



5.5.3 Cross Talk

Images obtained with the WFC are affected by a small amount of electronic cross talk between the four CCD quadrants that correspond to the four amplifiers of the two detectors. The effect produces electronic "ghost" (i.e., negative) images in a given quadrant that mirror real images recorded on other quadrants. Figure 5.17 shows an example of the cross talk in one of the images obtained during the Great Observatories Origins Deep Survey (GOODS) program, a pointing in the GOODS-S field taken through the F850LP passband. Source images a, b, c and d, which are recorded on quadrant C, produce negative ghost images a', b', c', d', a'', b'', c'', d'', and a''', b''', c''' and d''' in quadrants A, B and D, respectively.

Inspection of images affected by cross talk reveals a number of interesting characteristics of this effect. The cross talk ghosts have a surface brightness that is a few electron lower than those of the average background and there does not appear to be a simple proportionality between the light profile of the sources and the ghosts. Rather, as Figure 5.18 shows, the strength of the cross talk, measured as the average surface brightness of ghosts, seems to be roughly proportional to the logarithm of the sources.

The intensity of the cross talk ghost also appears to vary depending on which quadrant hosts the source and which hosts a ghost. For example, sources in quadrant D of Figure 5.17 produce clearly visible ghosts in quadrant C, but much fainter ghosts (if any at all) in quadrant A and B. Also, sources in quadrant A and B do not seem to produce any obvious ghosts in the other quadrants.

Figure 5.17: Cross Talk in the WFC Detectors. Due to the cross talk, galaxies a, b, c and d recorded in quadrant C have mirror ghost images in quadrants A, B, and D. Note the effect appears stronger in quadrants C and B than in quadrant A, and that sources in quadrant D produce ghosts that follow a similar trend. However, sources in quadrants A and B do not seem as effective in producing ghost images.

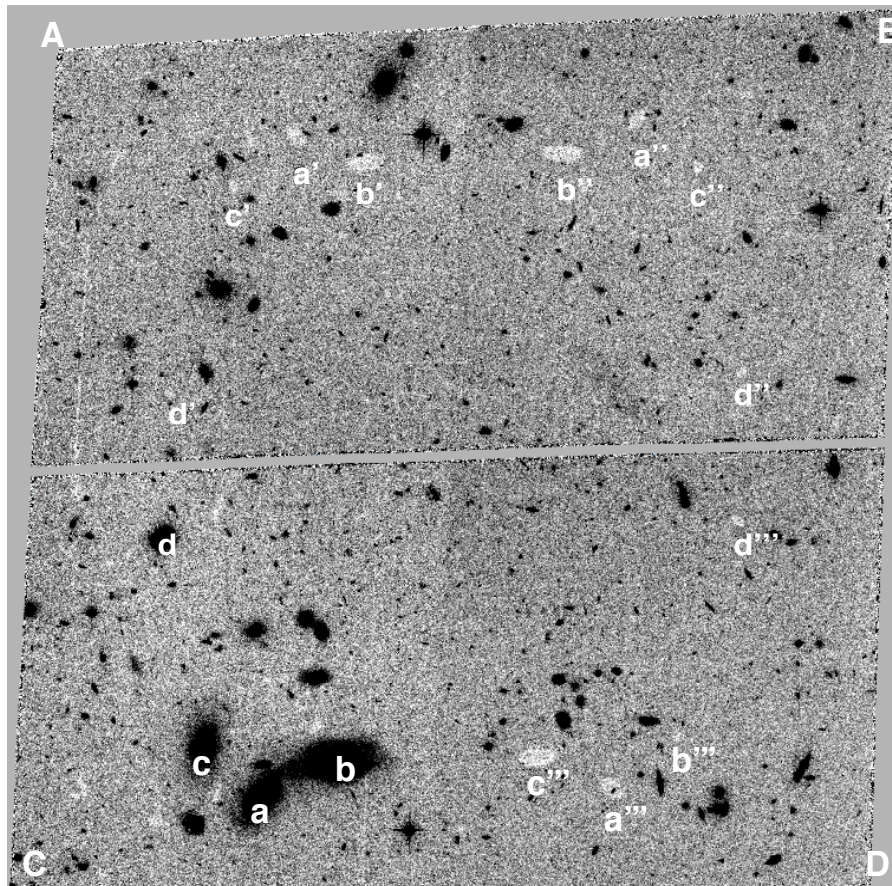
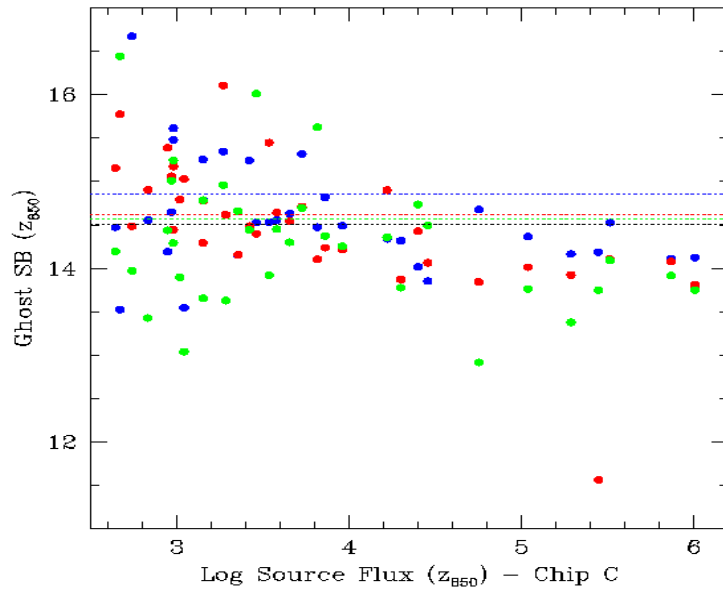


Figure 5.18: The average surface brightness of ghost images plotted as a function of the flux of the sources. All sources are in quadrant C. There is a correlation between the surface brightness of the ghosts and the flux of the sources.



The magnitude of the cross talk depends on the background of the victim quadrant, the sense of the correlation being that the cross talk is stronger when the background is lower. Figure 5.19 shows that in images with low background the cross talk creates ghosts whose surface brightness is $\sim 40\%$ lower than the average background. In images with high background the ghost surface brightness is only $\sim 3\%$ lower.

Tests on existing images show that the cross talk has minimal impact on flux measurements of sources superimposed on ghost images and thus mostly presents a cosmetic issue. The bottom panel of Figure 5.20 shows six growth curves of the same star derived from six individual frames, all with identical exposure time and filter. In three of the frames, due to the adopted dithering strategy, the star is located within a major ghost image (the annulus around the photometric aperture is also within the ghost image), while in the remaining three frames the star is outside the ghost image. The observed photometric scatter is typical for a star with this brightness of post-pipeline processed data (FLT files) with no correction for geometrical distortions and no cosmic ray removal. As a comparison, the upper panel shows the growth curves of a star of comparable apparent magnitude in the same frames but located in an region unaffected by cross talk. The photometry of the latter star has the same amount of scatter as the former one, showing in particular that the cross talk acts as an additive effect and not as a multiplicative one.

Figure 5.19: The dependence of the cross talk, measured as the difference of the sky background in affected and non affected areas, as a function of the sky background. The strength of the cross talk diminishes as the background increases.

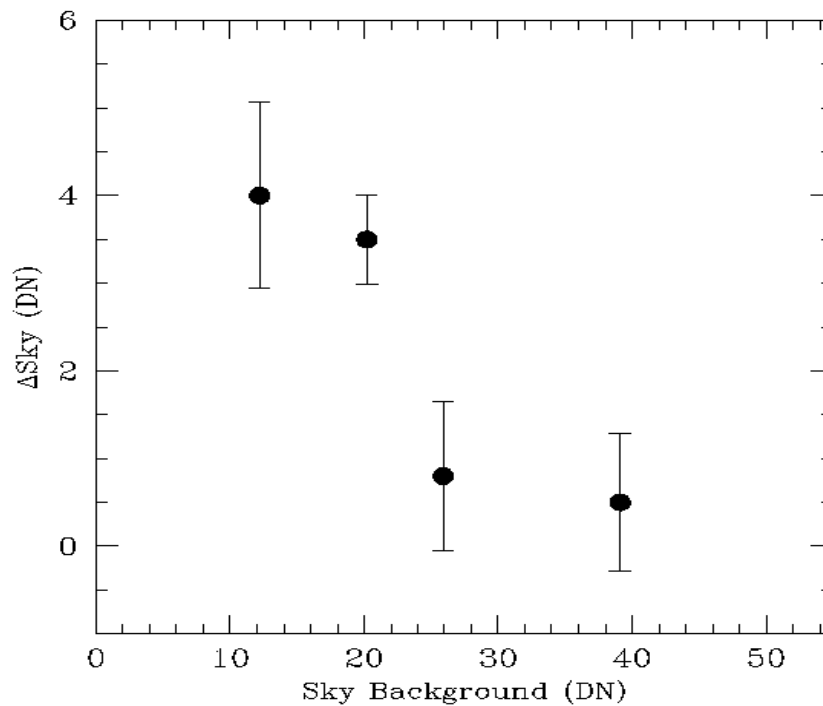
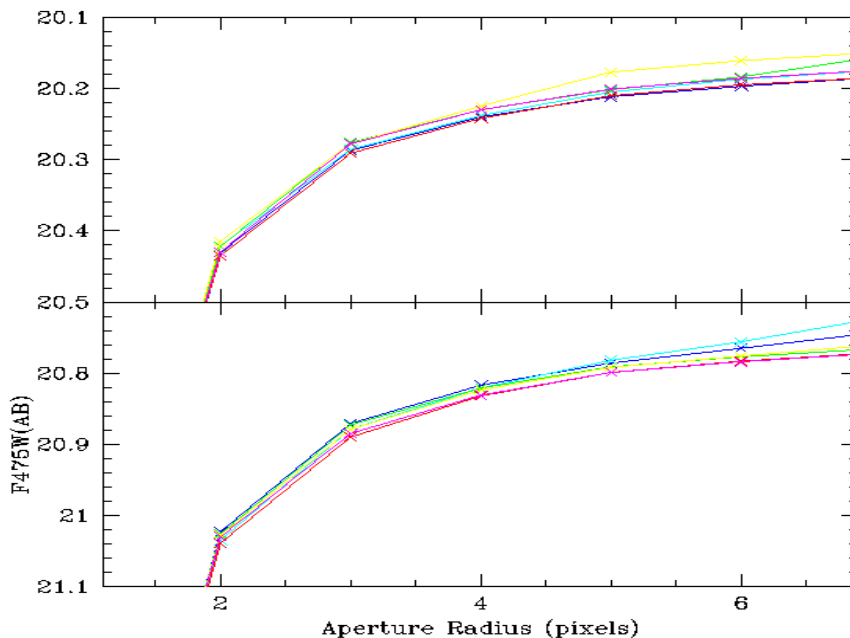


Figure 5.20: BOTTOM - Six growth curves of the same star derived from six individual frames, all with identical exposure time and filter. In three of the frames, due to the adopted dithering strategy for the observations, the star is located within a major ghost image (the annulus around the photometric aperture is also within the ghost image), while in the remaining three frames the star is outside the ghost image. The observed photometric scatter is typical, for a star with this brightness, of post-pipeline processed data (FLT files) with no correction for geometrical distortions and no CR removal. TOP - As above but for a star of comparable apparent magnitude and in the same frame that is located in a region unaffected by cross talk. The photometric scatter of the two stars is very similar.



There is evidence that the strength of the cross talk is significantly weaker in images acquired with GAIN=2, than in images taken with GAIN=1. Figure 5.21 shows two exposures through the same filter and with the same exposure time, the former acquired with GAIN=1 (upper panel), the second with GAIN=2 (lower panel). Cross talk ghosts are clearly observed in the former image, but not in the latter one. A more quantitative comparison supports the visual inspection. Figure 5.22 shows the median value of pixels in ghost images as a function of the source pixel flux, i.e., of pixels located in mirror symmetric locations relative to the pixels in the source quadrant.

Observers who do not need to sample the readout noise of the CCD with the accuracy afforded by the GAIN=1 setting, might want to consider to use GAIN=2 in order to minimize the cross talk.

At this time we have not yet determined a robust way to predict the size of a cross talk residual and the exposure parameters on which it depends. Efforts are ongoing to identify algorithms to remove the effect of cross talk from the WFC images and we encourage GO's to look for updates on the [ACS STScI web page](#).

Figure 5.21: Two exposures through the same filter (F814W) and with the same exposure time (22.5 sec), the former acquired with GAIN=1 (TOP), the second with GAIN=2 (BOTTOM). Cross talk ghost images are observed in the former image, but not in the latter one.

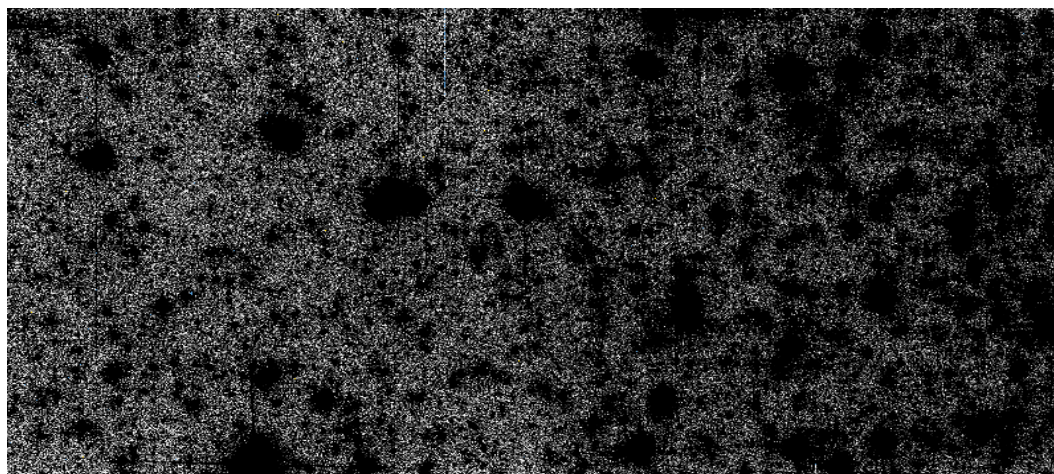
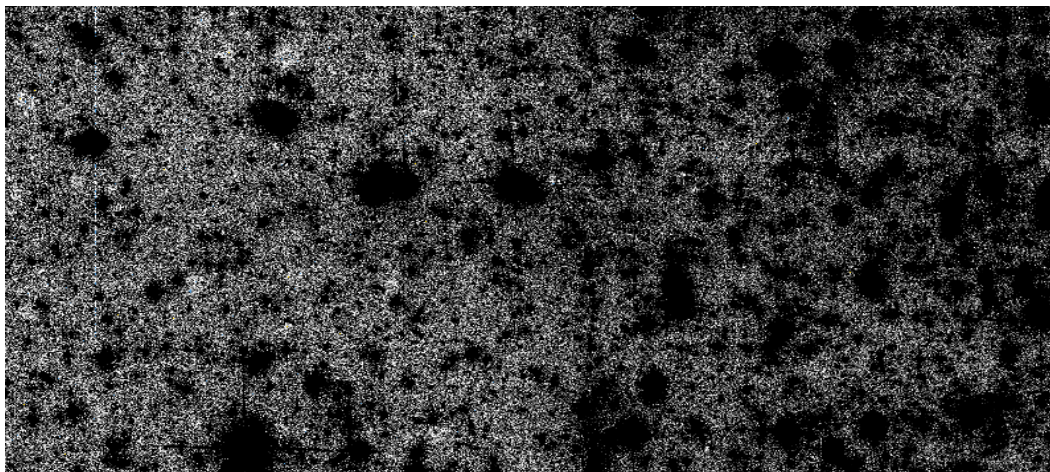
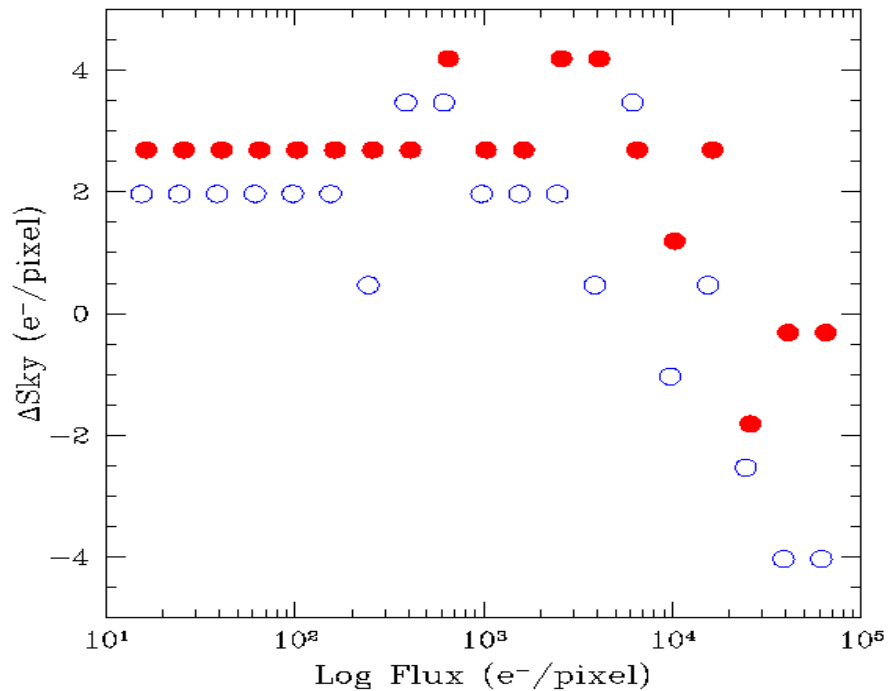


Figure 5.22: The median value of pixels in ghost images as a function of the source pixel flux, i.e. of pixels located in mirror symmetric locations relative to the pixels in the source quadrant, derived from the CCD frames shown in Figure 5.17. The images are sky subtracted. Source pixel flux is binned in logarithmic bins 0.2 in size. Using the median ensures that mostly background pixels, as opposed to pixels affected by sources, are being considered in the analysis. Filled symbols are for GAIN=2, hollow one for GAIN=1. The pixels in the image with GAIN=1 have systematically lower values than those of the image with GAIN=2, an indication that the cross talk is stronger in this case. Their values also decrease as a function of the source pixel value more steeply than in the GAIN=2 case.



5.5.4 Scattered Earth Light

Most observers will not experience significant issues with scattered Earth light in their observations. Normally observations are scheduled only when the bright Earth limb is more than 20 degrees from the HST pointing direction. This is sufficient to eliminate serious impacts from scattered Earth light -- the most severe impact will be for observers with targets in the CVZ who may notice the sky background increased by a factor of 2 or 3.

It is possible to make arrangements for observations at smaller bright Earth limb angles, and these images have a potential for serious impacts from scattered light. There are two types of impact: elevated background and non-uniformity in the background. For example, at a bright Earth

avoidance angle of 14 degrees it is possible for the sky level to be increased by a factor of 100 compared to normal pointings away from the Earth; this will of course have a serious impact on the background noise and detection of faint targets. Also, non-uniformity can arise since the scattered light is taking an increasingly non-standard path through the HST optics, and hence the flat fielding becomes corrupted. At this same angle of 14 degrees, it is possible to have both large scale gradients across the field of view (up to ~20% amplitude in the WFC) and small scale features in the background (up to ~12% in WFC and ~30% in the HRC). See [ACS ISR 2003-05](#) for more details.

5.6 Generic Detector and Camera Properties

5.6.1 Gain Calibration

The ACS CCDs have default gain values of approximately 1 e-/DN (WFC) and 2 e-/DN (HRC) -- these settings have, in particular, been used to establish adjustments to basic quantum efficiency curves. Minor errors in the default gain values would therefore be absorbed into revisions of the QE curves. There is no compelling reason to seek refinement in the quantitative values of the default used for primary calibrations and the bulk of the science program.

The default gains do not sample the full well depths of the CCDs, falling short by about 35% (WFC), and 22% (HRC). Use of the next higher gain values of approximately 2 e-/DN (WFC) and 4 e-/DN (HRC) do provide full sampling of the roughly 85,000 e- and 165,000 e- full well depths, respectively. Since the readout noise is only marginally higher than with the default gains, and is well-sampled even at these higher values (true for WFC, not quite for HRC, but even here its much better than for any WFPC2 data ever taken), many science programs and calibration programs may logically choose these gains.

ACS gain values in use between on-orbit installation and January 2004 can be traced to ground test results. The primary tool for measuring gain values is the photon-transfer method which is described in the Appendix at

http://acs.pha.jhu.edu/instrument/calibration/results/by_instrument/detector/wfc/build4/gain/

This technique relies on analysis of pairs of identical exposure flat fields taken at a range of intensity levels. Over an ensemble of pixels at a given exposure level, the relation between differences of intensity values (noise) and the direct signal level depends on the readout noise and the gain. This relation can be fit at a range of intensity levels to uniquely determine these two quantities. This technique can also provide limited information on

linearity (which depends on the constancy of the lamp source, or experimental techniques to control for drifts) and the saturation count level (which may be different for a uniform illumination pattern than for point sources) by determining where the photon-transfer curve departs from linearity at high signal levels. Errors of about 0.6% in the gain values are quoted for the WFC determinations, and similar values hold for the HRC as given at

http://acs.pha.jhu.edu/instrument/calibration/results/by_instrument/detector/hrc/build1/gain/

On the WFC, with two CCDs and the default use of two readout amplifiers per CCD, errors of 0.6% in the normalization of gains quadrant-to-quadrant would leave easily visible steps in observations of spatially flat sources. A redetermination of the WFC gains ([ACS ISR 02-03](#)), maintaining the same mean over all amplifiers at a given gain setting but using a continuity constraint across quadrant boundaries, provided an improvement in amplifier-to-amplifier gains at better than 0.1% for the default gains. Any steps of intensity seen in data across quadrant boundaries are likely to reflect minor errors in bias and overscan values (see section 5.2), rather than errors in relative gains.

In [ACS ISR 04-01](#) on-orbit data was used to redetermine the mean absolute gain values relative to the standard adopted for the default gains on each camera, e.g., WFC GAIN=1 and HRC GAIN=2 values were retained and other gains adjusted relative to these. Through analysis of observations of the same stellar field, in the same filter and at the two gain settings it is possible to obtain accurate adjustments of these gain values relative to the default gain levels. The results discussed here supply these to better than 0.1%, removing errors that averaged about 1% for the two cameras prior to updates made in January 2004.

Absolute errors of about 0.6% could remain in the default gain values, but these are of no real consequence since redetermination of the CCD quantum efficiency would compensate for this, since the QE adjustments were based on data acquired with the default gains. Basic photometric calibrations apply with equal accuracy to data acquired at all supported gains (assuming for the non-default gains that reprocessing was done after January 6, 2004 to include the revised values).

High Resolution Camera

The prelaunch GAIN = 4 value using the default AMP C is 4.289. The revised value based on on-orbit calibrations is 4.235. For GAIN = 1 (less important since this is available-but-unsupported), the prelaunch value of 1.185 has been revised to 1.163.

Wide Field Camera

Revisions of WFC GAIN = 2 values based on on-orbit testing are for prelaunch (new): AMP A 2.018 (2.002), AMP B 1.960 (1.945), AMP C

2.044 (2.028), and AMP D 2.010 (1.994). And for the available-but-unsupported GAIN = 4 the prelaunch (new) values are: AMP A 4.005 (4.011), AMP B 3.897 (3.902), AMP C 4.068 (4.074), and AMP D 3.990 (3.996).

5.6.2 Full Well Depth

Conceptually, full well depths can be derived by analyzing images of a rich star field taken at two significantly different exposure times, identifying bright but still unsaturated stars in the short exposure image, calculating which stars will saturate in the longer exposure and then simply recording the peak value reached for each star in electrons (using a gain that samples the full well depth, of course). In practice, as discussed in [ACS ISR 04-01](#), it is also necessary to correct for a ~10% "piling up" effect of higher values being reached at extreme levels of over-saturation relative to the value at which saturation and bleeding to neighboring pixels in the column begins.

Since the full well depth may vary over the CCDs, it is desired to have a rich star field observed at a gain that samples the full well depth, and for which a large number of stars saturate. Calibration programs have serendipitously supplied the requisite data of rich fields observed at two different exposure times.

High Resolution Camera

The HRC shows large scale variation of about 20% over the CCD. The smallest full well depth values are at about 155,000 e- and the largest at about 185,000 e-, with 165,000 representing a rough estimate at an area weighted average value. See Figure 3 of [ACS ISR 04-01](#) for details.

Wide Field Camera

As with the HRC there is a real and significant large scale variation of the full well depth on the WFC CCDs. The variation over the WFC CCDs is from about 80,000 e- to 88,000 e- with a typical value of about 84,000 e-. There is a significant offset between the two CCDs. The spatial variation may be seen as Figure 4 in Gilliland [ACS ISR 04-01](#).

5.6.3 Linearity at Low to Moderate Intensity

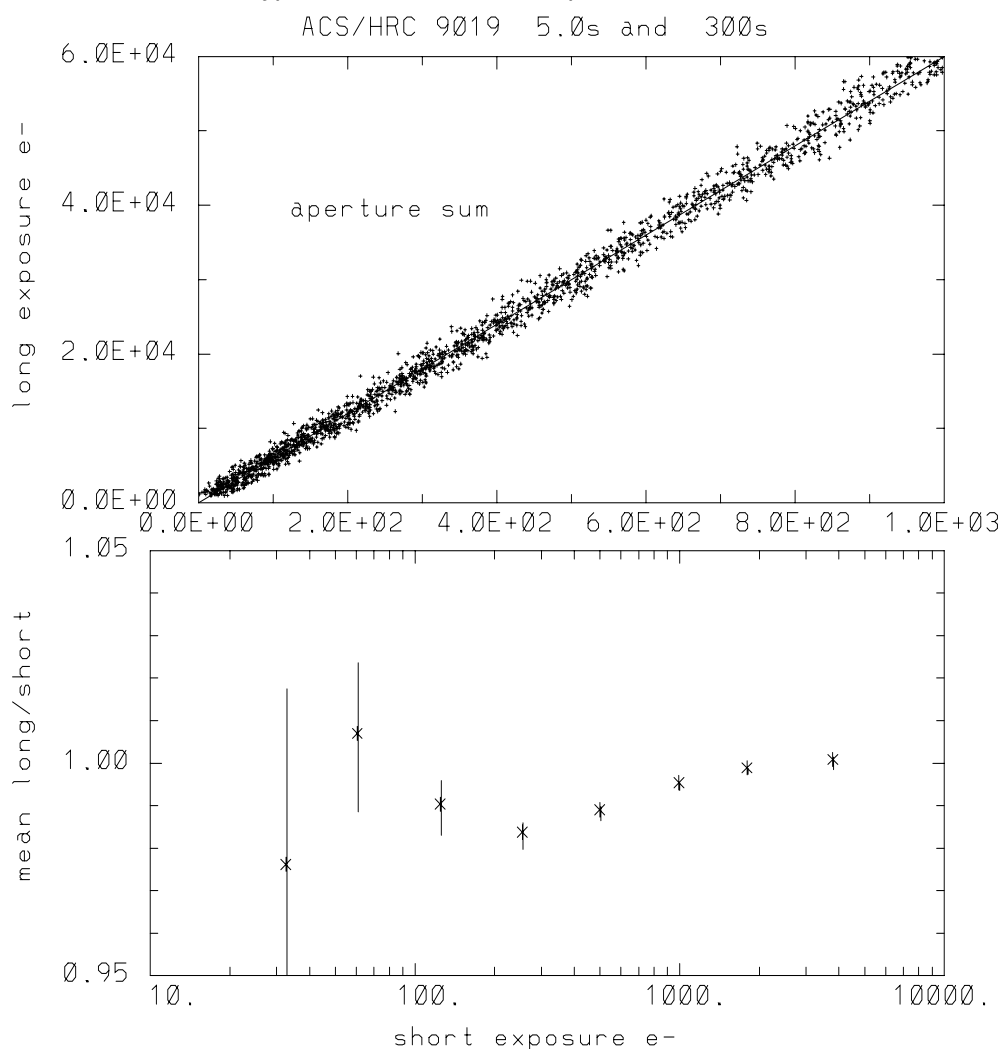
High Resolution Camera

Rich star fields observed at quite different exposure times provide a simple, direct test for linearity. In Figure 5.23 the results of such tests are shown in two ways. The first is a simple plot of aperture sum values in the long exposures versus the same stars on the same pixels in the short exposure -- no deviations from linearity are evident. For a more sensitive

test the bottom plot shows the results of summing counts over all stars within a defined magnitude range in the short and long cases separately, before then taking the ratio and normalizing to the relative exposure times.

The linearity of the HRC at low and moderate intensity levels, as evidenced by comparing stars observed with exposures differing by a factor of 60, appears to be excellent.

Figure 5.23: The upper panel shows aperture sums over 9 pixels for all stars used over a range emphasizing results at low to moderate count levels. The plotted line has a slope set by the relative exposure time. In the lower panel the ratio of counts in ensembles of stars divided into factor of two intensity bins, and further normalized by the relative exposure time are shown. One sigma error bars are derived based on the ensemble signal to noise of the short exposure case. The lowest bin has 81 stars, with typical values of 400 stars per bin above this.

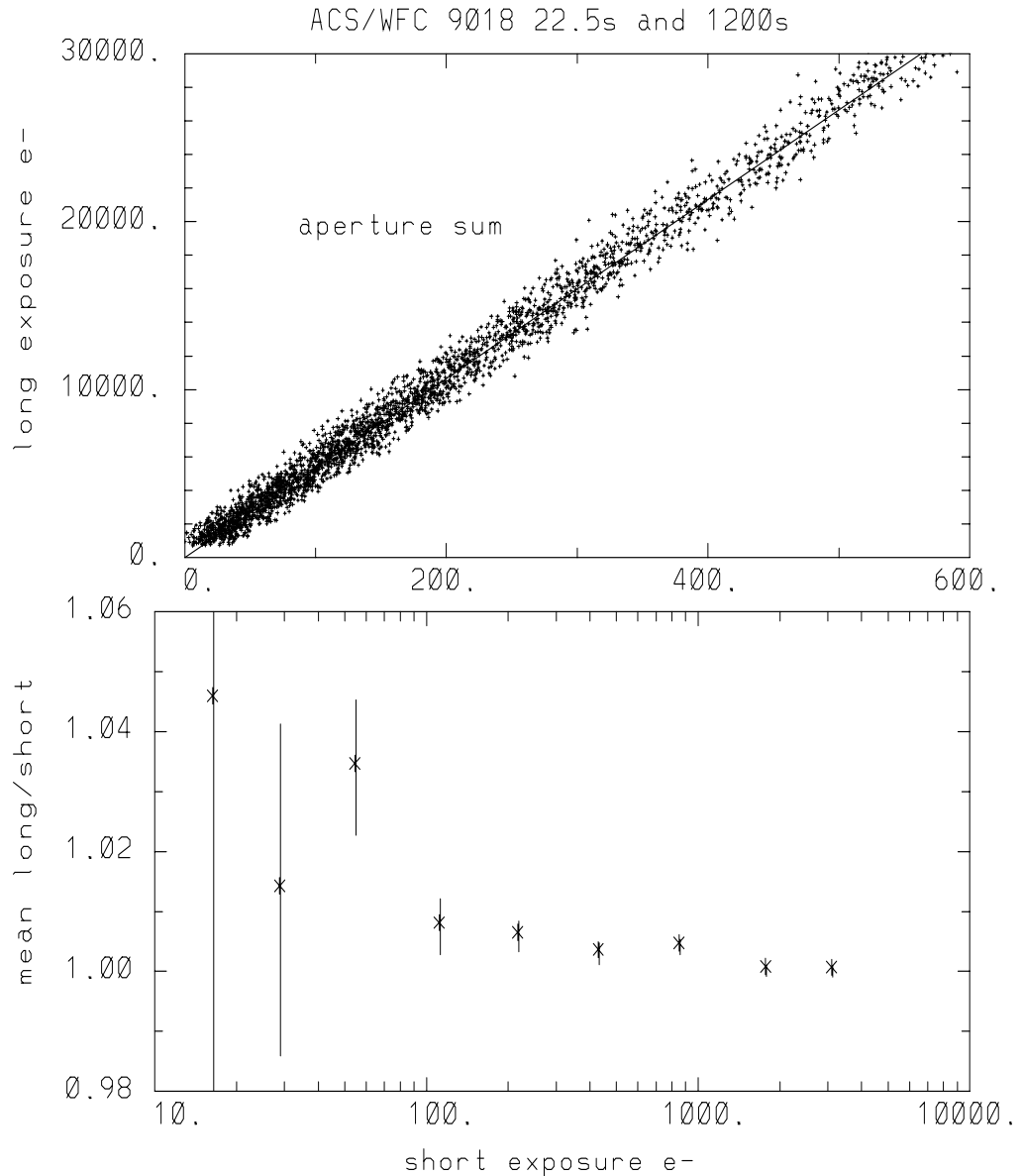


Wide Field Camera

As shown in Figure 5.24 counts within small apertures suggest excellent linearity down to quite low exposure levels of about 5 e- in the central pixel for stellar sources, i.e., down to a level where individual stars

cannot be recognized in single exposures. The data used for this test were taken in April, 2002, only one month after on-orbit installation of ACS. CTE effects are of course a source of low-intensity non-linearity (that correlates with y-positioned detector) and should have been very small at this early date (tests verified this). By contrast, for the approximately one year after launch observations used in ISR 2003-09 (quantifying CTE more generally), the CTE impact (at middle of CCDs) at the three lowest intensity levels in Figure 5.27 are at 24, 16, and 10.3% respectively. Thus, any intrinsic non-linearities that may exist at low intensity levels are small compared to the normal losses expected from CTE growth after ACS had been on orbit for only a few months. Within the mutual error bars results for the GAIN = 2 comparisons are entirely consistent -- the low-level linearity of the WFC channel is obviously excellent.

Figure 5.24: The upper panel shows simple aperture sums for the long and short exposures for all stars of low-to-moderate intensity, the line has a slope set by the relative exposure times. The lower panel shows ratio of counts summed over all stars within intensity bins (in factor of two steps) for the 1200 second to mean 22.5 second exposures after normalization to the relative exposure time. To account for minor encircled energy differences for the very small 9 pixel (total, not radius) apertures used all points have been normalized by 1.006, the initial value for the brightest bin. The error bars show plus/minus one sigma deviations based on the total signal to noise of the short exposure sums. The number of stars per bin is typically about 400, although the lowest bin contains only 33 stars.



5.6.4 Linearity Beyond Saturation

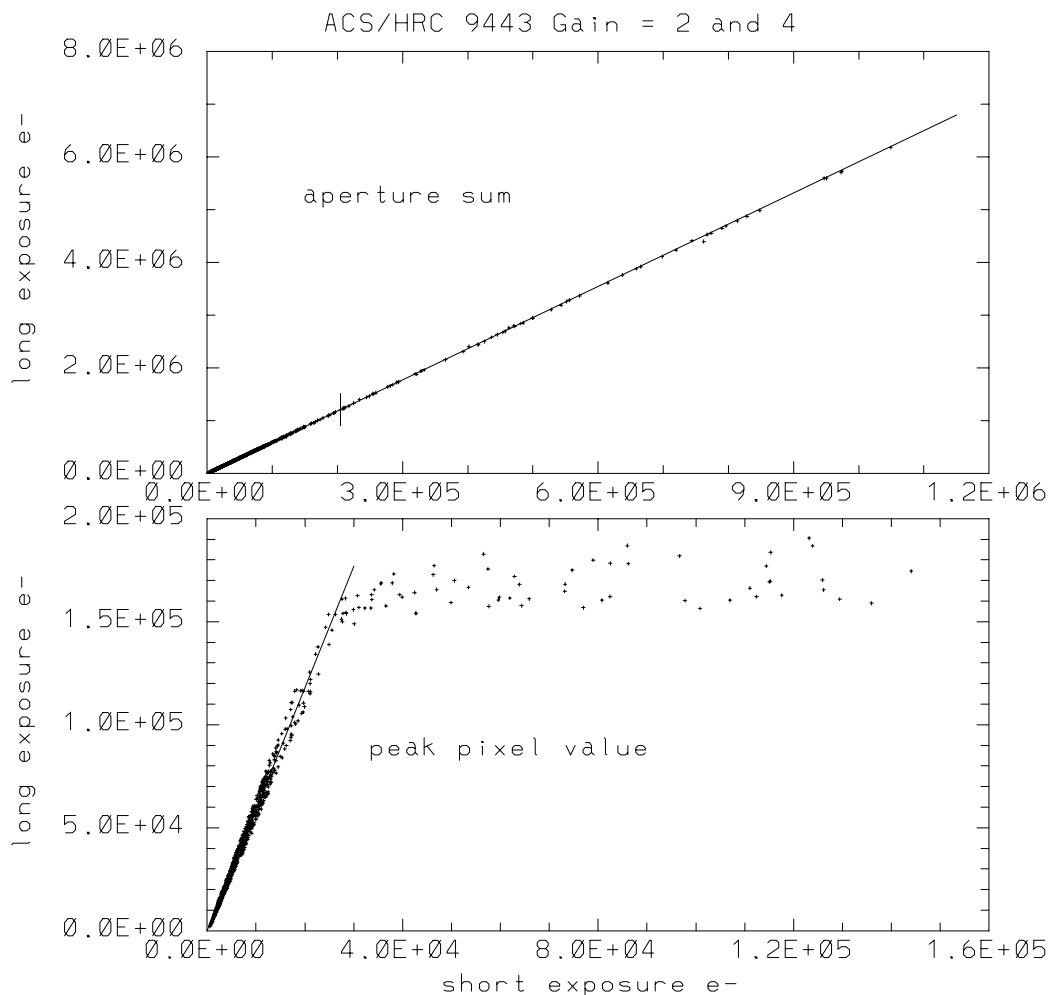
The response of the ACS CCDs remains linear not only up to, but well beyond, the point of saturation when using a GAIN value that samples the full well depth. [ISR 04-01](#) shows the well behaved response of the ACS: electrons are clearly conserved after saturation. This result is similar to that of the STIS CCD (Gilliland et al., 1999, PASP, 111 1009-1020) and the WFPC2 camera (Gilliland, R.L. 1994, ApJ, 435, L63-66). It is possible to easily perform photometry on point sources that remain isolated simply by summing over all of the pixels bled into with a GAIN value that samples the full well depth. Given the extra dynamic range afforded before saturation at GAIN=2 (WFC) and GAIN=4 (HRC), and the only modestly increased readout noise coupled with the potentially beneficial aspect of being able to recover photometry on saturated objects, more frequent use of these GAIN values may be appropriate.

Here the extent to which accurate photometry can be extracted for point sources in which one or more pixels have exceeded the physical full well depth is explored. Only the cases of GAIN = 4 for the HRC and GAIN = 2 for the WFC are considered, which provide direct sampling of the count levels independent of whether saturation has occurred. Ideal data for these tests consist of multiple exposures taken back-to-back on a moderate-to-rich star field with a broad range of exposure times resulting in both unsaturated and saturated data for many stars.

High Resolution Camera

Figure 5.25 illustrates linearity beyond saturation. In the lower panel peak values in the long exposure are plotted against peak values from the short exposure. Over the expected linear domain, points fall within a narrow cone centered on a line that has a slope equal to the exposure time ratio while above this the values in the long exposure saturate as expected. Deviations from lying perfectly along the line here result primarily from a 0.1 pixel offset between the images used for this test leading to different relative fractions of light falling on the central pixel. The upper panel of Figure 5.25 shows the same stars but now using identical extraction apertures in the two exposures, the vertical tick mark near 205,000 e⁻ on the x-axis flags stars below which the central pixel remained unsaturated, above which the central pixel experienced saturation. In the aperture sum data there is no difference in the long exposure photometric quality between stars that saturate, even up to a factor of 5, and those that do not. Within the domain sampled here the accuracy of saturated star photometry is much better than 1%.

Figure 5.25: The lower panel shows peak pixel values for many stars observed for 350 seconds (long) and 60 seconds (short) with the HRC; over the range for which stars are unsaturated in both exposures there is a linear relation with slope equal to exposure time ratio. Brighter stars in the long exposure saturate at values near the full well depth. The upper panel shows the same stars using photometry sums from identical apertures in both exposures, stars above the vertical mark near 205,000 on the x-axis separates stars that are saturated in the long exposure. The photometric response when summing over saturated pixels that have been bled into remains perfectly linear far beyond saturation of the central pixel at GAIN=4.

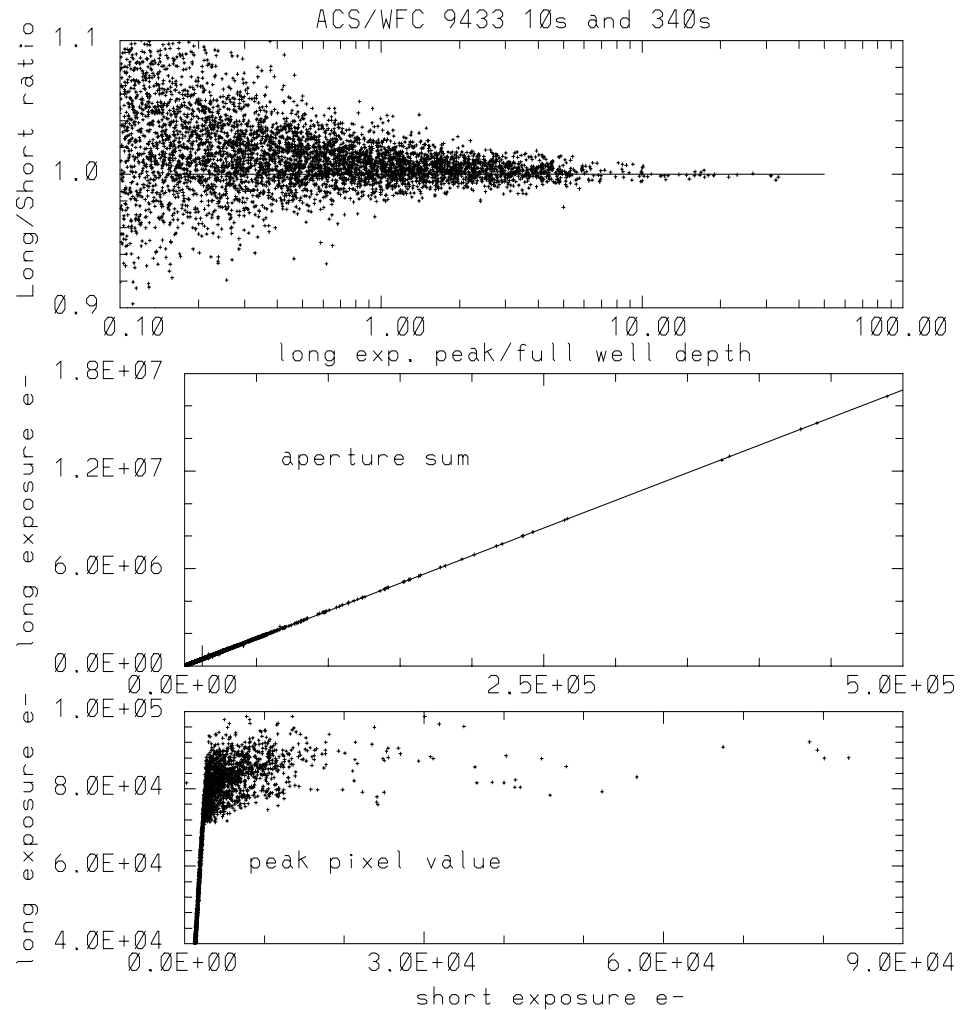


Wide Field Camera

Figure 5.26 shows results for the WFC. Over a range of nearly 4 magnitudes beyond saturation, photometry remains linear to < 1%.

All of the above linearity results are based upon comparisons of FLT images. The conservation of flux property of drizzle leads to equally good results for linearity beyond saturation comparing long and short DRZ images. An analysis of the drizzled data sets corresponding to Figure 5.26 show equally impressive results.

Figure 5.26: The lower panel shows peak pixel values for many stars observed for 340 seconds (long) and 10 seconds (short) with the WFC; over the range for which stars are unsaturated in both exposures there is a linear relation with slope equal to the exposure time ratio. Brighter stars in the long exposure saturate at values near the full well depth. The middle panel shows the same stars using photometry sums from identical apertures in both exposures, stars above the vertical mark near 12,350 on the x-axis separates stars that are saturated in the long exposure. The upper panel shows the ratio of long to short aperture sums normalized by the relative exposure time plotted against the degree of over-saturation in the central pixel of the long exposure. The plotted line in each panel represents perfect linearity, rather than fits to the data. The photometric response when summing over saturated pixels that have been bled into remains perfectly linear far beyond saturation of the central pixel at GAIN=2.



5.6.5 Shutter Stability

The shutter mechanism operation varies for different commanded exposure times. For example, the WFC shutter management varies for the shortest exposures, e.g., at 0.5 sec for WFC it is a continuous motion, 0.6

sec is not allowed and 0.7 sec begins the standard mode of operation. Similar differences exist for the shortest HRC options. Ground-based testing is discussed in Martel, Hartig and Sirianni (2001c) and should be referred to for background on operation of the shutters.

For short exposure times, field dependent timing (shutter shading), A versus B blade shutter control dependence, stability, and timing accuracy were assessed for the HRC and WFC during ground based testing and through utilizing on-orbit data. The ground results using time variable flat-field sources were suggestive of deviations in actual versus nominal exposure times up to, but no larger than ~1% for short exposures of less than 1.0 sec.

For a full discussion of on-orbit data used to assess shutter stability see [ACS ISR 03-03](#) (Gilliland and Hartig 2003).

Even at the shortest exposures, shutter shading measures did not exceed ~0.5% center-to-edge for either camera. This supports a previous decision to not actively invoke a shading correction in **CALACS**.

No systematic biases in timing reach the 0.5% level for any average over multiple pairs at any exposure time when switching between A and B shutter control. This supports not actively utilizing a record of which shutter was used.

Stability of shutter timing is a bit more problematic. For the HRC stability appears excellent, measured as rms across several exposures, the errors remain well under 0.5%. At exposure times of 1.0 sec peak-to-peak fluctuations on the HRC are at only the 0.1% level. Only at 0.1 sec do the HRC exposures exhibit peak-to-peak fluctuations up to ~1.0% (and part of this may be measurement error). The WFC shutter timing stability is not as good, but was also more difficult to quantify given a higher level of systematic errors. Again measured as an rms across several exposures, errors are well under 0.5%. However, out of 7 pairs of back-to-back 1.0 sec exposures, two of the pairs have individual components differing by > 1.3%. Where such differences exist on WFC, it is always the case that shutter timing under A control is shorter than the following B exposure. At 0.5 sec on WFC (where shutter operation is one continuous rotation) the greatest discrepancy A to B remains < 0.3%. These results suggest that if generally short WFC exposures are required (where short is taken to be < 2 sec), then 0.5 sec exposures seem stable and likely to support 1% accuracy, but exposures over the range 0.7 to <2.0 sec may experience timing fluctuations that could compromise such accuracy. This conclusion regarding WFC exposures is not regarded as robust, but is offered as that most consistent with a simple and conservative interpretation of the test data.

Adopting a threshold of 0.5% in absolute timing, only 4 exposure times over HRC and WFC required revisions from pre-launch values. The 0.1 sec HRC exposure is actually larger than this by 4.1% WFC exposures at 0.5 sec, 0.7 sec and 0.8 sec in reality differ by +1.6%, -1.0% and -0.6%

respectively. For these 4 exposures, using revised times in **CALACS** is recommended in order to support accurate photometry; retrievals made after March 11, 2004 will have invoked use of these corrected values.

ACS Data Analysis

In this chapter. . .

6.1 Photometry / 6-1
6.2 Astrometry / 6-23
6.3 Polarimetry / 6-26
6.4 Coronagraphy / 6-32
6.5 Ramp Filters / 6-35
6.6 Spectroscopy with the ACS Grisms and Prisms / 6-37

This chapter describes the data reduction applications available to work with ACS data, and the specific analyses you may want to apply to your imaging, spectroscopy, polarimetry and coronagraphy data.

6.1 Photometry

6.1.1 Photometric Systems, Units, and Zeropoints

We strongly recommend that, whenever practical, ACS photometric results be referred to a system based on its own filters. Transformation to other photometric systems is possible (see Sirianni et al., 2004, PASP submitted) but such transformations have limited precision and strongly depend on the color range, surface gravity and metallicity of the stars considered.

Three magnitude systems, based on ACS filters are commonly used, VEGAMAG, STMAG and ABMAG. They are all based on the knowledge of the throughput of the entire system (OTA + ACS CAMERA + FILTER + DETECTOR). These synthetic systems are tied to the observed standard

stars. VEGAMAG is a standard-based system and therefore is subject to systematic errors tied to the calibration of the standard star. In the last decades it has become more and more common to use photometric systems such as ABMAG (Oke, J.B. 1964, ApJ 140,689) or STMAG (Koorneef, J. et. al. 1986, in Highlights of Astronomy (IAU), Vol.7, ed. J.-P. Swings, 833) which are directly related to physical units, and they are therefore much simpler and cleaner. The choice between standard-based and flux-based systems is mostly a matter of personal preference. Any new determination of the absolute efficiency of the instrument results in revised magnitudes for all three systems.

VEGAMAG

The VEGAMAG system uses Vega (α Lyr) as the standard star. The spectrum of Vega used to define this system is a composite spectrum constructed by assembling empirical and synthetic spectra together (Bohlin & Gilliland, 2004, AJ, in press). The Vega magnitude of a star with flux F :

$$VEGAMAG = -2.5 \log_{10} \left(\frac{F}{F_{vega}} \right)$$

where F_{vega} is the calibrated spectrum of Vega in SYNPHOT. In the VEGAMAG system, by definition, Vega has zero magnitude at all wavelengths.

STMAG and ABMAG

These two similar photometric systems are flux-based. The conversion is chosen so that the magnitude in V corresponds roughly to that in the Johnson system.

In the STMAG system, the flux density is expressed per unit wavelength, while in the ABMAG system, the flux density is expressed per unit frequency. The magnitude definitions are:

- STMAG = -2.5 Log F_{λ} - 21.10
- ABMAG = -2.5 Log F_{ν} - 48.60

where F_{ν} is expressed in $\text{erg cm}^{-2}\text{sec}^{-1}\text{Hz}^{-1}$, and F_{λ} in $\text{erg cm}^{-2}\text{sec}^{-1}\text{\AA}^{-1}$. Another way to express these zeropoints is to say that an object with a constant $F_{\nu} = 3.63 \times 10^{-20} \text{ erg cm}^{-2}\text{sec}^{-1}\text{Hz}^{-1}$ will have magnitude AB=0 in every filter, and an object with $F_{\lambda} = 3.63 \times 10^{-9} \text{ erg cm}^{-2}\text{sec}^{-1}\text{\AA}^{-1}$ will have magnitude ST=0 in every filter.

Zeropoints

By definition the zeropoint of an instrument is the magnitude of an object that produces one count per second. Each zeropoint refers to a count rate measured in a specific aperture. Since for point source photometry it is not practical to measure counts in a very large aperture, counts are instead measured in a small aperture and an aperture correction is applied to transform the result to an "infinite" aperture. For ACS all zeropoints refer to

a nominal "infinite" aperture of radius 5.5 arcsec. Independently from the different definitions, the magnitude in the passband P in any of the ACS systems is given by:

$$\text{ACSmag}(P) = -2.5 \log(\text{total electrons/sec}) + \text{zeropoint}.$$

The choice of the zeropoint determines the magnitude system of ACSmag(P). There are several ways to determine the zeropoints:

- Using SYNPHOT you can renormalize a spectrum to 1 count/sec in the appropriate ACS passband and output the zeropoint in the selected magnitude system (provided that the most updated throughput tables are included in your SYNPHOT distribution). This example renormalizes a 10,000 K blackbody:

```
iraf> calcphot obsmode "rn(bb(10000),band(obsmode),1,counts)" form
```

where `obsmode` is the passband, for example "acs,hrc,f555w", and `form` is the output magnitude system: ABMAG, VEGAMAG or STMAG.

- Using the header keywords. The simplest way to calculate the most updated STMAG and ABMAG zeropoints for your data is to use the photometric keywords in the SCI extension of ACS images:
 - PHOTFLAM is the inverse sensitivity ($\text{erg cm}^{-2}\text{sec}^{-1}\text{\AA}^{-1}$); it represents the flux of a source with constant F_{λ} which produces a count rate of 1 electron per second.
 - PHOTPLAM is the pivot wavelength.

The header keywords PHOTFLAM and PHOTPLAM relate to the STMAG and ABMAG zeropoints through the formulae:

$$\text{STMAG_ZPT} = -2.5 \text{ Log}(\text{PHOTFLAM}) - \text{PHOTZPT} = -2.5 \text{ Log}(\text{PHOTFLAM}) - 21.10$$

$$\text{ABMAG_ZPT} = -2.5 \text{ Log}(\text{PHOTFLAM}) - 21.10 - 5 \text{ Log}(\text{PHOTPLAM}) + 18.6921$$

- Using a list of published zeropoints such as those published on the ACS web page:

<http://www.stsci.edu/hst/acs/analysis/zeropoints>

This web page will be maintained with the most current zeropoints. More information about the zeropoint determinations can be found in Sirianni et al. (2004, PASP submitted) and [ACS ISR 04-08](#).

6.1.2 Aperture and Color Corrections

In order to reduce errors due to the residual flat fielding error and background variations and to increase the signal-to-noise ratio, both of the

most popular photometric techniques, aperture photometry and PSF-fitting photometry, are usually performed by measuring the flux within a small radius around the center of the source. (For a discussion on the optimal aperture size, see Sirianni et al. 2004 PASP submitted). As a consequence, this measurement must be tied to the total count rate by applying an aperture correction.

For historic consistency, ACS zeropoints refer to the total flux in a nominal "infinite" aperture and are ready to use for surface photometry. When performing point source photometry, users should convert their count rates within any given aperture to the count rates in the infinite aperture. In order to facilitate this conversion Sirianni et al. (2004, PASP submitted) provides the correction derived from the encircled energy profiles of an intermediate aperture (0.5" radius) to an infinite aperture. This correction is filter dependent and varies between ~ 0.09 and ~ 0.12 for WFC and between ~ 0.09 and ~ 0.31 for the HRC.

Users should determine the offset between their own photometry and aperture photometry with a 0.5" radius aperture. This is usually done by measuring a few bright stars in an uncrowded region of the field of view and applying the offset to all photometric measurements. If such stars are not available, encircled energies have been tabulated (Sirianni et al. 2004, PASP submitted). However users should be reminded that accurate aperture corrections are a function of time and location on the chip and also depend on the kernel used in Drizzle. They should avoid the blind application of tabulated encircled energies especially at small radii.

The aperture correction for the near-IR filters presents further complications. All ACS CCD detectors suffer from scattered light at long wavelengths. These thinned backside illuminated devices are relatively transparent to near-IR photons. The transmitted long wavelength light illuminates and scatters in the CCD header, a soda glass substrate. It is then reflected back from the header's metallized rear surface and re-illuminates the CCDs front-side photosensitive surface (Sirianni et al. 1998, proc SPIE vol. 3555, 608, ed. S. D'Odorico). The fraction of the integrated light in the scattered light halo increases as a function of wavelength. As a consequence the PSF becomes increasingly broader at increasing wavelengths. WFC CCDs incorporate a special anti-halation aluminum layer between the frontside of the CCD and its glass substrate (Sirianni et al. 1998, proc SPIE vol.3355, 608, ed. S. D'Odorico). While this layer is effective at reducing the IR halo, it appears to give rise to a relatively strong scatter along one of the four diffraction spikes at wavelength $> 9000 \text{ \AA}$ (Hartig et al. 2003, proc SPIE vol.4854, 532, ed. J.C. Blades, H.W. Siegmund), see Section 6.1.4 for more details.

The same mechanism responsible for the variation of the intensity and extension of the halo as a function of the wavelength is also responsible for the variation of the shape of the PSF as a function of color of the source. As a consequence, in the same near-IR filter the PSF for a red star is broader

than the PSF of a blue star. Gilliland et al. (2002, proc SPIE vol.4854, 532, ed. J.C. Blades, H.W. Siegmund.) and Sirianni et al. (2004, PASP submitted) provide assessments of the scientific impact of these PSF artifacts in the red. The presence of the halo has the obvious effect of reducing the signal-to-noise and the limiting magnitude of the camera in the red. It also impacts the photometry in very crowded fields. The effects of the long wavelength halo should also be taken into account when performing morphological studies and performing surface photometry of extended objects (see Sirianni et al., 2004, PASP submitted, for more details).

The aperture correction for red objects should be determined using an isolated same-color star in the field of view, or using the effective wavelength versus aperture correction relation (Sirianni et al. 2004, PASP submitted). If the object's SED is available an estimate of the aperture correction is also possible with SYNPHOT. In fact, a new keyword "aper" has been implemented to call the encircled energy tables in the OBSMODE for ACS. A typical OBSMODE for an aperture of 0.5" would read like "acs,wfc,f850lp,aper\$0.5". A comparison with the infinite aperture magnitude using the standard OBSMODE "acs,wfc,f850lp" would give the estimate of the aperture correction to apply. See the SYNPHOT handbook for more details.

Color Correction

In some cases it may be necessary to compare ACS photometric results with existing datasets in different photometric systems (e.g., WFPC2, SDSS or ground Johnson-Cousins). Since the ACS filters do not have exact counterparts in any other "standard" filter set, the accuracy of these transformations is limited. Moreover if the transformations are applied to objects whose spectral type (e.g., color, metallicity, surface gravity) do not match the spectral type of the calibration observation, serious systematic effects could be introduced. The transformations can be determined by using SYNPHOT, or by using the published transformation coefficients (Sirianni et al., 2004, PASP submitted). In any case users should not expect to preserve the 1-2% accuracy of ACS photometry on the transformed data.

6.1.3 Pixel Area Maps

When ACS images are flat fielded by the CALACS pipeline the resultant FLT files are flat if the original sky intensity was also flat. However, there is still very significant geometric distortion remaining in these images and the pixel area on the sky will vary around the field. Consequently, the relative photometry in the FLT images must be corrected.

One option is to drizzle the data to remove the geometric distortion and allow the sky to remain flat. Therefore both surface and point-source relative photometry are correct in the resulting DRZ files and the

photometric zeropoint in the header will allow conversion from electrons/s to absolute flux units.

Unfortunately users who wish to perform photometry directly on the distorted FLT files, rather than the drizzled (DRZ) data products, will require a field-dependent correction to match their photometry with the DRZ zeropoint.

The correction to the FLT images may be made by multiplying the measured flux on the FLT image by the pixel area at the corresponding position using a pixel area map (PAM), and then dividing by the exposure time (DRZ images have units of electrons/s). The pixel area must be expressed in units of the reference scale (50mas/pixel for WFC and 25mas/pix for HRC). Once this correction has been applied the same zeropoint as applied to the drizzled data products may be used to convert to absolute flux units.

$$\text{DRZ_flux} = \text{FLT_flux} * \text{PAM} / \text{exposure time.}$$

The PAM for the WFC is approximately unity at the center of the WFC2 chip, ~0.95 near the center of the WFC1 chip and ~ 1.12 near the center of the HRC.

Users may construct PAM images for themselves by downloading the script example files as well as the coefficients files which are needed from the web page:

<http://www.stsci.edu/hst/acs/analysis/PAMS>

and executing the scripts in IRAF or PyRAF. Alternatively PAM images may be downloaded directly from the same page, but they are very large.

PAM Concept Illustration






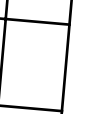



To illustrate the concepts of extended source and point source photometry on FLT and DRZ images we consider a simple idealized example of a 3x3 pixel section of the detector. We assume that the bias and dark corrections are zero and that the quantum efficiency is unity everywhere.

Example #1 Constant Surface Brightness Object

Let's suppose we are observing an extended object with a surface brightness of $2 \text{ e}^-/\text{pixel}$ in the undistorted case. With no geometric distortion the image is:

Actual scene on the sky:	2	2	2
	2	2	2
	2	2	2

In reality ACS suffers from geometric distortion and as a consequence pixels are not square and the pixel area varies across the detector:

Detector pixel layout on the sky:			
			
			

Let's suppose the pixel area map (PAM) is:

PAM:	1.4	1.2	1.0
	1.2	1.0	0.9
	1.0	0.9	0.8

As a result in the raw data there is an apparent variation in surface brightness.

RAW image:

2.8	2.4	2.0
2.4	2.0	1.8
2.0	1.8	1.6

The geometrical area of each pixel is imprinted in the flat field as well as the photometric sensitivity. In this example, since we assumed that the quantum efficiency is unity everywhere, the flat field is just the equivalent of the PAM:

Flat field:

1.4	1.2	1.0
1.2	1.0	0.9
1.0	0.9	0.8

ACS flat fields are designed to level out a uniformly illuminated source and not to conserve total integrated counts, so after the flat field correction the FLT image has the correct surface brightness and can be used to perform surface photometry. However the image morphology is distorted.

FLT image:

2.0	2.0	2.0
2.0	2.0	2.0
2.0	2.0	2.0

PyDrizzle can be run on the FLT image. The result is that each pixel is free of geometric distortion and is photometrically accurate.

DRZ image:	2.0	2.0	2.0
	2.0	2.0	2.0
	2.0	2.0	2.0

Example #2 Integrated photometry of a point source

Now let's suppose we are observing a point source and that all the flux is included in the 3x3 grid. Let the counts distribution be:

Actual scene on the sky:	2.0	10.5	2.0
	10.5	50	10.5
	2.0	10.5	2.0

The total counts are 100. Due to the geometric distortion, the PSF as seen in the raw image is distorted. The total counts are conserved, but they are redistributed on the CCD.

RAW image:	1.9	10	1.9
	10.1	50	10.1
	2.5	11	2.5

After the flat field correction, however, the total counts are no longer conserved:

FLT image:	1.36	8.33	1.9
	8.42	50	11.22
	2.5	12.22	3.13

In this example the counts now add up to 99.08, instead of 100.

In order to perform integrated photometry the pixel area variation need to be taken into account. this can be done by multiplying the FLT image by the PAM or by running **PyDrizzle**.

FLT×PAM	1.9	10	1.9	DRZ image	2.0	10.5	2.0
	10.1	50	10.1		10.5	50	10.5
	2.5	11	2.5		2.0	10.5	2.0

Only running of **PyDrizzle** removes the geometric distortion, but both approaches correctly recover the count total as 100. Users should be cautioned that this is just an idealized example. In reality the PSF of the star extends to a much bigger radius. If the user decides to work on the flat fielded image after correcting by the pixel area map, they need to calculate a new aperture correction to the total flux of the star. The aperture corrections discussed in Section 6.1.2 are only for **PyDrizzle** output images. In most cases the aperture correction for distorted images will be quite different from the same star measured in the DRZ image. This is particularly true for small radius apertures.

6.1.4 PSF

PSF Field Dependence

The point spread functions in the ACS cameras are more stable over the field of view than in any other HST camera, especially when compared to WFPC2. The PSF does vary slightly in both shape and width across the field to a degree that may affect photometric measurements. This effect is described in detail in [ACS ISR 03-06](#). The variations in the HRC are very small and probably negligible when using apertures greater than $r=1.5$ pixels or using PSF fitting. However, the WFC PSF varies enough in shape and width that significant photometric errors may be introduced when using small apertures or fixed-width PSF fitting.

The WFC PSF width variation is mostly due to changes in CCD charge diffusion. Charge diffusion, and thus the resulting image blur, is greater in thicker regions of the detector (the WFC CCD thickness ranges from 12.6 to 17.1 microns, see Figure 6.2). At 500 nm, the PSF FWHM varies by 25% across the field. Because charge diffusion in backside-illuminated devices like the ACS CCDs decreases with wavelength, the blurring and variations in PSF width will increase towards shorter wavelengths. At 500 nm, photometric errors of as much as 15% may result when using small ($r < 1.5$ pixel) apertures. At $r=4$ pixels, the errors are reduced to $<1\%$.

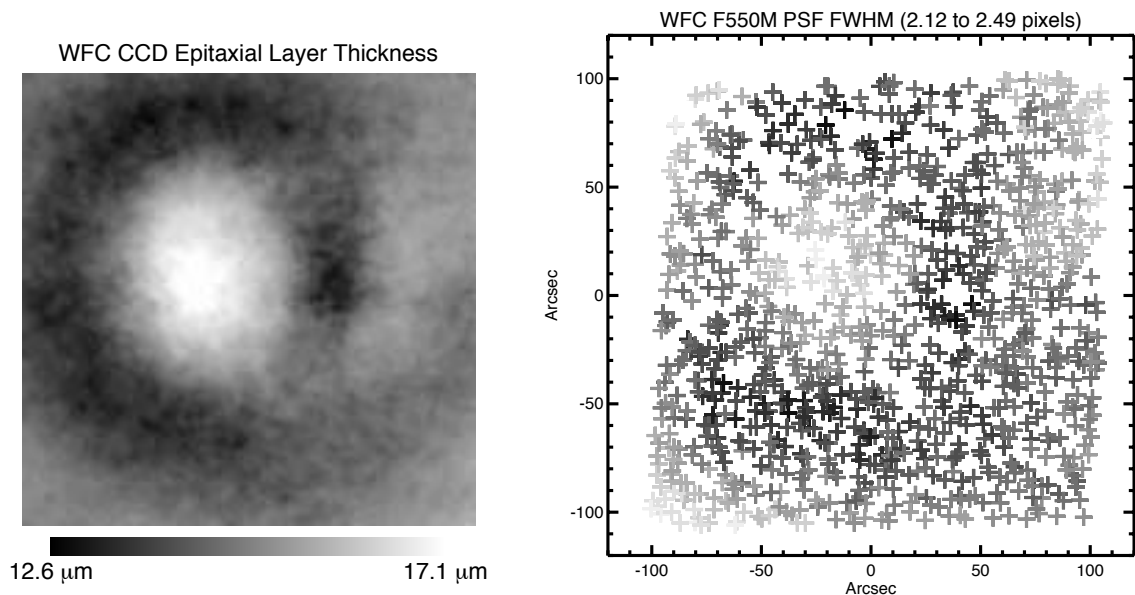
Significant errors may also be introduced when using fixed-width PSF fitting (see [ACS ISR 03-06](#)).

The shape of the PSF also changes over the WFC field due to the combined effects of aberrations like astigmatism, coma, and defocus. Astigmatism noticeably elongates the PSF cores along the edges and in the corners of the field. This may potentially alter ellipticity measurements of the bright, compact cores of small galaxies at the field edges. Coma is largely stable over most of the field and is only significant in the upper left corner, and centroid errors of ~ 0.15 pixels may be expected there.

Observers may use **TinyTim** to predict the variations in the PSF over the field of view for their particular observation. **TinyTim** accounts for wavelength and field-dependent charge diffusion and aberrations, and can be found at:

<http://www.stsci.edu/software/tinytim/>

Figure 6.2: WFC chip thickness (left) and PSF FWHM (right).



PSF Long-Wavelength Artifacts

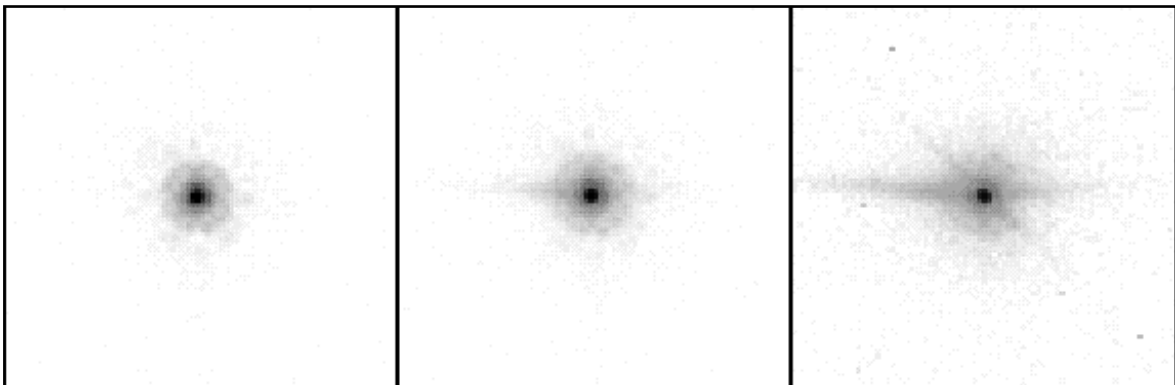
Long wavelength ($\lambda > 700$ nm) photons can pass entirely through a CCD without being detected and enter the substrate on which the detector is mounted. In the case of the ACS CCDs, the photons can be scattered to large distances (many arcseconds) within the soda glass substrate before reentering the CCD and being detected. This creates a large, diffuse halo of light surrounding an object, called the “red halo.” This problem was largely solved in the WFC by applying a metal coating between the CCD and the mounting substrate that reflects photons back into the detector. Except at wavelengths longer than 900 nm (where the metal layer becomes

transparent), the WFC PSF is unaffected by the red halo. The HRC CCD, however, does not have this fix and is significantly impacted by the effect.

The red halo begins to appear in the HRC at around 700 nm. It exponentially decreases in intensity with increasing radius from the source. The halo is featureless but slightly asymmetrical, with more light scattered towards the lower half of the image. By 1000 nm, it accounts for nearly 30% of the light from the source and dominates the wings of the PSF, washing out the diffraction structure. Because of its wavelength dependence, the red halo can result in different PSF light distributions within the same filter for red and blue objects. The red halo complicates photometry in red filters. In broadband filters like F814W and especially F850LP (in the WFC as well as the HRC), aperture corrections will depend on the color of the star, see Section 6.1.2. for more discussion of this. Also, in high-contrast imaging where the PSF of one star is subtracted from another (including coronagraphic imaging), color differences between the objects may lead to a significant residual over-or-under-subtracted halo.

In addition to the halo, two diffraction-spike-like streaks can be seen in both HRC and WFC data beyond 1000 nm (including F850LP). In the WFC, one streak is aligned over the left diffraction spike (right spike in the HRC) while the other is seen above the right spike (below the left spike in the HRC). These seem to be due to scattering by the electrodes on the back sides of the detectors. They are about five times brighter than the diffraction spikes and result in a fractional decrease in encircled energy. They may also produce artifacts in sharp-edged extended sources.

Figure 6.3: CCD scatter at red wavelengths. WFC images of the standard star GD71 through filters F775W (9s, left), F850LP (24s, middle), and FR1016N at 996 nm (600s, right). The CCD scatter, undetected below ~ 800 nm, grows rapidly with longer wavelength. In addition to the asymmetrical, horizontal feature, a weaker diagonal streak also becomes apparent near 1 micron.



HRC and SBC UV PSFs

Below 3500\AA the low and mid-spatial-frequency aberrations in HST result in highly asymmetric PSF cores surrounded by a considerable halo of scattered light extending 1-2 arcseconds from the star. The asymmetries may adversely affect PSF fitting photometry if idealized PSF profiles are

assumed. Also, charge scattering within the SBC MAMA detector creates a prominent halo of light extending about 1" from the star that contains roughly 20% of the light. This washes out most of the diffraction structure in the SBC PSF wings.

6.1.5 CTE

To date, all CCDs flown in the harsh radiation environment of HST suffer degradation of their charge transfer efficiency (CTE). The effect of CTE degradation is to reduce the apparent brightness of sources, requiring the application of photometric corrections to restore measured integrated counts to their "true" value.

The principle of correcting for CTE losses to achieve unbiased measurements is straightforward, but in practice it can be quite difficult to utilize. The charge loss due to imperfect CTE depends on combinations of scene characteristics such as the total counts and physical extent of a source on the detector, its background rate (global and local), the number of parallel and serial transfers, and elapsed time in the life of the detector. It is difficult to quantify and calibrate the degree to which each of these factors affects the measurements of a source over time. The general solution has been to use repeated observations of cluster fields to quantify the impact of CTE on point sources, to use internal measures of CTE degradation to track changes in between costly, external calibrations and, in the future, to do additional studies of the impact to more complicated sources (e.g., extended sources) when the CTE losses become large.

ACS has two CCDs, WFC and HRC, which clock charge and are therefore subject to CTE losses. Here we describe the external calibration of their photometric losses due to imperfect CTE. All CTE calibrations were derived from matched sets of observations of 47 Tucanae using a wide range of position, filter, and exposure time. All CTE measurements were performed by comparing the brightness of individual stars as a function of pixel transfers. The present discussion summarizes the results presented in [ACS ISR 03-09](#) and [ACS ISR 04-06](#).

Figure 6.4 gives a clear and consistent snapshot of the impact of imperfect CTE on photometry in early 2003 for WFC. The statistics are sufficient to detect non-trivial CTE losses from individual bins (~a few hundred stars) at the ~5-10 sigma confidence level for faint stars.

Figure 6.4: An example of the linear relationship determined for WFC for one filter and exposure time (F606W for 30 sec providing a sky level of $\langle s \rangle = 3.6$ electrons) at one brightness range (stars with an average of $\langle f \rangle \sim 400$ electrons and a full range of ± 1 mag) contained within a $r=3$ pixel aperture. The upper panel shows a parallel transfer relation, the lower panel illustrates the serial transfer dependence.

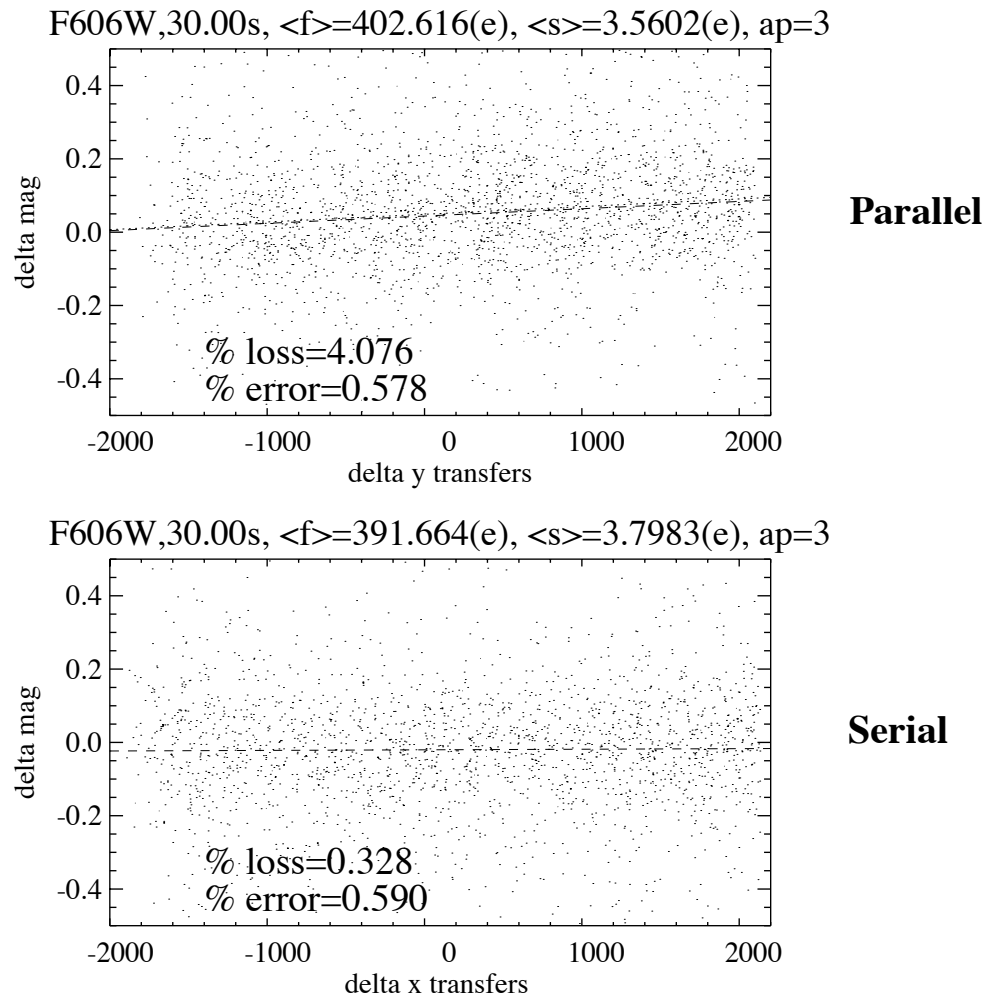
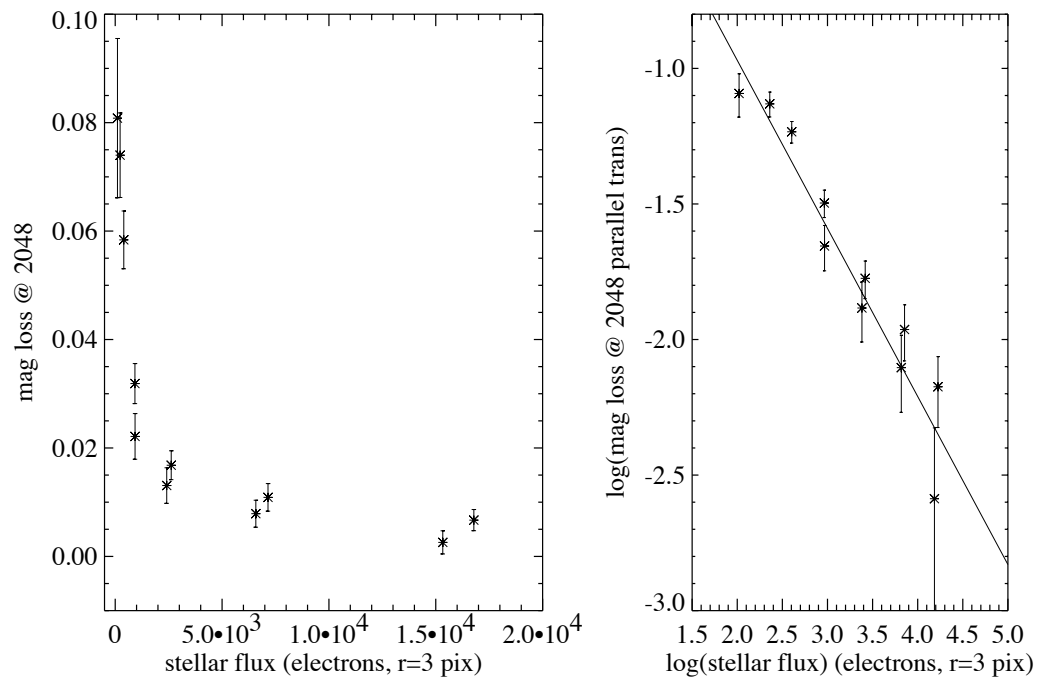


Figure 6.5: The dependence of parallel transfer CTE loss for WFC (over 2048 transfers) versus stellar flux contained within an $r=3$ pixel aperture (all images at low sky level and taken in March 2003). As shown, the relation is strong and suggests a power-law relation which was utilized in the correction formulae. The panel on the right shows the same in log-log space.

Low Background (2-4 e) Parallel CTE

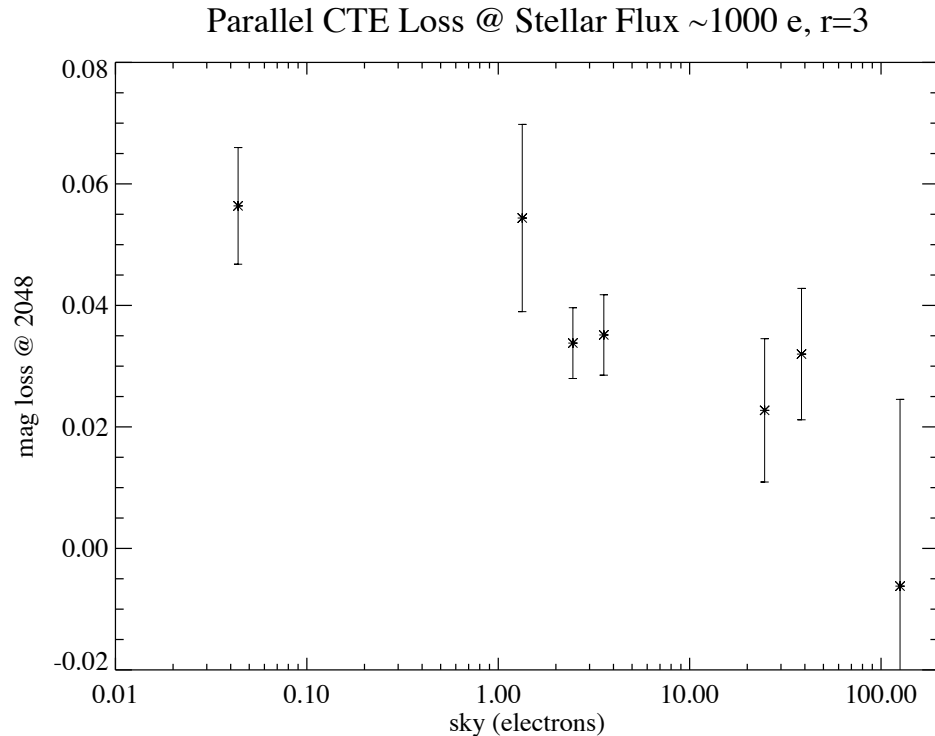


WFC Parallel and Serial CTE

As seen in Figure 6.5, the parallel WFC CTE loss appears to be a strong function of the stellar flux. This is not surprising given similar experience with WFPC2 and STIS. Charge traps may be present at varying depths of the silicon substrate of WFC. Larger charge packets (from brighter stars) may be subject to greater absolute flux losses by accessing deeper traps, but smaller charge packets (from fainter stars) appear to lose a larger fraction of their flux. As seen in Figure 6.5, the data suggests a power-law relationship between CTE loss and stellar flux which we have utilized in the following correction formula.

In Figure 6.6 we show the relationship between CTE loss and background. Increased background can mitigate the CTE losses (presumably by filling traps in advance of the arrival of the stellar charge packet during read-out), and this phenomenon has been seen for WFPC2 and STIS. For the dependence on background we also use a power-law in our correction formula.

Figure 6.6: The relationship between parallel transfer CTE loss and sky background for WFC in March 2003. As seen, the correlation is weak (and is insignificant for apertures of $r=5$ and 7) and in the sense of reduced loss with higher background. The measurement was made for stars with a flux of ~ 1000 electrons to remove the dependence on stellar flux.



Our CTE parameterization is:

$$Y_{CTE} = 10^A \times SKY^B \times FLUX^C \times \frac{Y}{2048} \times \frac{(MJD - 52333)}{365}$$

The term, $(Y/2048)$ reflects the linear relationship of the CTE loss with pixel transfer as seen in Figure 6.4. The last term involves the modified Julian date (MJD) from the header and reflects a linear degradation with time. The full range of sky values (SKY) sampled extended from ~ 0.1 e^- (F502N for 30 sec) to 125 e^- (F606W for 1100 sec). The full range of aperture flux values (FLUX) sampled ranged from ~ 100 e^- to 300,000 e^- .

Table 6.1 contains the best fit values of the parameters A, B, and C and their uncertainties for the equation given above, for 3 different aperture sizes. The parameters were determined by simultaneously fitting data obtained in March 2003, August 2003 and February 2004. The fit to the data is shown in Figure 6.7. To determine the CTE correction (Y_{CTE}) expressed in magnitudes for a particular image observers should:

- select the appropriate aperture size,

- measure the flux in the aperture and the sky value (if working with drizzled data in e^-/sec one must multiply by the exposure time),
- determine the number of parallel transfers, Y ($Y=y$ coordinate for $1 < y < 2048$, $Y=4096-y$ for $y > 2048$),
- determine the MJD,
- and evaluate the correction.

The correction must be derived for the individual exposures, not for a “stack”. Figure 6.7 shows the corrections derived from the formula for different observing parameters.

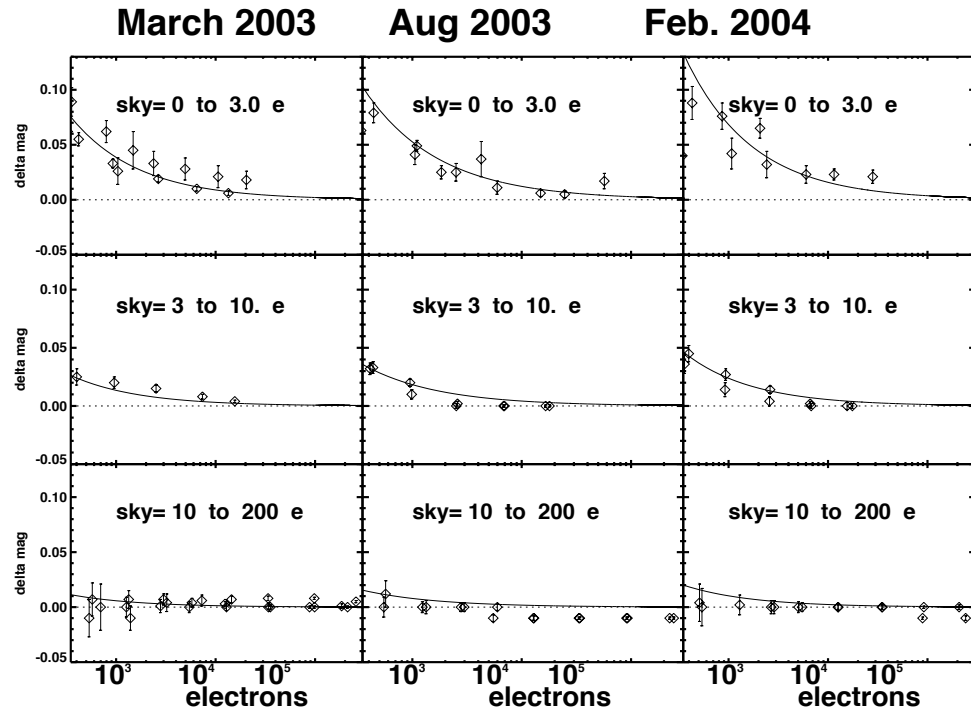
Table 6.1: CTE Correction Coefficients and Uncertainties for WFC Parallel Transfer.

aperture radius (pixels)	$A(\sigma)$	$B(\sigma)$	$C(\sigma)$
3	0.14(0.14)	-0.31(0.02)	-0.64(0.05)
5	-0.55(0.15)	-0.32(0.03)	-0.40(0.04)
7	-0.70(0.24)	-0.34(0.04)	-0.36(0.07)

In principle, first fitting the CTE trends in bins of stellar flux (and individual sky levels), followed by the step of constraining the multi-parameter fitting formula could result in information loss. However, the combination of contaminants (cosmic rays) and the large range of photon statistics (requiring different sigmas for sigma-clipping) dictated the adopted approach.

As of 2004 we have seen no evidence of photometric losses due to imperfect serial CTE on the WFC.

Figure 6.7: Parallel CTE trends. The rows and columns show the photometric losses (at $y=2048$) for different sky levels and dates, respectively for aperture radii of 3 pixels. The fitted line uses the global, time-dependent correction formula in “WFC Parallel and Serial CTE” with the appropriate stellar flux and sky levels.



HRC Parallel and Serial CTE

The same procedure described above for WFC was utilized to analyze HRC data and to quantify the photometric losses due to imperfect CTE. By utilizing all 4 read-out amplifiers, we obtained 2 independent measures of the CTE loss trend for all combinations of filter and exposure time and stellar flux.

Overall, the quantification of photometric losses due to imperfect CTE on HRC is of much lower accuracy than for WFC due to the small field of view of HRC (45 times less area) and the resulting paucity of stars. The adopted parallel CTE correction parameterization in magnitudes is:

$$Y_{CTE} = 10^A \times SKY^B \times FLUX^C \times \frac{Y}{1024} \times \frac{(MJD - 52333)}{(52714 - 52333)}$$

The terms in the equation are the same as for the WFC equation except for the modified Julian date (MJD) term, which characterizes the time evolution of the degradation of CTE away from the calibration date (MJD=52714). Table 6.2 contains the best fit values of the parameters A, B and C and their uncertainties for the Y_{CTE} equation for HRC. At present, our calibration of CTE on HRC is based on a single epoch (March 2003).

The method to correct HRC data is the same as for WFC. Observers should:

- select the appropriate aperture size,
- measure the flux in the aperture and the sky value (if working with drizzled data in electrons/sec one must multiply by the exposure time),
- determine the number of parallel transfers, Y ($Y=y$ coordinate for read-outs with amps C or D, $Y=1024-y$ for read-outs with amps A or B),
- determine the MJD,
- and solve for the correction.

For most observations the default amplifier, C, is used but users can determine which amplifier was used by consulting the header keyword CCDAMP. Similar to the case for WFC, at present we see no significant photometric loss in HRC due to serial transfers.

Table 6.2: CTE Correction Coefficients and Uncertainties for HRC Parallel Transfer.

aperture (pixels)	A(σ)	B(σ)	C(σ)
3	-0.89(0.26)	-0.24(0.13)	-0.21(0.07)
5	-0.87(0.36)	-0.27(0.17)	-0.21(0.09)
7	-2.22(1.10)	-0.58(0.46)	0.12(0.28)

Similar to the case for WFC, at present we see no significant photometric loss in HRC due to serial transfers.

CTE Trending and Evolution

The YCTE correction formulae include a linear time-dependence term which is bounded by the condition that the photometric losses due to imperfect CTE were zero at launch (MJD=52333). While future data will help refine the time-dependence, we believe the linear time-dependence is well motivated by other, on-orbit data such as internal CTE measurements (see “Internal CTE”) and measurements of cosmic ray charge deferred tails.

The Relationship between field-dependent Charge Diffusion and CTE

The CCDs in HRC and WFC are both thinned and the thinning process inevitably results in large-scale variations in the CCD thickness. Because the charge diffusion of the CCD depends on thickness (thicker regions suffer greater diffusion), the breadth of the PSF and aperture corrections can also exhibit a field-dependence (see Section 6.1.4.). [ACS](#)

ISR 03-06 has characterized the spatial variation of charge diffusion as well as its impact on fixed-aperture photometry. For intermediate and large apertures ($r > 4$ pixels), the spatial variation of photometry is very small (less than 1%), but becomes significant for small apertures ($r < 3$ pixels).

It may be of interest to decouple the impact of imperfect CTE and charge diffusion on the field dependence of photometry but it is not necessary for observers to consider these two effects separately. The correction formulae given in Section 6.1.5 include both effects and hence corrects photometry for both effects. This is true as long as the user seeks to correct photometry to a perfect CTE and fiducial PSF width standard. An aperture correction from the fiducial aperture to a nominal "infinite" aperture is still necessary (see Section 6.1.2).

Internal CTE

Internal CTE measurements have been obtained since before launch for both the WFC and HRC. Extended Pixel Edge Response (EPER) and First Pixel Response (FPR) data is routinely collected to monitor the relative degradation of the CTE over time. These data indicate a linear degradation but are otherwise not directly useful for the scientific calibration of ACS data. These results will be published in an ISR that is presently in preparation and will be posted on the web site:

<http://www.stsci.edu/hst/acs/documents/isrs/>.

6.1.6 Red Leak

HRC

When designing a UV filter, a high suppression of off-band transmission, particularly in the red, has to be traded with overall in-band transmission. The very high blue quantum efficiency of the HRC compared to WFPC2 makes it possible to obtain an overall red leak suppression comparable to that of the WFPC2 while using much higher transmission filters. The ratio of in-band versus total flux, determined with SYNPHOT, is given in Table 6.3 for the UV and blue HRC filters, where the cutoff point between in-band and out-of-band flux is defined as the filter's 1% transmission points. The same ratio is also listed for the equivalent filters in WFPC2. Clearly, red leaks are not a problem for F330W, F435W, and F475W and are more important for F250W and F220W. In particular, accurate UV photometry of objects with the spectrum of an M star will require correction for the red leak in F250W and will be essentially impossible in F220W. For the latter filter a red leak correction will also be necessary for K and G types.

Table 6.3: In-band Flux as a Percentage of the Total Flux.

Stellar Type	WFPC2 F218W	HRC F220W	WFPC2 F255W	HRC F250W	WFPC2 F300W	HRC F330W	WFPC2 F439W	HRC F435W	WFPC2 F450W	HRC F475W
O5V	99.8	99.8	99.6	99.7	99.9	99.9	99.9	99.9	99.9	99.9
B1V	99.7	99.7	99.6	99.7	99.9	99.9	99.9	99.9	99.9	99.9
A1V	99.4	99.1	99.2	99.3	99.2	99.9	99.9	99.9	99.9	99.9
F0V	98.5	97.8	98.8	99.0	98.8	99.9	99.9	99.9	99.9	99.9
G2V	92.5	90.2	97.4	98.4	97.4	99.9	99.9	99.9	99.8	99.9
K0V	71.7	69.6	95.0	97.3	95.0	99.9	99.9	99.9	99.8	99.9
M2V	0.03	2.5	45.5	71.9	45.4	99.9	99.9	99.9	99.6	99.9

SBC

The visible light rejection of the SBC is excellent, but users should be aware that stars of solar type or later will have a significant fraction of the detected flux coming from outside the nominal bandpass of the detector. Details are given below in Table 6.4.

Table 6.4: Visible-Light Rejection of the SBC F115LP Imaging Mode.

Stellar Type	Percentage of all Detected Photons which have $\lambda < 1800\text{\AA}$	Percentage of all Detected Photons which have $\lambda < 3000\text{\AA}$
O5V	99.5	100
B1V	99.4	100
A0V	98.1	100
G0V	72.7	99.8
K0V	35.1	94.4

6.1.7 UV Contamination

In an ongoing calibration effort we are regularly observing the star cluster NGC 6881 to monitor the UV performance of the HRC and SBC detectors in ACS. Our results indicate that ACS does not suffer from UV contamination (see [ACS ISR 04-05](#) for details). Over time both the HRC and the SBC have a constant sensitivity within 1% (except the SBC filter F165LP which is within 2%). The three HRC filters F220W, F250W and F330W, do not change sensitivity by more than 1%. Presently, for the SBC the only conclusion is that any change in sensitivity is less than 2% because the precision of the current analysis is limited by the lack of an accurate L-flat field for the SBC.

6.2 Astrometry

6.2.1 Coordinate Transformations

There are three coordinate systems applicable to ACS images. First, there is the position of a pixel on the geometrically distorted raw image (RAW) or, identically, the position on the flat fielded images (FLT) after pipeline processing through **CALACS**. Second, there is the pixel position on the drizzled images (DRZ) created by **PyDrizzle** which corresponds to an undistorted pixel position on a tangent plane projection of the sky. Third, there is the corresponding position (RA, Dec) on the sky.

A position measured on a drizzled output image (DRZ) from **PyDrizzle** or **MultiDrizzle** may be transformed to a position on the celestial sphere using the utility **xy2rd** in the **stsdas.toolbox.imgtools** package within **STSDAS**. There is also a corresponding **rd2xy** task to go from the sky to the pixel position on a drizzled frame. These tools cannot be used for RAW/FLT files as they do not include the very large effects of geometric distortion.

The transformation between the pixel position on a distorted (RAW or FLT) and drizzled (DRZ) image may be performed using utilities in the **dither** package in **STSDAS**. The tasks **traxy** and **tranback** implement the same geometric mapping as **drizzle** and **blot** except applied to X,Y pixel positions rather than images. For more details on how to use these the IRAF help files should be consulted. As a more convenient high-level wrapper the **tran** task, which will be released as part of the late-2004 version of **STSDAS**, allows the mapping of X,Y positions between FLT and DRZ images. This task uses information in the image headers to define the necessary geometrical transformation and requires that the coefficients files are also available. These files are not provided by the **STSDAS** archive and must be recreated by running **PyDrizzle** on the FLT file on the local machine. Please note that to run **PyDrizzle** you also need to have access to the **IDCTAB**.

For example, if an object was found to be at (123,234) on an FLT image (`test_flt.fits[sci,1]`) the position on the drizzled DRZ product (`test_drz.fits`) can be found as follows:

```
pyraf> tran test_flt.fits[sci,1] test_drz.fits[1] forward 123 234
Running Tran Version 0.11 (May 2004)
-Reading drizzle keywords from header...
-Image test_flt.fits[sci,1] was # 1
-to be drizzled onto output test_drz.fits[1]
-Transforming position...
Xin,Yin: 123.00000 234.00000 Xout,Yout: 128.20980 239.63962
PAM: 0.96218
```

The last number refers to the pixel-area map value at this point (see Section 6.1.3).

The reverse operation, DRZ to FLT, can be applied as follows:

```
pyraf> tran test_flt.fits[sci,1] test_drz.fits[1] backward
      128.20980 239.63962
Running Tran Version 0.11 (May 2004)
-Reading drizzle keywords from header...
-Image test_flt.fits[sci,1] was # 1
-to be drizzled onto output test_drz.fits[1]
-Transforming position...
Xin,Yin: 123.00000 234.00000 Xout,Yout: 128.20980
239.63962
```

Finally the **xytosky** task, which forms part of the current dither package, will convert a pixel position on a distorted FLT file directly to a sky position, by applying the distortion correction from the IDCTAB and using the world coordinate information from the header. It may be used as follows:

```
pyraf> xytosky test_flt.fits[sci,1] x=123 y=234 linear=no
- IDCTAB: Distortion model from row 22 for chip 2 : CLEAR and F814W
RA (deg.) = 5.53751673945 , Dec (deg.)= -72.0652050903
```

This task doesn't require the coefficient files but, like **PyDrizzle**, requires a copy of the IDCTAB to be available. Both **tran** and **xytosky** have options for lists of positions to be supplied as text files to allow multiple positions to be efficiently transformed.

6.2.2 Absolute and Relative Astrometry

The astrometric information in the header of an ACS image comes indirectly from the positions of the guide stars used. As a result the absolute astrometry attainable by using the image header world coordinate system directly is limited by two sources of error. First, the positions of guide stars are not known to better than about 1 arcsecond. Second, the mapping from the guide star to the instrument aperture introduces a smaller, but significant error.

Although absolute astrometry cannot be done to high accuracy without additional knowledge, relative astrometry with ACS is possible to a much higher accuracy. In this case the limitations are primarily the accuracy with which the geometric distortion of the camera has been characterized. This is discussed in detail in Section 4.1. Typical accuracy of the distortion correction in the pipeline with fourth order polynomial solutions is 0.1 pixels (5mas for the WFC). The application of the work of Anderson and

King will allow even better distortion correction by the inclusion of additional small corrections held in image files (e.g., Figure 4.6) which can model effects not included in the polynomial solutions. We plan to include these corrections in the pipeline in late 2004.

6.2.3 Impact of Guide Star Failure

The guiding performance and pointing stability of HST are described in the HST Primer. The normal guiding mode uses two guide stars that are tracked by two of HST's Fine Guidance Sensors (FGSs). However, sometimes two suitable guide stars are not available and single-star guiding is used instead with the telescope roll controlled by the gyros. These observations will suffer from small drift rates. To determine the quality of tracking during these observations please review Appendix C of the Introduction to the HST Data Handbook.

The gyros have a typical drift rate of 1.0 mas/sec, although on rare occasions it can be as large as 1.5 mas/sec. This causes a rotation of the target around the single guide star, which in turn introduces a small translational drift of the target on the detector. The exact size of the drift depends on the exact roll drift rate and distance from the single guide star to the target in the HST field of view. For ACS with single-star guiding, the typical and maximum drift rate of the target on the detector are in Table 6.5.

Table 6.5: Drift Rates for single-star guiding with ACS

TYPICAL	per 1000 sec exposure	per orbit(96 min)
WFC	0.0041 arcsec (0.08pix)	0.024 arcsec (0.47 pix)
HRC	0.0048 arcsec (0.19 pix)	0.028 arcsec (1.11 pix)
MAXIMUM	per 1000 sec exposure	per orbit (96 min)
WFC	0.0080 arcsec (0.16 pix)	0.046 arcsec (0.92 pix)
HRC	0.0092 arcsec (0.37 pix)	0.053 arcsec (2.12 pix)

The drift over an orbital visibility period can be calculated from these numbers; the typical visibility period in an orbit (outside the Continuous Viewing Zone [CVZ]) is in the range 52-60 minutes, depending on target declination (see Section 6.3 of the Cycle 12 HST Primer). The drifts inherent to single-star guiding have two consequences:

- There will be a slight drift of the target on the detector within a given exposure. For the majority of observations and scientific applications this will not degrade the data (especially if the exposures are not very long). The drift is smaller than the FWHM of the point spread func-

tion (PSF). Also, the typical jitter of the telescope during an HST observation is 0.003-0.005 arcsec, even when two guide stars are used.

- There will be small shifts between consecutive exposures. These shifts can build up between orbits in the same visit. This is not a problem if, during data reduction, different exposures are aligned before co-addition or comparison. The **MultiDrizzle** software provides a way of doing this. Also, there is always a slow drift of the telescope up to 0.01 arcsec/orbit due to thermal effects, even when two guide stars are used. So even when two guide stars are used it is often necessary to align exposures before co-addition.

In summary, for most scientific applications single-star guiding will not degrade the usefulness of ACS data. However, we do not recommend single-star guiding for the following applications:

- Programs that require very accurate knowledge of the PSF, including coronagraphic programs and astrometric programs.
- Programs that rely critically on achieving a dithering pattern that is accurate on the sub-pixel scale. (However, note that even with two-star guiding this can often not be achieved).

Observers who are particularly concerned with the effect of pointing accuracy on the PSF can obtain quantitative insight using the **TinyTim** software package. While this does not have an option to simulate the effect of a linear drift, it can calculate the effect of jitter of a specified RMS value.

6.3 Polarimetry

6.3.1 Absolute Calibration

ACS contains a set of six filters which are sensitive to linear polarization; there are three visible polarizer filters with their polarization directions set at nominal 60 degree angles to each other, and three UV polarizer filters arrayed in a similar manner. These filters are typically used in combination with a spectral filter which largely defines the spectral bandpass. In most cases observers will obtain images of the target in each of the three filters. The initial calibration steps for polarization data are identical to that for data taken in any other filter -- the data are bias corrected, dark subtracted, and flat fielded in the normal manner. The polarization calibration itself is accomplished by combining the set of images (or the resulting counts measured on the images) in the three filter rotations to produce a set of I, Q, and U images, or equivalently, a set of

images giving the total intensity, fractional polarization, and polarization position angle.

6.3.2 Instrumental Issues

The design of ACS is far from ideal for polarimetry. Both the HRC and WFC optical chains contain three tilted mirrors and utilize tilted CCD detectors. These tilted components will produce significant polarization effects within the instrument that must be calibrated-out for accurate results. There are two primary effects in the tilted components -- diattenuation and phase retardance. Diattenuation refers to the fact that a tilted component will likely have different reflectivities (or transmissions) for light which is polarized parallel and perpendicular to the plane of the tilt. This can be an important source of instrumental polarization, and can also alter the position angle of the polarization E-vector. The second effect, phase retardance, will tend to convert incident linear polarized light into an elliptically polarized light. These effects will have complex dependencies on position angle of the polarization E-vector, and hence will be difficult to fully calibrate. Additional discussion of these effects can be found in (WFPC2 ISR 97-11) and Biretta, et al., 2004 (ACS ISR in preparation).

The instrumental polarization, defined as the instrument's response to an unpolarized target, provides a simple measure of some of these effects. Figure 6.8 shows the instrumental polarization derived for the HRC through on-orbit observations of unpolarized stars (HST programs 9586 and 9661). The instrumental polarization is approximately 5% at the red end of the spectrum, but rises in the UV to about 14% at the shortest wavelengths. Also shown is a rough model for the effects of the M3 mirror together with a very crude model of the CCD. The mirror is Aluminum with a 606 Å thick overcoat of Magnesium Fluoride and has an incidence angle of 47 degrees. The details of the CCD are proprietary so we have simply modeled it as Silicon at an incidence angle of 31 degrees; no doubt this is a serious over-simplification. Figure 6.9 shows the same plot for the WFC, which has an instrumental polarization around 2%. Here the IM3 mirror is a proprietary Denton enhance Silver Coating with an incidence angle of 49 degrees, and the CCD has an incidence angle of 20 degrees. While the lower instrumental polarization of the WFC seems attractive, we caution that the phase retardance effects are not known for the Denton coating, and have some potential to cause serious problems -- if sufficiently large, the retardance could produce a large component of elliptical polarization which will be difficult to analyze with the linear polarizers downstream.

Figure 6.8: Instrumental Polarization for the HRC

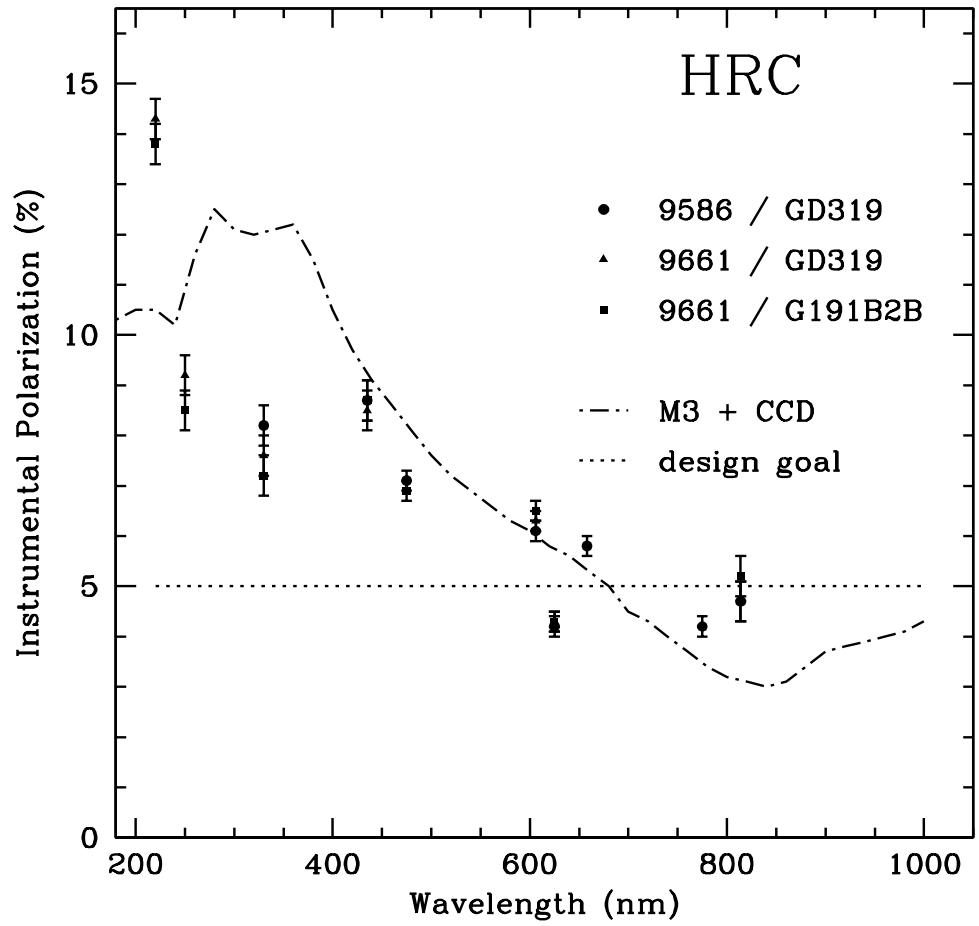
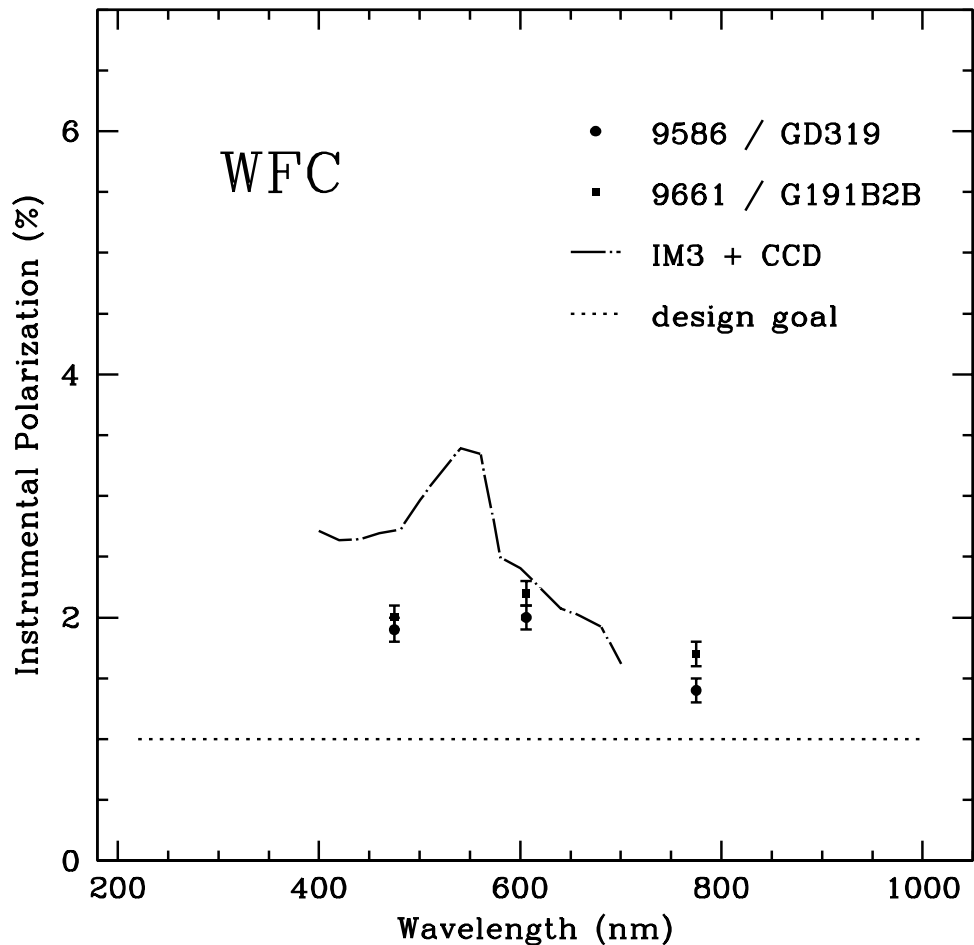


Figure 6.9: instrumental Polarization for the WFC



The ACS polarizer filters were characterized prior to installation in ACS by Leviton and the results as summarized in Figure 5.1 of the ACS Instrument Handbook. The cross-polarized transmissions are essentially zero for the POLV set, but are significant in the UV (5% - 10%) and far-red (20%) for the POLUV set.

One further issue for polarizer data is added geometric distortion. The polarizers contain a weak lens which corrects the optical focus for the presence of two filters in the light path. The lens causes a large-scale distortion which appears to be well-corrected by the drizzle software. There is also, however, a weak (± 0.3 pixel) small-scale distortion in the images caused by slight ripples in the polarizing material. There is presently no correction available for this, although we hope to have one soon. There is also the possibility of polametric field dependencies; while there has been study of intensity flats for the polarizers, the polarization field dependencies are not known.

6.3.3 Flats

Flat fields for the ACS polarizers were obtained in the laboratory and corrected for low-frequency variations using the in-flight L-flat corrections which were derived for the standard (non-polarizer) filters. The pivot wavelength of the combined optical components is typically within 1% when the standard filters are used in combination with the polarizers instead of with the clear filters. To assess the accuracy of this approximation, we compared in-flight observations of the bright earth using the F435W+POLUV filters and the F475W+POLV filters with the corrected laboratory flats. The HRC earth flats agreed with the corrected lab flats to better than 1%, where the largest deviations occurred near the edges of the detector. Improved polarizer L-flats are currently being derived from stellar observations, and these should bring the flat field uncertainties down to <1%.

6.3.4 Polarization Calibration

An extensive series of on-orbit polarization calibration observations were carried out at the end of Cycle 11 (programs 9586 and 9661). These included observations of unpolarized and polarized standard stars, the star cluster 47 Tuc, and an extended reflection nebula. Additional observations of polarized standards are needed over a wide and well-sampled range of HST roll angles; this will help quantify the angular dependences which are expected as the wavefront interacts with the diattenuation and phase retardation in the mirrors and CCD. Some of these observations are included in program 10055, which is underway (May 2004).

We now present a highly-preliminary calibration for use by polarization observers which is based largely on data in programs 9586 and 9661. (Observers are urged to check the ACS web pages for updates and improvements, since the material below will almost certainly be obsolete before this appears in print.) The strategy here is to create a calibration based upon the small amount of data which are available as of this writing. This calibration can be applied to either aperture photometry results, or to the images themselves (i.e., for an extended target). We begin by calibrating the polarization “zeropoint” using corrections which are derived from observations of unpolarized standard stars. Corrections $C(\text{CCD}, \text{POLnXX}, \text{spectral filter}, n)$ are applied to the observed count rate r_{obs} in each of the three polarizers (POLnUV or POLnV, where $n=0, 60, 120$). These corrections are tabulated in Table 6.6, and have been scaled such that Stokes I will approximate the count rate seen with no polarizing filter.

$$r(n) = C(\text{CCD}, \text{POLnXX}, \text{spectral filter}, n) r_{\text{obs}}(n)$$

Next, an “instrumental” Stokes vector is computed for the target. We note our definition of U is slightly different from that in the ACS Instrument Handbook:

$$I = \left(\frac{2}{3}\right)[r(0) + r(60) + r(120)]$$

$$Q = \left(\frac{2}{3}\right)[2r(0) - r(60) - r(120)]$$

$$U = \left(\frac{2}{\sqrt{3}}\right)[r(60) - r(120)]$$

Next we compute the fractional polarization of the target. We include a factor which corrects for cross-polarization leakage in the polarizing filters. This correction is useful for the POLUV filters, and values of T_{par} and T_{perp} can be found in Figure 5.4 of the ACS Instrument Handbook:

$$P = \frac{\sqrt{Q^2 + U^2}}{I} \times \left[\frac{T_{\text{par}} + T_{\text{perp}}}{T_{\text{par}} - T_{\text{perp}}} \right]$$

Finally we compute the position angle on the sky of the polarization E-vector. The parameter PAV3 is the roll angle of the HST spacecraft, and is called PA_V3 in the data headers. The parameter χ contains information about the camera geometry which is derived from the design specifications; for HRC we have $\chi = -69.4$ degrees, and for the WFC we have $\chi = -38.2$ degrees. Note that the arc tangent function must be properly defined; here we use a definition where the result is positive in quadrants I and II, and negative in III and IV.

$$PA = \left(\frac{1}{2}\right) \text{Tan}^{-1}\left(\frac{U}{Q}\right) + (PAV3) + \chi$$

For example, a target that gives 65192, 71686, and 66296 counts per second in the HRC with F606W and POL0V, POL60V, and POL120V, respectively, is found to be 5.9% polarized at PA=96.9 degrees.

We have modeled the full instrumental effects and the above calibration together in an effort to determine the impacts of the remaining uncalibrated systematic errors. We believe these will cause the fractional polarizations to be uncertain at the one-part-in-ten level (e.g. a 20% polarization has an uncertainty of 2%) for highly polarized sources; and at about the 1% level for weakly polarized targets. The position angles will have an uncertainty of about 3 degrees. (This is in addition to uncertainties which arise from photon statistics in the observer’s data.) We have checked this calibration against polarized standard stars (~5% polarized) and found it to be reliable within the stated errors. Better accuracy will require improved models for the mirror and detector properties as well as additional on-orbit data. We have not given a calibration for F220W, F250W, or F814W, as we believe

these are too unreliable at this time. There is also some evidence of a polarization pathology in the F625W filter, and observers should be cautious of it until the situation is better understood. In addition, we have seen one incidence of a 5 degree PA error for F775W, suggesting this waveband is not calibrated as well as the others.

Table 6.6: Correction factors C(CCD, POLnXX, specfilt, n) for polarization zeropoint.

CCD	POLnXX	Spectral Filter	n=0	n=60	n=120
HRC	POLnUV	F330W	1.7302	1.5302	1.6451
		F435W	1.6378	1.4113	1.4762
	POLnV	F475W	1.5651	1.4326	1.3943
		F606W	1.4324	1.3067	1.2902
		F625W ¹	1.0443	0.9788	0.9797
		F658N ¹	1.0614	0.9708	0.9730
		F775W	1.0867	1.0106	1.0442
WFC	POLnV	F475W	1.4303	1.4717	1.4269
		F606W	1.3314	1.3607	1.3094
		F775W	0.9965	1.0255	1.0071

1. Not scaled for Stokes I

6.4 Coronagraphy

The coronagraphic mode of ACS HRC provides improved imaging of low-contrast features, such as circumstellar disks, compared to what is possible with direct, normal imaging. It does this by preventing saturation of the detector and suppressing the diffraction pattern of the central source.

While the coronagraphic data appears at first to be similar to normal ACS images, there are some important differences. The calibration steps for both types of data are the same, except that the coronagraphic flat fields are different and for the moment must be manually applied to the images. The coronagraphic Lyot stop reduces throughput and must be accounted for when performing photometric measurements, and the stop also alters the PSF of field sources.

6.4.1 Flat Fielding

The coronagraphic flat fields are different from normal HRC flats. The most obvious differences are the shadows of the two occulting spots located at the center and the upper left corner. Because the spots are in the aberrated HST beam, there is vignetting out to 0.5 arcsec past the spot edges, causing them to appear somewhat diffuse. In addition, the Lyot stop alters the large scale flat field variation pattern as well as the diffraction pattern caused by dust specks and filter pinholes. Because the occulting spots wander by several pixels over weekly time scales, their effects must be treated separately from the static features in the flat field. Therefore, there are two separate flats that must be applied to coronagraphic images.

The first is the static flat that is essentially like a normal flat, with dust specks, detector response patterns, etc. For certain filters, static flats have been created using either coronagraphic ground test flats (F606W, F814W) or earth flats (F330W, F435W, F475W). These have been modified by replacing the occulting spot patterns with corresponding regions from normal flats. These should be accurate to better than 2% over multi-pixel spatial scales, but they may be less accurate for pixel-to-pixel response. For all other filters, normal HRC flat fields are used as the static flat in the pipeline. These may create some visible artifacts where dust spots occur, causing localized errors of up to 10%. Observers should examine the flat fields used to calibrate their data to verify that any features are not caused by these errors. The STScI pipeline divides all coronagraphic data by the corresponding static flat when calibrated images are requested from the archive.

The second flat that applies to coronagraphic data is the spot flat that only contains the vignetting patterns of the occulting spots. This flat must be shifted so that the patterns match the positions of the spots at the time a given set of science images was taken. STScI regularly monitors the position of the small occulting spot using earth flats. A table of the positions over time is available on the ACS web site. As of the time of writing, the pipeline does not create nor uses spot flats, so they must be manually applied by the observer. They are available for download from the ACS web site in the HRC Coronagraph Flat Image List of the reference files section at:

http://www.stsci.edu/hst/acs/analysis/reference_files/hrc_coron_spotflat_list.html

There is a different spot flat for each fully-supported filter (F330W, F435W, F475W, F606W, F814W). The observer must shift the flat by the amount listed in the spot position table (also in the Reference Files section of the web site). Shifting via bilinear interpolation (an option in `imshift` in IRAF) is sufficient. Then, the pipeline-produced (or **CALACS**) FLT or CRJ files should be divided by the shifted spot flat. The images must be

divided by the spot flat before geometric correction (the pipeline-produced DRZ files do not account for the spot flat).

6.4.2 PSF Subtraction

While the coronagraph suppresses the central source's diffracted light, it does not affect the halo of scattered light created by the polishing errors in the telescope's optics. That must be subtracted out by using the image of an isolated star with a color similar to that of the source. Observers should observe a PSF reference star immediately before or after their science observations.

The reference PSF must be scaled to match the intensity of the central source in the science observation. The normalization constant can be derived from photometry of non-coronagraphic images of the stars (if such were taken), or by iterative adjustment until the subtraction residuals are visually minimized. Experience shows that the by-eye value is typically accurate to within 5% or less.

The reference PSF must also be aligned with the science image for optimal subtraction. Alignment errors of about 0.05 pixels can be seen by eye in the subtracted image. Subpixel registration is required and is best accomplished by iteratively adjusting the shift of the reference PSF until the subtraction residuals are minimized and symmetrical about the source (discounting any potential circumstellar material). To achieve the best results, higher-order interpolation methods should be used for shifting -- due to the high frequency structure in the scattered light halo, bilinear interpolation is insufficient. In IDL, for instance, the INTERPOLATE routine with cubic convolution interpolation will provide good results. In IRAF imshift using sinc interpolation may also suffice. It may be necessary to iteratively interleave shift and normalization adjustments.

6.4.3 Photometry

Observers with ACS coronagraphic data must account for the effects of the Lyot stop when making photometric measurements. The stop decreases field source intensity by 52.5% relative to the normal HRC imaging mode. Note that this reduction is included in SYNPHOT when the coron specifier is included in the observing mode (e.g., "acs,hrc,f606w,coron"). The stop also broadens the field PSF and places more light into the diffraction spikes and Airy rings (see Chapter 5 in the ACS Instrument Handbook).

6.4.4 Ghosting

A point-source-like ghost appears about 30 pixels to the lower right of every field star and is about 8 magnitudes fainter than the star. There are no ghosts known to be associated with the occulted central source.

6.5 Ramp Filters

6.5.1 Introduction

Five slots on ACS filter wheel #2 each contain a ramp filter unit that is made up of an inner, an outer, and a middle segment. For the sake of simplicity, we will refer to these three segments as individual inner (IRAMP), outer (ORAMP), and middle (MRAMP) ramp “filters” although strictly speaking, they form part of the same physical ramp filter unit.

All fifteen ramp filters can be used with the WFC over specified regions of the WFC1 and WFC2 chips. IRAMP and ORAMP filters can only be placed on the WFC1 or WFC2 chips, respectively, over regions which define the WFC1-IRAMP and WFC2-ORAMP apertures. The MRAMP filter is by default placed on the WFC1 chip over the WFC1-MRAMP aperture, but it can in principle also be placed on the WFC2 chip over the WFC2-MRAMP aperture. The different ramp apertures and their reference points on the WFC chip are shown in Table 8.1 and Figure 8.3 of the ACS Instrument Handbook. In practice, the observer specifies a ramp filter and a central wavelength, and the filter wheel is appropriately rotated in order to place the central wavelength at the reference point of the relevant aperture. With the HRC, only the five middle ramp filters are available and they cover the region over the HRC chip defined by the HRC aperture (Table 8.2; ACS Instrument Handbook).

The fifteen ramp filters provide a total wavelength coverage of 3700-10,700 Å. Twelve of the filters have a narrow bandpass ($\Delta\lambda/\lambda$) of 2% and three have a medium bandpass of 9% (Table 4.2; ACS Instrument Handbook).

6.5.2 Existing Ground Calibrations

For each of the 15 ramp filters and different specified central wavelengths, the transmission was measured as a function of wavelength over the bandpass. For each ramp, analytical fits to these measurements generated eleven transmission profiles spaced by 10% of the wavelength coverage of the ramp ([ISR ACS 00-05](#)). Using these delivered profiles, SYNPHOT interpolates the transmission curve for any wavelength within

the bandpass of each ramp filter. The interpolation errors are below 1% which is better than the measurement accuracy.

WFC and HRC flat fields have been derived from pre-flight data by illuminating the ramps with continuum and monochromatic light using the Refractive Aberrated Simulator/Hubble Opto-Mechanical Simulator (RAS/HOMS) (ISR ACS 02-01). Each of the five WFC ramp filter units associated with the five wheel #2 slots is made up of three segments (inner, outer, middle) which cover three different adjacent wavelength ranges. A composite WFC flat was made for each of the five ramp filter units by combining three pieces of full-field broadband surrogate LP-flats (flats that incorporate both low frequency and pixel-to-pixel variations) nearest in central wavelength. For instance, for the ramp filter unit made up of FR505N (middle), FR551N (inner), and FR601N (outer) segments, surrogate broadband LP-flats of F475W, F555W, and F606W were used. HRC ramp flats were made from surrogate HRC flats appropriate for the middle segment. These flats have been delivered to the ACS pipeline. The broadband flats reduce the rms structure in monochromatic P-flats from 1% to 0.2% except at the shortest measured wavelength of 3880 Å.

After flat fielding test ramp images with the relevant composite L-flats, correcting for anamorphic geometric distortion, rotating and median filtering to remove residual noise, the response for FR505N is found to be uniform to below 1% over a 30x80 arcsec field of view for the WFC and 11.5x24 arcsec FOV for the HRC. Out of the five middle ramp filters, the FR388N shows the least uniform response (< 10%) over such regions (ISR ACS 02-01).

Preliminary values for the dispersion of the middle ramp filters FR388N, FR459M, FR505N, FR656N, and FR914M are, respectively 0.192, 0.0399, 0.146, 0.120, 0.0219 mm/Å at the ramp filters

(<http://acs.pha.jhu.edu/instrument/filters/ramp/coefficients/wavelength/>).

Due to the extra divergence of the beam from the filter to the detector, the dispersion at the WFC and HRC detector is estimated to be a factor of 1.2 and 1.3 lower respectively.

6.5.3 Future In-flight Calibrations

While the existing ground-based surrogate broadband L-flats provide some preliminary calibrations, improved monochromatic and continuum ramp L-flats are planned in order to take into account the fact that the transmitted wavelengths and sensitivity varies as a function of position within the ramp aperture. Two calibration steps are planned in Cycle 12:

1. A well-calibrated region in the Orion Nebula is being observed with the primary goal of deriving monochromatic L-flats at specified central wavelengths for (a) the WFC (inner+outer ramp) H-alpha

(FR716N, FR782N) and [OIII] (FR551N, FR601N) ramp filters, (b) WFC middle ramp filters (FR388N, FR459M, FR505N, FR656N, FR914M), and (c) HRC middle ramp filters (FR388N, FR459M, FR505N, FR656N, FR914M).

- Two well-calibrated standard star fields in 47 Tucanae are being observed with the primary goal of deriving WFC and HRC continuum L-flats for three middle ramp filters (FR656N, FR388N, and FR505N). These filters are selected as they are frequently used to map the H-alpha 6563 Å, [OII] 3727 Å and [OIII] 5007 Å lines, which are of great astrophysical interest, and they include the filters which seem to be the least (FR388N) and most (FR505N) uniform.

6.6 Spectroscopy with the ACS Grisms and Prisms

The ACS WFC has a grism (G800L) which provides slitless spectra at a dispersion of ~ 40 Å/pixel in the first order over the whole field of view. The same grism used with the HRC produces higher dispersion spectra (~ 23 Å in first order). A prism (PR200L) available with the HRC provides slitless spectra from the UV cut-off to ~ 3500 Å. The SBC is fitted with two prisms (PR110L and PR130L) which provide far-UV spectra, the latter blocked below 1230 Å to prevent transmission of Lyman alpha.

All of the ACS spectroscopic modes present the well-known advantages and disadvantages of slitless spectroscopy. The chief advantage is area coverage enabling spectroscopic surveys; among the disadvantages are overlap of spectra, high background from the integrated signal over the passband and modulation of the resolving power by the sizes of dispersed objects. For ACS the low background from space and the narrow PSF make slitless spectra competitively sensitive to ground-based spectra from larger telescopes in the presence of sky background.

The primary aim of reduction of slitless spectra is to provide one-dimensional wavelength and flux calibrated spectra of all the objects with detectable spectra. The reduction presents special problems on account of the dependence of the wavelength zero point on the position of the dispersing object, the blending of spectra in crowded fields and the necessity of flat fields over the whole available wavelength range. For ACS slitless modes, a dedicated package has been developed by the ST-ECF to enable automatic and reliable extraction of large numbers of spectra. This package, called aXe

(<http://www.stecf.org/software/axe>)

is sufficiently general to be used for other slitless spectra applications. Nevertheless, for the more exacting extractions of small numbers of objects

requiring a high degree of interaction, then the relevant packages in `noao.twodspec`

(<http://www.stecf.org/scripts/irafhelp?twodspec>)

and `noao.onedspec`, may be more appropriate.

(<http://www.stecf.org/scripts/irafhelp?onedspec>)

6.6.1 What to Expect from ACS Slitless Spectroscopy Data

The default method for taking ACS slitless spectra is to take a pair of images, one direct and one dispersed, in the same orbit (using special requirement `AUTOIMAGE=YES`, see the [STScI Phase II Proposal Instructions](#) for details). Or at the least, to take them with the same set of guide stars and without a shift in between. The filter to be used for the direct image is set by the `AUTOIMAGE=YES` parameter or can be user selected in the range of the spectral sensitivity by setting `AUTOIMAGE=NO`. The direct image provides the reference position for the spectrum and thus sets the pixel coordinates of the wavelength zero point on the dispersed image. The adequacy of the direct image to provide the wavelength zero point can be assessed by checking the WCS of the direct and dispersed images. In general this can be trusted at the level of a pixel or better. Some information on the jitter during an exposure is available in the ACS Quicklook products (see [ACS ISR 99-10](#)). More detailed information about the pointing during an observation can be obtained from the jitter files. In the case of a direct image not being available, such as when the pairing with the direct image was suppressed (`AUTOIMAGE=NO`), then it is possible for G800L grism observations to use the zeroth order to define the wavelength zero point. However the detected flux in the zeroth order is a few percent of that in the first order and it is dispersed by about 5 pixels, so the quality of the wavelength information will be degraded.

As an example, Figure 6.10 shows a WFC2 image (one chip only) taken with F775W and Figure 6.11 shows the companion G800L dispersed image (for one chip). This illustrates well the general characteristics of slitless spectra and the characteristics of ACS spectra in particular.

Figure 6.10: Direct image of WFC chip2 in F775W.

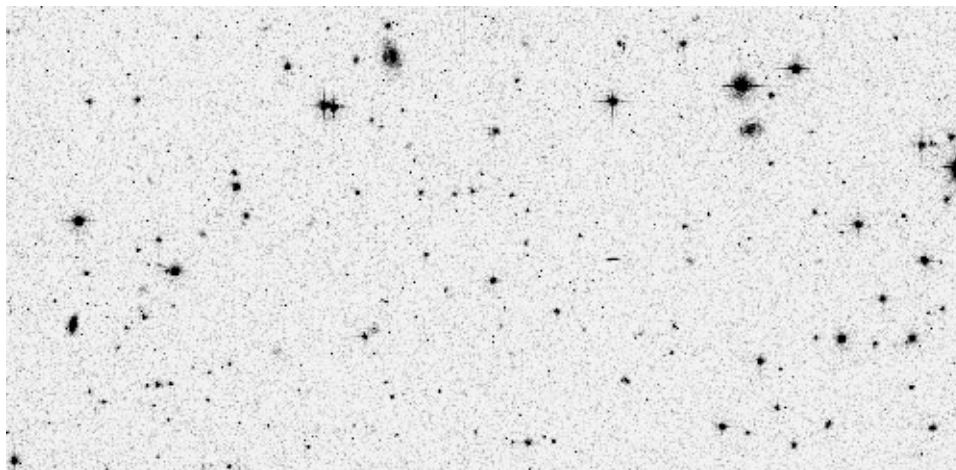
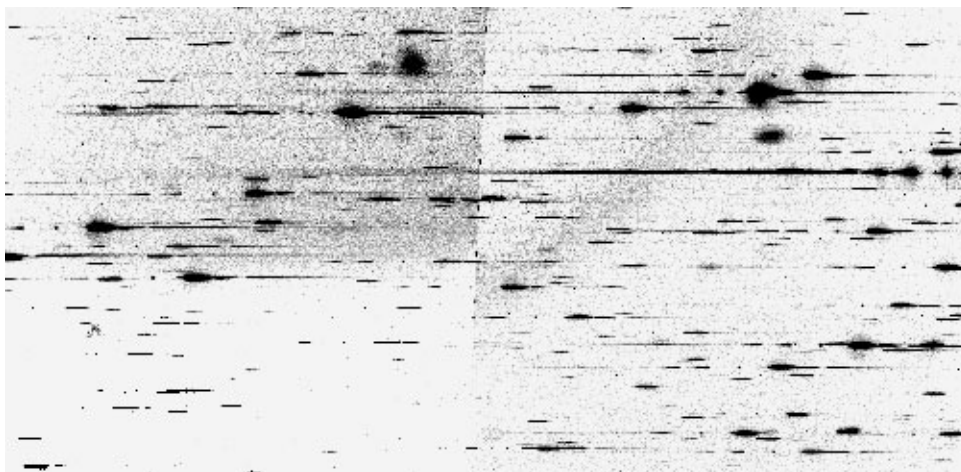


Figure 6.11: WFC G800L slitless image corresponding to the direct image.



Bright Stars

The brightest objects produce spectra which can extend far across the chip. Although for the G800L grism only about 10% of the flux is in orders other than the first, for bright objects, orders up to +4 and -4 can be detected. These spectra, while in principle can be analyzed (dispersion solutions have been derived for orders -3, -2, -1, 0, 1, 2 and 3 for the WFC G800L for example; see [ACS ISR 03-01](#)), provide a strong source of contamination for fainter spectra. In addition for the grism, the higher order spectra are increasingly out of focus and thus spread in the cross-dispersion direction. In the case of the prisms, the strongly non-linear dispersion can result in heavily saturated spectra to longer wavelengths. An additional effect for bright stars is the spatially extended spectra formed by the wings of the PSF.

Resolution and Object Size

In slitless spectrometry the object itself provides the "slit". The ACS PSF has a high Strehl ratio for most channels and over most of the accessible wavelength range of the dispersing elements. Therefore, the degradation of point sources beyond the theoretical resolution is minimal. However for an extended object the spectral resolution will be degraded depending on the size and light distribution in the object and spectral features will be diluted (see [ACS ISR 01-02](#)).

Zeroth Order

The grism zeroth order, only detectable for brighter objects since it contains about 3% of the total flux, can be mistaken for an emission line. The direct image can be used to determine the position of the zeroth order and for brighter sources, distinguish unequivocally between the zeroth order and an emission line.

Background

The background in a single pixel is the result of the transmission across the whole spectral range of the disperser and can thus be high depending on the spectrum of the sky background. The SBC PR130L prism, for example, provides lower detected background than the PR110L since it excludes the geocoronal Lyman alpha line. Bright objects in the field elevate the local background both in the dispersion direction and in the cross-dispersion direction. This background needs to be carefully removed before extracting the spectra of targets.

Crowding

Although Figure 6.10 and Figure 6.11 show a relatively uncrowded field, close examination shows many instances where spectra overlap, either in the dispersion direction or between adjacent spectra. It is important to know if a given spectrum is contaminated by that of a neighbor. This can be done by obtaining slitless spectra at a number of different roll angles, which improves the chances of cleanly extracting spectra, which overlap in the dispersion direction.

Extra-field Objects

There will inevitably be cases of objects outside the field which produce detectable spectra. This is more serious for the HRC and SBC where the fields are small and the spectra are long relative to the size of the detector. In such cases reliable wavelengths cannot be assigned since the zero point of the wavelength scale cannot be determined unless the zeroth order is also present. Brighter sources will be a source of contamination but correct flagging is hampered by the lack of knowledge of the precise image position.

6.6.2 Pipeline Calibration

The direct image half of the grism dispersed and direct image pair can be fully reduced by **CALACS** including bias subtraction, dark subtraction, flat fielding and inclusion of photometric zero point in the header (processing by ACS2D, see Section 3.4.3). However on account of the individual wavelength coverage of each object in the dispersed image, the pipeline reduction of slitless images is necessarily more restricted. Only the steps of the ACSCCD routine, involving noise and data quality initialization, bias subtraction and trimming of the overscan are applied. In addition, division by a dummy flat field (unity filled) is also applied in order to correct for the CCD gain, providing an image in electrons. This "flat" image (unity) is supplied as a pixel-to-pixel flat field file. Application of this dummy flat corrects very well the offsets in level between both quadrants of the WFC1 and WFC2 chips which have differing gain values. For the single quadrant detectors (HRC and SBC) this step merely corrects the ADU values by a single gain value. Table 6.7 shows the calibration switches appropriate for a WFC G800L frame.

Table 6.7: Calibration switch settings for ACS WFC slitless spectrometry frames

Keyword	Switch	Description
STATFLAG	F	calculate statistics? (T/F)
WRTErr	T	write out error array extension? (T/F)
DQICORR	PERFORM	data quality initialization?
ATODCORR	OMIT	correct for A to D conversion errors?
BLEVCORR	PERFORM	subtract bias level computed from overscan image?
BIASCORR	PERFORM	subtract bias image?
FLSHCORR	OMIT	subtract post flash image?
CRCORR	OMIT	combine observations to reject cosmic-rays?
EXPSCORR	PERFORM	process individual observations after cr-reject?
SHADCORR	OMIT	apply shutter shading correction?
DARKCORR	PERFORM	subtract dark image?
FLATCORR	PERFORM	apply flat field correction?
PHOTCORR	OMIT	populate photometric header keywords?
RPTCORR	OMIT	add individual repeat observations?
DRIZCORR	PERFORM	process dithered images?

No unique photometric keywords can be attached to all the spectra in a slitless image, so the photometric header keywords are left as default - see Table 6.7.

Table 6.8: ACS Slitless spectra (grism or prism) photometric keywords

Keyword	Value	Description
PHOTMODE	'ACS WFC2 G800L'	observation con
PHOTFLAM	0.00000E+00	inverse sensitivity, ergs/cm ² /Ang/electron
PHOTZPT	0.00000	ST magnitude zero point
PHOTPLAM	0.00000	Pivot wavelength (Angstroms)
PHOTBW	0.00000	RMS bandwidth of filter plus detector

In contrast to direct imaging data, no single flat field image can be correctly applied to slitless spectroscopy data, since each pixel cannot be associated with a unique wavelength. Rather, flat fielding is applied during the extraction of spectra in the aXe task aXe_PETFF (see “Main Steps of an aXe Extraction” in Section 6.6.3). Each pixel receives a flat field correction dependent on the wavelength falling on that pixel as specified by the position of the direct image and the dispersion solution.

For all subsequent reductions of slitless data, the pipeline product FLT files should be the starting point. The units of the SCI and ERR extensions are electrons.

6.6.3 Extracting and Calibrating Slitless Spectra

The set of C programs under the package name of aXe (also available in **pyraf.stsdas.acs**) provides a streamlined method for extracting spectra from the ACS slitless spectroscopy data. The starting point is always a direct image, a dispersed slitless image and a catalogue of the objects on the direct image. For each slitless mode (e.g., WFC G800L, HRC G800L, HRC PR200L, SBC PR110L and SBC PR130L), a configuration file is required which contains the parameters of the spectra. Information about the location of the spectra relative to the position of the direct image, the tilt of the spectra on the detector, the dispersion solution for various orders, the name of the flat field image and the sensitivity (flux per Å/e⁻/sec) table, is held in the configuration file and enables the full calibration of extracted spectra. Since the configuration file is quite general, it can be adapted to other HST instruments with slitless spectroscopy facilities, such as NICMOS.

There is a detailed [aXe manual](#) and so a only a brief outline of its use will be presented here. A demonstration package with direct and slitless

spectroscopy images and a script to drive the extraction can be downloaded from the aXe web page:

(<http://www.stecf.org/software/aXe>).

All catalogues are ASCII files and all image files are in FITS format. The data [SCI], data quality [DQ] and statistical error [ERR] data are accessed through the extension names.

The input slitless images for aXe are assumed to be in e^-/sec . Thus the FLT file must be normalized by the exposure time. Use of files in these units ensures that errors are correctly handled and propagated through the extraction process.

Main Steps of an aXe Extraction

1. Produce a deep catalogue (ASCII) of the objects on the direct image. The aXe package was designed to use a catalogue generated by Sextractor (<http://terapix.iap.fr/cplt/oldSite/soft/releases.html>). A minimum number of catalogue keyword parameters is required (see <http://www.stecf.org/software/aXe/sex/aXe.param>) to run aXe from this catalogue, but other catalogue formats could be converted to hold columns with these names.
2. Run aXe_SEX2GOL to convert the direct image Sextractor catalogue into a list of slitless image objects (called a Grism Object List, GOL). This task reads the WCS of both direct and slitless images and so can allow for shifts (usually small) in pointing between both images.
3. Run aXe_GOL2AF to convert the output catalogue of aXe_SEX2GOL into a list of BEAM specifications where slitless spectra are predicted, called an Object Aperture File (OAF). A BEAM is defined as a single order of the spectrum of a single catalogued object. This task requires the configuration file for the slitless mode. aXe_GOL2AF can be used to specify the extraction boxes for objects (OAF) and for the corresponding region for the background (Background Aperture File, BAF).
4. Run aXe_AF2PET to copy the pixel values for each aperture defined by an OAF or BAF to a Pixel Extraction Table (PET). For the object apertures the PET is referred to an Object PET (OPET) and for background as a Background PET (BPET). A PET is a FITS binary table which for each pixel within an aperture holds trace information (such as wavelength and distance from the predicted position of the spectrum trace) as well as data value, statistical error and data quality.
5. Run aXe_BE to generate a FITS file of the background used in subtracting the object spectra. There are various interpolation options for the background including the option to use the whole column for background estimation. The resulting FITS file is not used for further

processing but provides the user with an image of the background used. This is very useful for checking purposes or could be used for subsequent processing, outside aXe.

6. Run `aXe_PET2CONT` to check whether any pixels within a BEAM are contaminated by pixels from another BEAM. If a pixel within one beam is contaminated by one or more BEAM of other targets, the contamination column of the PET file is incremented.
7. Run `aXe_PETFF` to flat field a PET table by a flat field cube. The flat field cube is formed by a polynomial fit to the flat field variation of each pixel with wavelength over the wavelength range of the detector. Both the OPET and BPET must be identically flat fielded.
8. Run `aXe_PET2SPC` to extract each BEAM of each spectrum from its pixels, collapsing the 2D spectrum to a 1D spectrum, subtracting the background and applying wavelength and flux calibration. The result is a single FITS binary table containing all the spectra extracted from the slitless image.
9. Run `aXe_STAMP` to produce a "stamp" image of the region of each BEAM of each extracted object. This is also a single FITS binary table for the slitless image. This tool is primarily used for checking the veracity of an extraction. The stamp images can be rectified to give an orthogonal wavelength cross-dispersion image, similar to a long slit image.

Output Products: Extracted Spectra

The primary output of aXe is the file of extracted spectra (SPC). This is a multi-extension FITS binary table with as many table extensions as there are extracted beams. The beams are designated by an alphabetic postscript; the correspondence with order being defined by the configuration file. For example the G800L WFC 1st order is BEAMA in the configuration file `ACS.WFC.CHIP2.Cycle11.4.conf` see:

<http://www.stecf.org/instruments/acs/calib/Cycle11/WFC/ACS.WFC.CHIP1.Cycle11.4.conf>.

The first order spectrum of the object with number 8 in the input SExtractor catalogue for the slitless image `grism.fits`, would therefore be accessed as `grism.SPC.fits[BEAM_8A]`. The table contains 12 columns listing wavelength, total and extracted and background counts and their errors, the calibrated flux and error, the weight and a contamination flag. The header of the primary extension of the SPC table is a copy of the header of the frame from which the spectra were extracted.

The stamp image is also a multi-extension fits file with each extension containing the image of a single extracted beam. The other output products -- background image and contamination map are simple FITS files. The PET files are multi-extension FITS binary tables organized similarly to the

SPC file, but are files for intermediate products and their contents are not normally inspected.

Tuning the Extraction Process

All these tasks have input and output parameters so they may be tuned in various ways. See the aXe manual for further details.

aXe was planned as a set of tools which can be used to build up a full reduction of a slitless spectroscopy image. For example, several of the tools are run repeatedly as part of a full reduction. It proves most convenient to run aXe as a script which successively calls the required tasks. An example script is contained in the demo and is described in detail in the aXe manual. Often a reduction will need to be repeated, for example, changing the cut between the magnitude of objects and BEAMs to be extracted and those "marked" (i.e., considered to contribute to contamination but are not extracted, such as the zeroth order of grism spectra or saturated spectra).

The output products from aXe consist of ASCII files, FITS images and FITS binary tables. The FITS binary tables can be assessed using the tasks in the **stdas.ttools** package and wavelength flux plots, with error bars, can be plotted using **stdas.graphics.stplot.sgraph** (see Figure 6.12 for an example). When there are many detected spectra on a single image, such as for WFC G800L, then a dedicated task **aXe2web** is available (<http://www.stecf.org/software/aXe>) to make html pages consisting of direct image cut-outs, stamp images and 1D spectra (see Figure 6.13). This enables convenient browsing of large numbers of spectra.

Two facets of the reduction of ACS slitless data are not yet handled. First, fringing occurs in the WFC and HRC detectors arising from multiple reflections within the CCD layers, with an amplitude highest in the wavelength range 8000-9500Å (see [ACS ISR 03-12](#)). Fringing modulates the signal of the slitless spectra and should be corrected for the highest fidelity applications. Second, for optimal extraction of faint spectra, weights should be applied to the data points during the extraction process. While the PET files have a column for weight, which is applied during the extraction process (**aXe_PET2SPC**), the current version sets the value of the weights to unity. A task to reset the weights based on the cross-dispersion profile shape is required.

Figure 6.12: Example WFC G800L extracted spectrum.

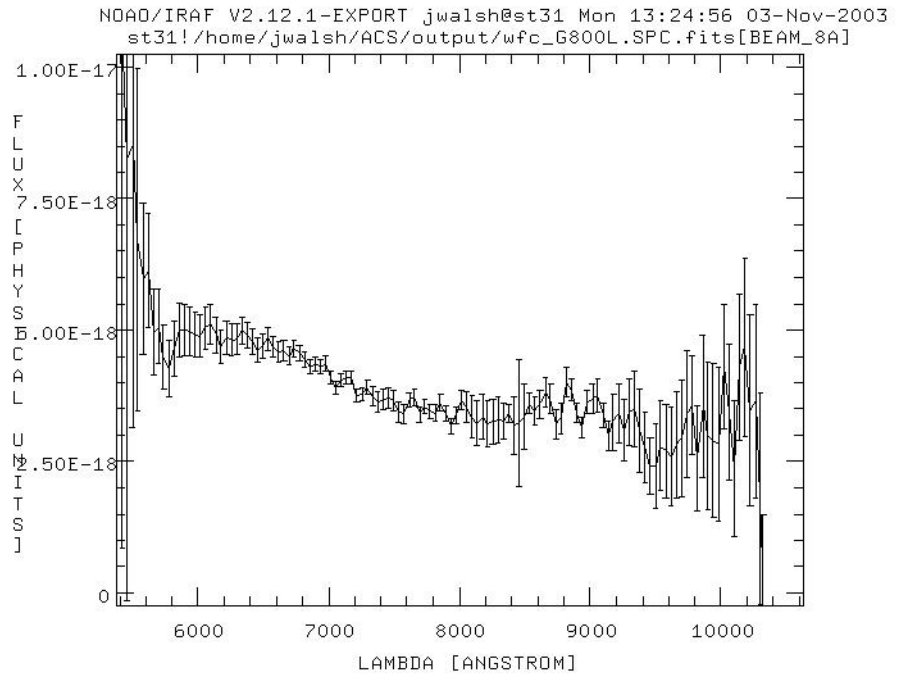
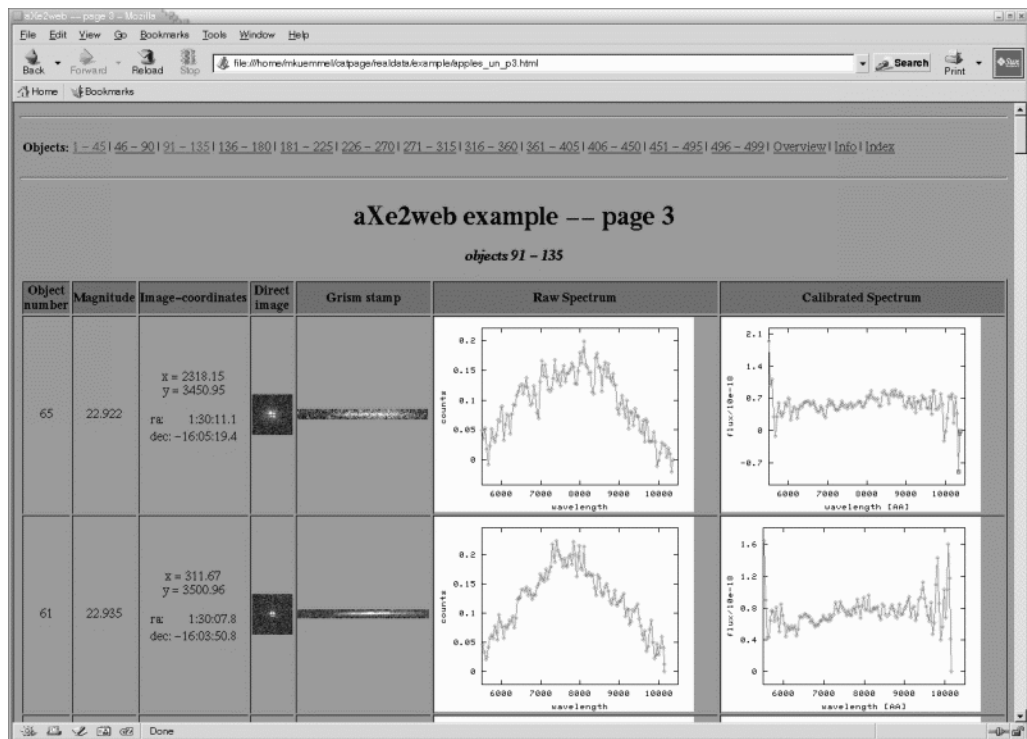


Figure 6.13: Web page produced by axe2Web for browsing extracted spectra.



6.6.4 How to Handle Dithered Slitless Spectroscopy Data

The necessity to dither ACS imaging data in order to improve the spatial resolution and to facilitate removal of hot pixels, applies equally well to slitless spectroscopy data. Typically for long slitless spectroscopic exposures, and especially for parallel slitless data, the data taking is broken into many sub-orbit dithered exposures. During the combination of ACS imaging data by **PyDrizzle**, the correction for the large geometrical distortion has to be taken into account. However slitless data has both a spatial and a spectral dimension and correction for the geometrical distortion, while applicable to the cross-dispersion direction, will distort the wavelength dispersion. Tracing the light path for slitless data shows that the distortions apply to the position and shape of images as they impinge on the grism, but not from the grism/prism to the detector.

Extraction of drizzled and geometrically corrected slitless data for WFC (the ACS channel in which the geometrical distortion is largest) have shown that the correction has the effect of decreasing the resolution (increasing the dispersion in Å/pixel) across the whole field by around 10%. This is clearly disadvantageous but a configuration file for extracting dithered WFC G800L images is available at

<http://www.stecf.org/instruments/acs/calib/Cycle11/WFC/ACS.WFC.drizzle.2.conf>

although this configuration file only handles first and zeroth order spectra. No flat field can be applied to drizzled slitless combined data since each pixel in the drizzled data is a combination of multiple input pixels, each with different wavelength response.

The **MultiDrizzle** script (see Section 4.5.2) however provides a very convenient tool for combining many dithered slitless spectral images in order to provide a deep slitless image, free of cosmic rays, for examination. The companion direct images can also be combined with **MultiDrizzle** to derive a deep catalogue. However for extraction of the spectra, it is recommended to extract the spectra from each image separately and combine the 1D spectra for each object. **MultiDrizzle** is then applied to provide data quality images with the cosmic rays and hot pixels flagged, so that these affected pixels are not included in the extracted spectra. Thus **MultiDrizzle** is run on all the slitless data files in a set, the data quality arrays in the FLT files are updated, and then aXe is run on this set of data files. The individual extracted wavelength, flux spectra can then be converted to FITS images and noao.onedspec.scombine used, for example, to merge all the spectra of each object. An alternative, preferable, approach under investigation is to combine the images of the slitless spectra in 2D. The pixels from the PET files from different exposures could then be drizzled and cosmic rays flagged using **MultiDrizzle**. Spectra could be

extracted using `axe_PET2SPC` to produce the final combined wavelength and flux calibrated 1D data.

6.6.5 Accuracy of Slitless Spectra Wavelength and Flux Calibration

Wavelength Calibration

The dispersion solution was established by observing astronomical sources with known emission lines (e.g., for the WFC G800L and the HRC G800L, Wolf-Rayet stars were observed see [ACS ISR 03-01](#) and [ACS ISR 03-07](#)). The field variation of the dispersion solution was mapped by observing the same star at different positions over the field. The internal accuracy of these dispersion solutions is good (see the ISR's for details) with an rms generally less than 0.2 pixels.

For a given object the accuracy of the assigned wavelengths depends most sensitively on the accuracy of the zero point and the transfer of the zero point from the direct to the slitless spectrum image. Provided that both direct and slitless images were taken with the same set of guide stars, systematic pointing offsets less than 0.2 pixels can be expected. For faint sources the error on the determination of the object centroid for the direct image will also contribute to wavelength error. Realistic zero point errors of up to 0.3 pixels are representative. For the highest wavelength accuracy it is advised to perform a number of pixel offsets of about ten percent of the field size for each of several direct and slitless image pairs and average the extracted spectra.

Flux Calibration

The sensitivity of the dispersers was established by observing a spectrophotometric standard star at several positions across the field. The sensitivity (aXe uses a sensitivity tabulated in $\text{ergs/cm/cm/sec/\AA}$ per detected \AA) was derived using data flat fielded by the flat field cube (see Section 6.6.2). In fact the adequacy of the WFC flat field was established by comparing the detected counts in the standard star spectra at several positions across the field. These tests showed that the sensitivity at the peak of the WFC G800L varied by less than 5% across the whole field. However at the lower sensitivity edges of the spectra, to the blue ($<5,800 \text{ \AA}$) and to the red ($>10,000 \text{ \AA}$), the counts in the standards are low, and the errors in flux calibration approach 20%. In addition, small errors in wavelength assignment have a large effect in the blue and red where the sensitivity changes rapidly with wavelength. This often leads to strong upturns at the blue and red end of extracted flux calibrated spectra, whose reality may be considered suspect.



PART III:

Appendixes

■ Part III:Appendixes



APPENDIX A:

IRAF Primer

In this appendix . . .

A.1 Initiating IRAF / A-2
A.2 IRAF Basics / A-4
A.3 Getting IRAF and STSDAS / A-15

The Image Reduction and Analysis Facility (IRAF), developed by the National Optical Astronomy Observatories (NOAO), forms the basis of the Space Telescope Science Data Analysis System (STSDAS). IRAF contains numerous packages of programs, called *tasks*, that perform a wide range of functions from reading data tapes to producing plots and images. Most astronomers will already be familiar with IRAF, but we provide this tutorial for HST observers who are beginners with IRAF. It includes information on:

- How to set up IRAF the first time you use the software.
- How to start and stop an IRAF session.
- Basic concepts, such as loading packages, setting parameters, etc.
- How to use the on-line help facility.

Additional information on IRAF, in particular *A Beginner's Guide to Using IRAF* is available through the NOAO IRAF Home Page at: <http://iraf.noao.edu>

A.1 Initiating IRAF

This section explains:

- How to set up your IRAF working environment.
- How to start and logout of the IRAF program.



We assume that your site has IRAF and STSDAS installed. If not, you must obtain and install the software. See Appendix Section A.3 for details.

A.1.1 Setting Up IRAF in Unix/Linux

Before running IRAF for the first time you need to follow these three steps:

1. Create your IRAF root directory.
2. Move to that directory and set the necessary environment variables or system logicals and symbols.
3. Run **mkiraf** to create a `login.c1` file and a `uparm` subdirectory.

Users generally name their IRAF home directory `iraf` (also referred to as your IRAF *root* directory) and set it up in their account's root directory (i.e., the default directory that you are in when you log in to the system). The IRAF home directory doesn't need to be in your account's root directory, nor does it need to be called `iraf`, but you should *not* put it on a scratch disk that is periodically erased.

If you call your root IRAF directory "`iraf`", you can set up IRAF as follows:

Under Unix:

```
% mkdir iraf
% cd iraf
% setenv iraf /usr/stsci/iraf/
% source $iraf/unix/hlib/irafuser.csh
% mkiraf
```

Can be placed
in `.login` file →

← The directory name is
site-dependent—check
with your system staff

The **mkiraf** command initializes IRAF by creating a `login.cl` file and a subdirectory called `uparm`. After typing the **mkiraf** command, you will see the following:

```
% mkiraf
-- creating a new uparm directory
Terminal types: gterm=ttysw+graphics,vt640...
Enter terminal type:
```

Enter the type of terminal or workstation you will most often use with IRAF.¹ Generic terminal types that will work for most users are:

- `vt100` for most terminals.
- `xtermjhs` for most workstations running under X-Windows.
- `xgterm` for sites that have installed X11 IRAF and IRAF v2.10.3 BETA or later.

IRAF for other systems, like Mac OSX, can be obtained from the IRAF web page at: <http://iraf.noao.edu>



You can change your terminal type at any time by typing (`set term=new_type`) during an IRAF session. You can also change your default type by editing the appropriate line in your `login.cl` file.

After you enter your terminal type, you will see the following output before getting your regular prompt:

```
A new LOGIN.CL file has been created in the current ...
You may wish to review and edit this file to change ...
```

The `login.cl` file is the *startup file* used by the IRAF command language (CL). It is similar to the `.login` file used by Unix. Whenever IRAF starts, it looks at the `login.cl` file. You can edit this file to customize your IRAF environment. In fact, you should look at it to make sure that everything in it is correct. In particular, there is a line starting with `set home =` that tells IRAF where to find your IRAF home directory. You should verify that this statement does, in fact, point to your IRAF directory. If you will be working with standard IRAF format images you should also insert a line saying `set imdir = "HDR$"`. The `imdir` setting is ignored when working with GEIS format images.

1. Users at STScI should consult the *STScI Site Guide for IRAF and STSDAS*.

The `uparm` directory will contain your own copies of IRAF task parameters. This directory allows you to customize your IRAF environment by setting certain parameter values as defaults. Once you set up IRAF, you should rarely need to do it again, except when updated version of IRAF are installed.

A.1.2 Starting and Stopping an IRAF Session

To start an IRAF session:

1. Move to your IRAF home directory.
2. Type `c1`.

IRAF starts by displaying several lines of introductory text and then puts a prompt at the bottom of the screen. Figure A.1 is a sample IRAF startup screen.

Figure A.1: IRAF Startup Screen

```

NOAO Sun/IRAF Revision 2.11 Fri Aug 15 15:34:46 MST 1997
This is the EXPORT version of Sun/IRAF V2.11 for SunOS 4 and Solaris 2.5

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type `help command'. To run a
command or load a package, type its name. Type `bye' to exit a
package, or `logout' to get out of the CL. Type `news' to find out
what is new in the version of the system you are using. The following
commands or packages are currently defined:

  apropos      euv.          local.         spptools.
  ared.        fitsutil.     mem0.         stlocal.
  aspec.       focas.        newimred.     stsdas.
  c128.        ftools.       noao.         system.
  color.       hst_pipeline. obsolete.     tables.
  ctio.        images.       plot.         utilities.
  dataio.     imcnv.        proto.        vol.
  dbms.        language.    rvsao.        xray.
  digiphotx.  lists.       softtools.

c1>

```

Startup Messages
Change from Day
to Day

Available Packages
and Tasks

To quit an IRAF session:

1. Type `logout`.

A.2 IRAF Basics

This section describes basic IRAF techniques such as:

- Loading packages (below).
- Running tasks and commands.

- Getting online help.
- Viewing and setting parameters (see Appendix Section A.2.4).
- Setting and using environment variables (see Section A.2.5).
- File management
- Troubleshooting

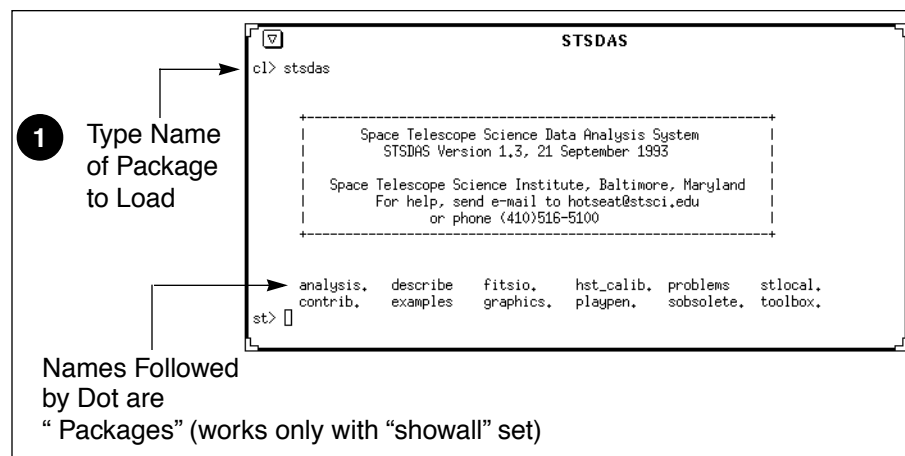
A.2.1 Loading Packages

In IRAF jargon, an application is called a *task* and logically related tasks are grouped together in a *package*. Before you can use a task, you must load the package containing that task. To load a package, type the name of the package. The prompt will then change to the first two letters of the package name, and the screen will display the names of all the newly available tasks and subpackages. Even though the prompt has changed, previously loaded packages remain loaded, and all their tasks remain available.

Note that the standard way to specify a path through the IRAF package hierarchy to a task in a particular subpackage is to separate the package names with periods (e.g., **stsdas.hst_calib.foc.focgeom.newgeom**).

Most frequently used packages can be added to your `login.c1` file. Just list the packages one per line in that file and each will be loaded automatically when Iraf is started.

Figure A.2: Loading Packages



Some helpful commands for managing packages are:

- ? - Lists tasks in the most recently-loaded package.
- ?? - Lists all tasks loaded, regardless of package.

- `package` - Lists names of all loaded packages.
- `bye` - Exits the current package.

A.2.2 Running Tasks

This section explains how to run tasks, background tasks, and system-level commands, and how to use piping and redirection.

Running a Task

The simplest way to run a task is to type its name or any unambiguous abbreviation of it. The task will then prompt you for the values of any required *parameters*, such as the names of input files. Alternatively, you can specify the values for the required *parameters* on the command line when you run the task. For example, if you want the task `imheader` to print header information on the file `myfile.hhh`, you can type

```
st> imhead myfile.hhh
```



IRAF does not require you to type the complete command name—only enough of it to make it unique. For example, “dir” is sufficient for directory.

Escaping System-Level Commands

To run an operating system-level command (i.e., Unix or VMS commands) from within the IRAF CL, precede the command with an exclamation point (!). This procedure is called *escaping* the command. For example:

```
st> !system_command
```

Piping and Redirection

You can run tasks in sequence if you desire, with the output of one task being used as the input for another. This procedure, called *piping*, and is done by separating commands with a vertical bar (|), using the following syntax:

```
st> task1 filename | task2
```

For example, if a particular task prints a large volume of textual output to the screen, you will often want to pipe it to `page`, which allows you to read the output one page at a time:

```
st> task1 filename | page
```

You can also redirect output from any task or command to a file by using the greater-than symbol (`>`) as follows:

```
st> command > outputfile
```

Background Tasks

To run a task as a background job, freeing your workstation window for other work, add an ampersand (`&`) to the end of the command line, like this:

```
st> taskname &
```

A.2.3 Getting Help

This section describes:

- How to use IRAF's on-line help facility.
- How to find a task that does what you want (see "Finding Tasks" on page A-8).
- IRAF tutorials and `c1` tips.

On-Line Help

You can get on-line help with any IRAF task or package by using the **help** command,² which takes as an argument the task or package name about which you want help. Wildcards are supported. For example, to display the on-line help for the STSDAS **mkmultispec** task, you would type:

```
fi> help mkmultispec
```

2. There is an optional *paging* front-end for help called **phelp**. For more information, type `help phelp` from within IRAF.

Figure A.3: Displaying On-line Help

```

STSDAS
MKMULTISPEC (Apr93)      stdasd,hst_calib,ctools      MKMULTISPEC (Apr93)

NAME
  mkmultispec -- Create a MULTISPEC WCS based on wavelength tables.

USAGE
  mkmultispec input wave output

DESCRIPTION
  This task takes input spectra and wavelength tables, uses the IRAF
  'curfit' routines to fit the wavelength table, creates a MULTISPEC
  world coordinate system (WCS) description of the fit and creates a
  new copy of the input spectrum with the new WCS.

  This task is intended to be used with the Hubble Space Telescope
  (HST) spectrographic data from Faint Object Spectrograph (FOS) and
  Goddard High Resolution Spectrograph (GHRS) to merge the
  wavelengths that result from the calibration into the data itself
  so that other IRAF tasks, such as 'splot', may be used. However,
  [q=quit,d=downhalf,f|sp=downfull,|lcr=downline,N=next]]
  
```

Available Commands Display Next Page

Two STSDAS tasks that display only certain sections of the help file are also available:

- **examples** - Displays only the examples for a task.
- **describe** - Displays only the description of the task.

Typing `help package` will produce one-line descriptions of each task in the package.

Finding Tasks

There are several ways to find a task that does what you need:

- Use `help package` to search through the IRAF/STSDAS package structure.
- Use the **apropos** task as shown in Figure A.4 to search the online help database. This task looks through a list of IRAF and STSDAS package menus to find tasks that match a specified keyword. Note that the name of the package containing the task is shown in parentheses.
- Ask a more experienced user, who can usually point you in the right direction.

IRAF Tutorials and c1 tips

A number of tricks and tutorials are available online to help the user to become more familiar with IRAF. See:

<http://iraf.noao.edu/iraf/web/tutorials/tutorials.html> and
<http://iraf.noao.edu/tips/c1.html>

Figure A.4: The **apropos** task Using **apropos**

```

STSDAS
ct> apropos WCS
wcslab - Overlay a displayed image with a world coordinate grid (cl.images,
tv)
wcsedit - Edit the image coordinate system (cl.proto)
wcsreset - Reset the image coordinate system (cl.proto)
makewcs - Write the WCS on the image header based on the plate sol. (stdsas,a
nalysis.gasp)
wcslab - Produce sky projection grids for images. (stdsas.graphics.stplot)
wlpars - Pset to specify characteristics of WCS labelled graphs. (stdsas.gra
phics.stplot)
wcpars - Pset to specify a WCS. (stdsas.graphics.stplot)
mkmultispec - Combine wavelength and data with the MULTISPEC MWCS. (stdsas.hst_c
alib.ctools)
ct>
  
```

A.2.4 Setting Parameters

Parameters specify the input information for IRAF tasks. They can be the names of input or output files, particular pixel numbers, keyword settings, or many other types of information that control the behavior of the task.

The two most useful commands for handling parameters are:

- **lparam** to display the current parameter settings (often abbreviated **lpar**).
- **eparam** to edit parameters (often abbreviated **epar**).

Viewing Parameters with **lparam**

The **lpar** command lists the current parameter settings for a given task (Figure A.5).

Figure A.5: Displaying Parameter Settings with **lpar**

```

STSDAS
1 Type lpar
  Followed by
  Name of Task
Parameters and
  Current Settings
fi> lpar strfits
fits_file = "mtg"           FITS data source
file_list = "1-999"        File list
iraf_file = ""             IRAF filename
(template = "")            template filename
(long_header = no)         Print FITS header cards?
(short_header = yes)       Print short header?
(datatype = "default")     IRAF data type
(blank = 0,)               Blank value
(scale = yes)              Scale the data?
(xdimgtf = yes)           Transform xdim FITS to multigroup?
(oldirafname = yes)       Use old IRAF name in place of iraf_file?
(offset = 0)               Tape file offset
(mode = "ql")
fi>
  
```

Setting parameters with **epar**

The **epar** command is an interactive parameter set editor. It displays all of the parameters and their current settings on the screen. You can move around the screen using the arrow keys (also called *cursor* keys) and type

new settings for any parameters you wish to change. Figure A.6 shows a sample of the **epar** editor at work (invoked by typing `epar strfits`).

Figure A.6: Editing Parameters with `epar` in IRAF

- 1 Move Through Parameters Using Arrow Keys
- 2 Type New Values For Parameter Settings
- 3 Type `:g` to Save Parameters and Run Task
- 4 Exit by typing `:q`

To List Line Editing Commands, Press

```

STSDAS
  IRAF
Image Reduction and Analysis Facility
PACKAGE = fitsio
TASK = strfits

fits_fil=          mtg  FITS data source
file_lis=          1-999  File list
iraf_fil=          )  IRAF filename
                   )  template filename
(long_he=         no)  Print FITS header cards?
(short_he=        yes) Print short header?
(datatyp=        default) IRAF data type
(blank =          0.)  Blank value
(scale =          )  yes) Scale the data?
(xdimtog=        yes) Transform xdim FITS to multigroup?
(oldiraf=        yes) Use old IRAF name in place of iraf_file?
(offset =         0)  Tape file offset
(mode =          ql)

ESC-? for HELP

```

Parameter Data Types—What to Specify

Parameters are either *required* or *hidden*, and each parameter expects information of a certain *type*. Usually, the first parameter is required, and very often it expects a file name. Parameters are described in the online help for each task [include reference to help]. Hidden parameters, shown in parentheses in the online help and the **lpar** and **epar** listings, need not be specified at each execution because their default values frequently suffice.



Wise IRAF users will check the values of hidden parameters, as they often govern important aspects of a task's behavior.

If you specify the wrong type of information for a parameter, **epar** will usually display an error message saying something like “Parameter Value is Out of Range.” The message is displayed when you move to another parameter or if you press . Table A.1 lists the different parameter types.

Table A.1: Parameter Data Types

Type	Description
File Name	Full name of the file. Wild card characters (* and ?) are often allowed. Some tasks allow you to use special features when specifying file names, including “@” lists, IRAF networking syntax, and image section or group syntax. (See “File Management” below).
Integer	Whole number. Often the task will specify minimum or maximum values (see the help pages).
Real	Floating point numbers, can be expressed in exponential notation. Often will have minimum and maximum values.
Boolean	Logical “yes” or “no” values.
String	Any characters. Sometimes file names are specified as string.
Pset	Parameter set.

Restoring Parameter Default Values

Occasionally, IRAF (or you) will get confused by your parameter values. To alleviate this confusion, you can restore the default parameters with the **unlearn** command. You can use **unlearn** on either a task or on an entire package.



The unlearn command generally will restore the parameters to reasonable values, a big help if you are no longer sure which parameter values you have changed in a complicated task.

A.2.5 Setting Environment Variables

IRAF uses *environment variables* to define which devices are used for certain operations. For example, your terminal type, default printer, and the disk and directory used for storing images are all defined through environment variables. Environment variables are set using the **set** command and are displayed using the **show** command. Table A.2 lists some of the environment variables that you might want to customize.

Table A.2: Environment Variables

Variable	Description	Example of Setting
printer	Default printer for text	set printer = lp2
terminal	Terminal type	set term = xgterm
stdplot	Default printer for all graphics output	set stdplot = ps2
stdimage	Default terminal display setting for image output (most users will want this set to either imt512 or imt800)	set stdimage = imt800
stdgraph	Default graphics device	set stdgraph = xgterm
clobber	Allow or prevent overwriting of files	set clobber = yes
imtype	Default image type for output images. "imh" is original IRAF format, (OIF) "hhh" is STS-DAS GEIS format.	set imtype = "hhh"



If you are working with GEIS files, you should set imtype to "hhh". If you are working with ACS, STIS and NICMOS data in FITS files, you can set imtype to "fits".

You can set your environment variables automatically each time you login to IRAF by adding the appropriate commands to your `login.cl` file. Use your favorite text editor to specify each variable on its own line. The **show** command with no arguments prints the names and current values of all environment variables.

A.2.6 File Management

This section describes:

- File formats commonly used with STSDAS and IRAF.
- Specification of file names.
- Navigation through directories.

File Formats

IRAF recognizes a number of different file structures. Among them are the standard HST file formats known as GEIS and FITS (see chapter 2 of the HST Introduction), both of which differ from the original IRAF format (OIF). GEIS is closer to OIF, in that two files are *always* used together as a pair:

- A *header file*, which consists of descriptive information. IRAF (OIF) header files are identified by the suffix `.imh`. GEIS header files are in ASCII text format and are identified by the suffix `.hhh` or another suffix ending in “h”, such as `.c0h` or `.q1h`.
- A *binary data file*,³ consisting of pixel information. IRAF data file names end with a `.pix` suffix. STSDAS data files end with a suffix of `.hhd` or another suffix that ends with “d”, such as `.c0d` or `.q0d`.

STSDAS always expects both component files of a GEIS image to be kept together in the same directory. A single FITS file contains both the header information and the data.



When working with IRAF or STSDAS images, you need only specify the header file name—the tasks will automatically use the binary data file when necessary.

File Specification

Most tasks in IRAF and STSDAS operate on files and expect you to specify a file name for one or more parameters. Several types of special syntax can be used with certain tasks when specifying file names. These syntax features include:

- **Wild card characters**, often called *templates*, which are used to specify multiple files using pattern matching techniques. The wild cards are:
 - * Matches any number of characters, e.g. `z*.c0h`
 - ? Matches any single character, e.g. `z01x23x.c?h`

3. The binary data file format is host-dependent and may require translation before it can be moved to a computer using a different architecture.



When using wildcards with image-processing tasks, be sure to exclude the binary pixel files by ending your file name specification with an “h”, for example: `y.??h`*

- **List files**, often called *@-files*, are ASCII files that contain lists of file names, one per line. If your task supports the list file feature, you would type the name of your list file, preceded by the “@” character. For example: `@infile.txt`
- **Image section** specification. Tasks that work with image data will allow you to specify a part of the image rather than the entire image. To extract a particular image section, specify each axis range in square brackets, for example: `image.hhh[10:200,20:200]`
- **IRAF networking** specification. IRAF is capable of reading and writing files to and from remote systems on a network. This feature is often used with tasks in the **fitsio** and **convfile** packages, or with image display tasks. The *STSDAS Users Guide* and the online help (type `help networking`) describe how to enable this feature. To specify that you want to use the IRAF networking feature, type the remote host name followed by an exclamation point (!), followed by the file or device name. For example: `ra!mta`.

For example, when displaying from an IRAF session running on a remote machine back to your work station set the environment variable “node” by typing: `set node= my_workstation!`

Directory Navigation

To navigate through directories, you can use the following commands:

- **path** or **pwd** - Lists the current working directory.
- **cd** *directory* - Move to the named directory.

A.2.7 Troubleshooting

There are a couple of easy things you can do to make sure that you don’t have a simple memory or parameter conflict—common causes of problems.

- Look at the parameter settings and make sure that you have specified reasonable values for every parameter.

- When you run an IRAF task for the first time in a session, IRAF stores the executable file in its *process cache*. If IRAF appears not to be running your tasks properly, you may need to use the **flprcache** command to clear the process cache. To do this type: `flpr`. Sometimes you will need to execute this command twice in succession.
- Occasionally, you may need to logout of the CL, restart IRAF, and try your command again.

If you still have a problem, contact the STSci Help Desk at help@stsci.edu

A.3 Getting IRAF and STSDAS

Both IRAF and STSDAS are provided free of charge to the astronomical community. You must have IRAF and PyRAF to run STSDAS. Detailed information about installing and retrieving STSDAS is found in the *STSDAS Site Manager's Installation Guide and Reference* which can be downloaded in postscript or PDF format from the World Wide Web page at:

http://www.stsci.edu/resources/software_hardware/stsdas/download

If you have any problems getting and installing STSDAS, TABLES, or any other packages or data described in this handbook, please contact the Help Desk by sending e-mail to: help@stsci.edu.

A complete description of how to install the **synphot** data files is provided in Section A.3.2.

A.3.1 Retrieving the IRAF and STSDAS Software

There are two ways to get the software:

- Use the World Wide Web page shown above

This page provides links and instructions for downloading the appropriate files to your local system or to display the software directory, from which you can select the series of smaller files.

- Use anonymous FTP

Type `ftp ftp.stsci.edu`

login: anonymous

password: (your email address)

and go to the directory `software/stsdas` to retrieve source and binary files. Binaries are platform-specific and the filename contains the name of the platform, see the README file in that directory.



When you retrieve STSDAS, you must also retrieve the TABLES package, and TABLES must be installed first. You must have IRAF and Pyraf installed on your system in order to install TABLES and STSDAS.

The complete instructions for installing STSDAS, TABLES, and all of the supporting software and reference files (including instrument reference files and the **synphot** dataset) for the IRAF environment are found in the file called `InstallGuide.book32.ps` (which contains a copy of the *STSDAS Site Manager's Installation Guide and Reference*). Ancillary information and documentation for STSDAS and **synphot** are available in the `software/stsdas/docs` directory on the ftp site.

Registration

The software can also be registered and requested using on-line forms available through World Wide Web at the following URL:

http://www.stsci.edu/resources/software_hardware/stsdas/STSDASRegistrationForm

When you request the STSDAS software, you can also ask for the appropriate version of IRAF, which will be requested for you— simply check the appropriate box on the form under “Do You Already Have IRAF Installed?” If you prefer to request the IRAF software independent of STSDAS, you can do so by sending e-mail to: iraf@noao.edu

A.3.2 Getting the Synphot Database

This manual sometimes refers to the **synphot** dataset, which must be available in order to run tasks in the STSDAS **synphot** package. These data files are not included with the STSDAS software and must be retrieved independently. To do this, you may either retrieve the Synphot data online at www.stsci.edu/resources/software_hardware/stsdas/synphot or you can retrieve a series of compressed tar files from the STScI FTP site in the directory `software/stsdas/refdata/synphot`.

After uncompressing and extracting the tar files (see below), you need to unpack the FITS files as described below.

The synthetic photometry data are read in similar way as the instrument datasets, using the script `unpack.c1` provided in the top directory. This script is run within IRAF to convert data from FITS format into the format used by the **synphot** task. This script assumes you have the logical `crrefer` set up in your `extern.pkg` file (which is in the directory

\$iraf/unix/hlib or have it set up in your session. You do this by placing the command below in `extern.pkg` or by typing it on the command line:

```
set crrefer = "/node/partition/stdata/synphot/"
```

Figure shows how to convert the files.

Unpacking Synthetic Photometry Files

```
% cl
cl> cd /node/partition/stdata/synphot
cl> set crrefer = "/node/partition/stdata/synphot/"
cl> task $unpack = unpack.cl
cl> tables
ta> fitsio
fi> unpack
```

Just in case... ←

← The "\$" is used because the task has no parameter file

Alternatively you may retrieve the Synphot data online at:

http://www.stsci.edu/resources/software_hardware/stsdas/synphot

and follow the instructions for downloading and installing.



Note that all five synphot files must be unloaded for the script to complete successfully.

A.3.3 Extracting the Synphot Unix Tar Files

If you retrieved the **synphot** database as compressed tar files, you will need to copy them to an appropriate subdirectory and then expand and unpack the files. The `tar` and `compress` utilities that do this are commonly available on most Unix systems, but are not standard in the VMS environment. The examples shown below reflect Unix usage. If you are on a VMS system, you should consult with your systems support staff regarding the availability and usage of these commands. To process the files on a Unix system:

1. Get the compressed tar file that you want, as described in previous sections.
2. Make an appropriate subdirectory using the `mkdir` command.
3. Pipe the compressed tar file through the `uncompress` and `tar` files to expand and unpack the file.

Appendix: A-18 ■ Getting IRAF and STSDAS

The following example shows how to do this. The example assumes that you are putting the files in a subdirectory under `/usr/iraf/stdata` (note that the name of your file here is assumed to be `XXX.tar.Z`).

```
% pwd
/usr/iraf/stdata
% mkdir XXX
% mv XXX.tar.Z XXX/
% cd XXX
% cat XXX.tar.Z | uncompress | tar -xf -
```

HST File Names

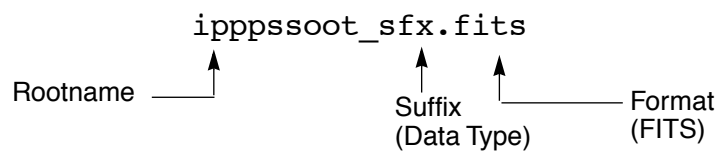
In this appendix. . .

B.1 Rootnames / B-3

B.2 Suffixes of Files Common to all Instruments / B-4

B.3 Associations / B-5

This appendix describes the syntax of HST data file names, which encode a large amount of information about the files themselves. Datasets retrieved from the Archive as described in Chapter 1 of the HST Introduction (Part I) consist of multiple files in FITS format, each with a name that looks like this:



- **Rootname:** The first part of the file name (`ippssoot`) is the *root-name* of the dataset to which the file belongs. All files belonging to a given dataset share the same rootname.
- **Suffix:** The three-character second part of the name (`sfx`) is called the *suffix*, and it indicates the type of data the file contains.
- **Format:** The identifier `.fits` indicates that this file is in FITS format.

For example, an FOC data file named `x3180101t_d0f.fits` is a FITS file belong to the dataset with rootname `x3180101t`, and its suffix `d0f` indicates that it contains raw science data.

In order to use IRAF/STSDAS tasks to work with data from instruments other than NICMOS, STIS, or ACS, you will want to convert these FITS files into GEIS format. See Section 2.1 in the HST Introduction for instructions on how to convert FITS files to GEIS files using `strfits`. Like FITS files, the names of GEIS files also derive from a file's rootname and suffix, and they look like this:

```
ippssoot.sfx
```

Generally the suffixes of GEIS files end either in “d”, indicating a binary data file, or “h”, indicating an ASCII header file. The two GEIS files `x3180101t_d0h` and `x3180101t_d0d` together contain the same information as the single FITS file `x3180101t_d0f.fits`.



The identifier referred to here as a “suffix” has often been called an “extension” in the past. However, the individual pieces of FITS files are also known as “extensions” (see Section 2.1.1 in the HST Introduction). For clarity, this handbook will use the term “extension” when referring to a component of a FITS file and the term “suffix” when referring to the three character identifier in a filename.

B.1 Rootnames

Rootnames of HST data files follow the naming convention defined in Table B.1.

Table B.1: IPPPSSOOT Root File Names

Character	Meaning
I	Instrument used, will be one of: <i>E</i> - Engineering data <i>F</i> - Fine Guidance Sensors <i>J</i> - Advanced Camera for Surveys <i>N</i> - Near Infrared Camera and Multi-Object Spectrograph <i>O</i> - Space Telescope Imaging Spectrograph <i>S</i> - Engineering subset data <i>T</i> - Guide star position data <i>U</i> - Wide Field/Planetary Camera-2 <i>V</i> - High Speed Photometer <i>W</i> - Wide Field/Planetary Camera <i>X</i> - Faint Object Camera <i>Y</i> - Faint Object Spectrograph <i>Z</i> - Goddard High Resolution Spectrograph
PPP	Program ID; can be any combination of letters or numbers (46,656 combinations possible). There is a unique association between program ID and proposal ID.
SS	Observation set ID; any combination of letters or numbers (1,296 possible combinations).
OO	Observation ID; any combination of letters or numbers (1,296 possible combinations).
T	Source of transmission or association product number <i>M</i> - Merged real time and tape recorded <i>N</i> - Retransmitted merged real time and tape recorded <i>O</i> - Retransmitted real time (letter 'O') <i>P</i> - Retransmitted tape recorded <i>Q</i> - Solid-state recorder <i>R</i> - Real time (not recorded) <i>S</i> - retransmitted solid-state recorder <i>T</i> - Tape recorded <i>0</i> - Primary association product (number zero) <i>I-8</i> - NICMOS or ACS background association product

B.2 Suffixes of Files Common to all Instruments

The three-character suffix of a data file (e.g., d0h) identifies the type of data that a file contains. Because the meanings of these suffixes change from instrument to instrument, please refer to the appropriate instrument-specific Data Structures chapter for their definitions. Several types of file suffixes are, however, common to all instruments.

OMS Files

Observatory Monitoring System (OMS) files were replaced by the Engineering Data Processing System (EDPS) after February 2003. Observations before February 2003 have OMS files. OMS files with suffixes `cm*` or `ji*`, contain Observation Logs describing how the HST spacecraft behaved during a given observation. OMS headers, which you can read with the IRAF task **imheader** (see Section 2.2.3 in the HST Introduction), are divided into groups of keywords that deal with particular topics such as spacecraft data, background light, pointing control data, and line-of-sight jitter summary. The headers themselves provide short descriptions of each keyword. OMS tables and images record spacecraft pointing information as a function of time. For more information on OMS files, you can consult Appendix C or the STScI Observation Logs Web pages at:

http://www.stsci.edu/hst/observatory/pointing/obslog/OL_1.html

PDQ Files

The suffix `pdq` denotes Post Observation Summary and Data Quality Comment files which contain predicted as well as actual observation parameters extracted from the standard header and science headers. These files may also contain comments on any obvious features in the spectrum or image, as noted in the OPUS data assessment, or automatically extracted information about problems or oddities encountered during the observation or data processing. These comments may include correction to the keywords automatically placed in the OMS files. PDQ were discontinued on May 9, 2002.

OCX Files

The suffix `ocx` denotes Observer Comment Files which are produced by STScI personnel to document the results of real-time commanding or monitoring of the observation, along with keywords and comments. Prior to April 17, 1992, OCX files were not always archived separately and, in some cases, were prepended to the trailer file.

After early February 1995, OCX files were produced only when an observation was used to locate the target for an Interactive Target Acquisition. At this time, mission and spacecraft information were moved

to the PDQ reports and the Observation Logs (OMS jitter image and jitter table). OCX files were also discontinued on May 9, 2002.

Trailer Files

Trailer files (suffix `tr1`) are FITS ASCII tables that log the processing of your data by the OPUS pipeline.



Note that trailer files are formatted with 132 columns

EDPS Files (post February 19, 2003)

These files contain Observation Logs describing how the HST spacecraft behaved during a given observation. These files replaced the OMS files after February 19, 2003.

B.3 Associations

The NICMOS, STIS, and ACS calibration pipelines sometimes produce single calibrated images from *associations* of many exposures. These associations allow HST pipeline processing to proceed further than it has in the past. For example, a NICMOS or ACS observer might specify a dithering pattern in a Phase II proposal. NICMOS or ACS would then take several exposures at offset positions, and the pipeline would combine them into a single mosaic (suffix `mos`). In this case, the original set of exposures constitutes the association, and the mosaic is the *association product*. Similarly, a STIS observer might specify a CR-SPLIT sequence in a Phase II proposal. STIS would gather several exposures at the same pointing, and the STIS pipeline would process this association of exposures into single image, free of cosmic rays, that would be the association product (suffix `crj`).

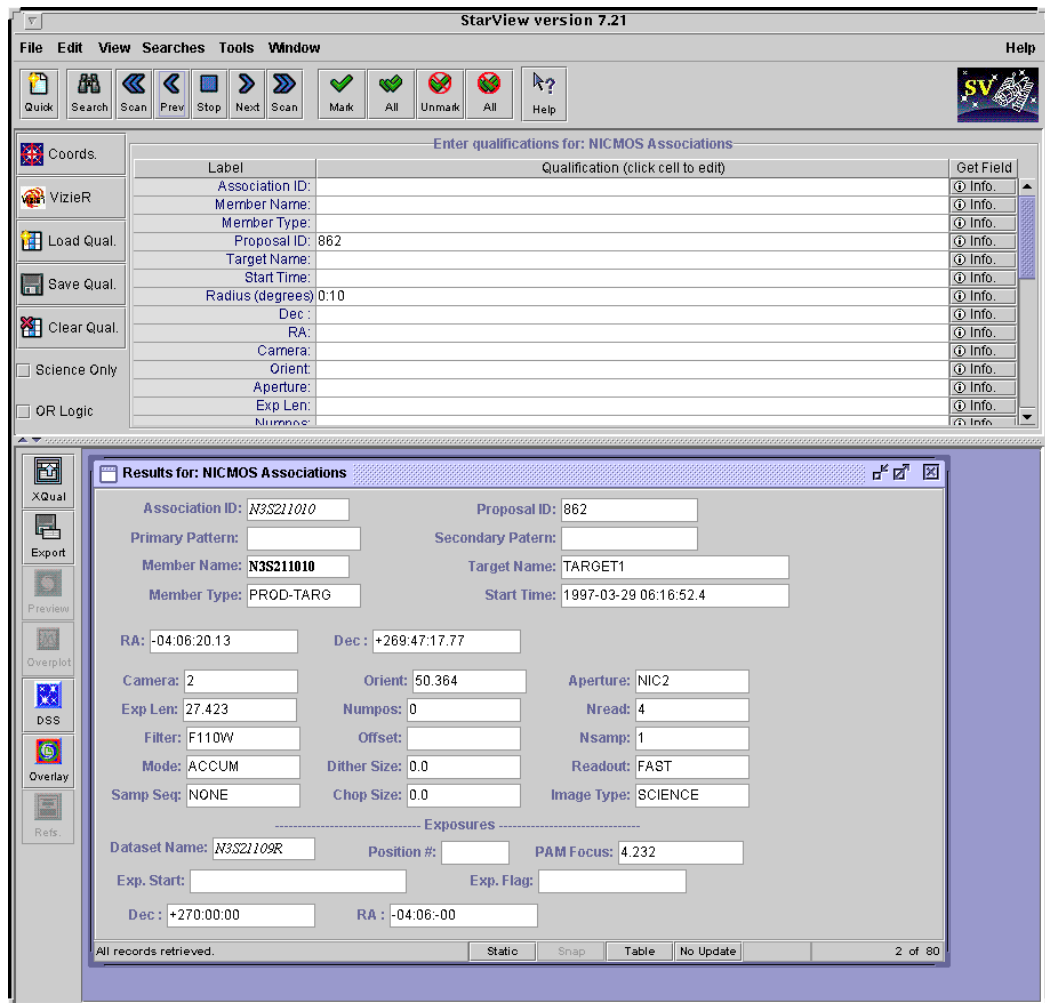
When you search the Archive with StarView for observations involving associations of exposures, your search will identify the final association product. The rootnames of association products always end in zero (see Table B.1 above.) If you request both Calibrated and Uncalibrated data from the Archive, you will receive both the association product and the exposures that went into making it. The corresponding association table, located in the file with suffix `asn` and the same rootname as the association product, lists the exposures belonging to the association. You can read this file using the STSDAS `tprint` or `tread` tasks (see Section 2.1.2 in the HST Introduction). The exposure IDs in the association table share the same `ippsss` sequence as the association rootname, followed by a base 36

number nn ($n = 0-9, A-Z$) that uniquely identifies each exposure, and a character t that denotes the data transmission mode (see Table B.1).

In practice, STIS stores the exposures belonging to associations differently than NICMOS or ACS. The exposures belonging to a STIS association all reside in the same file, while the exposures belonging to a NICMOS or ACS association reside in separate datasets. See the relevant Data Structures chapters for more details.

Information on the exposures belonging to an association is also available through StarView (see chapter 1 of the HST Introduction). From the <Welcome> Screen, click on [HST Instrument Searches] to get the <HST Instruments> screen, and then click on the [Associations] button for the instrument of interest. You can then search for the various exposures belonging to an association by entering the rootname of the association in the Association ID field and clicking on [Search]. An Association Results Screen will display the results of the search, which you can step through using the [Scan], [Prev], [Next] buttons. Figure B.1 below gives an example of a NICMOS Association Results Screen. Note the differences between the association rootname and coordinates and those of the individual exposure.

Figure B.1: Association Results Screen from StarView



Observation Logs

In this appendix . . .

<p>C.1 Observation Log Files / C-1 C.2 Retrieving Observation Logs / C-11 C.3 Using Observation Logs / C-12</p>

This Appendix describes the *Observation Log Files*, also known as OMS or *jitter* files. These files record pointing, jitter, and other Pointing Control System (PCS) data taken during an HST observation. You can use them to assess the behavior of the HST spacecraft during your observation, and in particular, to evaluate the jitter of the spacecraft while it was taking data. Here we describe the contents and structure of the observation log files, how to retrieve them from the Archive, and how to work with the data they contain.

C.1 Observation Log Files

Observation log files associated with each HST dataset contain pointing and specialized engineering data taken during the observation. These data files are produced by the Engineering Data Processing System (EDPS), an automated software system that interrogates the HST engineering telemetry and correlates the time-tagged engineering stream with HST's Science Mission Schedule (SMS), the seven-day command and event list that drives all spacecraft activities. The EDPS replaced the Observatory Monitoring System (OMS) in February 2003. EDPS provides observers with information about guide star acquisition, pointing, and tracking that is not normally provided in the science headers.

The observation log files share the same rootname as the observation they are associated with, except for the final character, which for observation log files is always a “j” (see appendix B for more on the names of HST data files). When OMS was installed in October 1994, it initially generated files with the suffixes `cmh`, `cmj`, `cmi`, which contained header information, high time resolution pointing data, and three-second average pointing data, respectively (see Table C.1). OMS observation logs changed to the `jih/jid/jif` image format after August 1995, at which time the `cmi` table was renamed `jit` to keep the naming convention consistent. In the OMS version of August 1995, `cmj` tables were replaced with a jitter image, which is a two-dimensional histogram of jitter excursions during the observation. The suffixes of the GEIS jitter image are `jih` for the header and `jid` for the image data. The `jit` table accompanies the jitter image. The header file of the image replaces the `cmh` file but includes the same information with the addition of some image-related keywords.



A detailed description of the old observation log files can be found on-line: http://www.stsci.edu/hst/observatory/pointing/obslog/OL_1.html, but for EDPS files, a description of the new Jitter File Format can be found on-line: http://www.ess.stsci.edu/projects/edps/jitter_format.html

Table C.1: OMS Observation Log Files

Suffix	Contents
October 1994 to August 1995	
<code>cmh</code>	OMS header
<code>cmj</code>	High time resolution (IRAF table)
<code>cmi</code>	Three-second averages (IRAF table)
<code>_cmh.fits</code>	Archived FITS file of <code>cmh</code>
<code>_cmj.fits</code>	Archived FITS file of <code>cmj</code>
<code>_cmi.fits</code>	Archived FITS file of <code>cmi</code>
August 1995 to February 1997	
<code>jih/jid</code>	Two-dimensional histogram and header (GEIS)
<code>jit</code>	Three-second averages (IRAF table) ¹
<code>_jif.fits</code>	Archived FITS file which bundles the <code>jih/jid</code> files.
<code>_jit.fits</code>	Archived FITS file of <code>jit</code> .
February 1997 onward	
<code>_jif.fits</code>	Two-dimensional histogram (FITS)
<code>_jit.fits</code>	Three-second averages table (FITS)
<code>_jwf.fits</code>	Two-dimensional histogram for STIS wavecal.
<code>_jwt.fits</code>	Three-second averages for STIS wavecal.

1. After May 11, 1995, the `jit` tables for exposures shorter than 6 seconds contain higher-resolution, one-second average pointing data.



Pointing and tracking information prior to October 1994 is not routinely available. Interested observers with data from this epoch, can send E-mail to help@stsci.edu.

C.1.1 Observation Log File Contents (October 1994 version)

Observation logs created between October 1994 and August 1995 contain:

- *rootnamej.cmh*: This ASCII header file contains the time interval, the rootname, averages of the pointing and spacecraft jitter, the guiding mode, guide star information, and alert or failure keywords. Figure C.1 shows a representative observation log header file.
- *rootnamej.cmj*: This table presents the data at the highest time resolution for the telemetry mode in use. It contains the reconstructed pointing, guide star coordinates, derived jitter at the instrument aperture, and guiding-related flags. The intent is: (1) to provide high-time resolution jitter data for deconvolution or for assessing small aperture pointing stability, and (2) to display the slew and tracking anomaly flags with the highest resolution. Table C.2 lists the table column heading, units and a brief definition.
- *rootnamej.cmi*: This table contains data that were averaged over three-second intervals. It includes the same information as the *.cmj* table and also includes orbital data (e.g., latitude, longitude, limb angle, magnetic field values, etc.) and instrument-specific items. It is best suited for a quick-look assessment of pointing stability and for studying trends in telescope or instrument performance with orbital environment. Table C.3 lists the table column heading, units and a brief definition.
- *rootnamej_cmi/j/h.fits*: The above three GEIS files are actually archived as FITS files. They may be worked with as such, or run through the STSDAS task **strfits**, to convert them.
- Observation Log File Contents (August 1995 version)
- The contents of observation log files created between August 1995 and February 1997 are as follows:

- *rootnamej.jih*: This GEIS header file, the analog to the *cmh* file, contains the time interval, the rootname, averages of the pointing and spacecraft jitter, the guiding mode, guide star information, and alert or failure keywords. Figure C.1 shows a representative observation log header file.
- *rootnamej.jid*: This GEIS image—a significant enhancement of the old *cmj* file—presents a two-dimensional histogram of the pointing fluctuations during the observation. You can display it to visualize the spacecraft stability during your observation, and is information for deconvolutions and PSF analyses.
- *rootnamej.jit*: This table, the analog to the *cmi* table, contains data that were averaged over three-second intervals. Its content is identical (see Table C.3).
- *rootnamej_jif.fits*: FITS file that is actually the de-archived product. This file can be converted to the *jih/jid* GEIS file via the **strfits** routine.
- *rootnamej_jit.fits*: The de-archived FITS file corresponding to the *jit* IRAF table. It can be converted via **strfits**.

C.1.2 Observation Log File Contents (February 1997 version)

The contents of observation log files created since February 1997 are as follows:

- *rootnamej_jif.fits*: The de-archived FITS file. Unlike the previous OMS epoch, this FITS file does not bundle a GEIS file and cannot be converted with **strfits**. This was done to more closely correlate the observation log files with the NICMOS, STIS, and ACS FITS files with extensions and associations. OMS will normally put all associated observation logs into a single file, to correspond to the associated science exposures. However, if even one science exposure is orphaned (not associated) then an individual observation log FITS file will be produced for every exposure in that association. For a description of NICMOS, STIS, and ACS association files, see appendix B. All of the information contained in the old *cmh/jih* ASCII header is now available as keywords in the FITS files.
- *rootnamej_jit.fits*: The FITS file containing the table information. The comments for the *_jif* file apply here as well.

Table C.2: Contents of .cmj Table

Parameter	Units	Description
seconds	seconds	Time since window start
V2 dom	arcseconds	Dominant FGS V2 coordinate
V3 dom	arcseconds	Dominant FGS V3 coordinate
V2 roll	arcseconds	Roll FGS V2 coordinate
V3 roll	arcseconds	Roll FGS V3 coordinate
SI V2	arcseconds	Jitter at aperture reference
SI V3	arcseconds	Jitter at aperture reference
RA	degrees	Right ascension of aperture reference
DEC	degrees	Declination of aperture reference
Roll	degrees	Angle between North and +V3
DayNight	0,1 flag	Day (0) or night (1)
Recenter	0,1 flag	Recentering status
TakeData	0,1 flag	Vehicle guiding status
SlewFlag	0,1 flag	Vehicle slewing status

Figure C.1:A Representative .jih or .cmh Header

```

dSIMPLE = F / data conforms to FITS standard !
BITPIX = 32 / bits per data value !
DATATYPE= 'INTEGER*4' / datatype of the group array !
NAXIS = 2 / number of data axes !
NAXIS1 = 64 / length of the 1st data axis !
NAXIS2 = 64 / length of the 2nd data axis !
GROUPS = T / image is in group format !
GCOUNT = 1 / number of groups !
PCOUNT = 0 / number of parameters !
PSIZE = 0 / bits in the parameter block !
OMS_VER = '16.2C' / OMS version used to process this observation !
PROCTIME= '1994.133:06:24:18.35' / date-time OMS processed observation !
/ date-times format (yyyy.ddd:hh:mm:ss.ss)

/ IMAGE PARAMETERS
CRVAL1 = 0.0 / right ascension of zero-jitter pixel (deg)
CRVAL2 = 0.0 / declination of zero-jitter pixel (deg)
CRPIX1 = 32 / x-coordinate of zero-jitter pixel
CRPIX2 = 32 / y-coordinate of zero-jitter pixel
CTYPE1 = 'RA---TAN' / first coordinate type
CTYPE2 = 'DEC--TAN' / second coordinate type
CD1_1 = 0.0 / partial of ra w.r.t. x (deg/pixel)
CD1_2 = 0.0 / partial of ra w.r.t. y (deg/pixel)
CD2_1 = 0.0 / partial of dec w.r.t. x (deg/pixel)
CD2_2 = 0.0 / partial of dec w.r.t. y (deg/pixel)
COORDSYS= 'WFPC2' / image coordinate system
XPIXINC = 2.0 / plate scale along x (mas per pixel)
YPIXINC = 2.0 / plate scale along y (mas per pixel)
PARITY = -1 / parity between V2V3 frame and image frame
BETA1 = 134.72 / angle from +V3 to image +x (toward +V2)
BETA2 = 224.72 / angle from +V3 to image +y (toward +V2)

/ OBSERVATION DATA
PROPOSID= 05233 / PEP proposal identifier
PROGRMID= '288' / program id (base 36)
OBSET_ID= '02' / observation set id
OBSRVTN= '03' / observation number (base 36)
TARGNAME= 'NGC3379-PO' / proposer's target name
STARTIME= '1994.133:06:24:18.35' / predicted observation window start time
ENDTIME = '1994.133:06:39:18.35' / predicted observation window end time
SOGSID = 'U2880203' / SOGS observation name !

/ SCIENTIFIC INSTRUMENT DATA
CONFIG = 'WFPC2' / proposed instrument configuration
PRIMARY = 'SINGLE' / single, parallel-primary, parallel-secondary
OPERATE = '1994.133:06:22:46.91' / predicted time instr. entered operate mode
TLMFORM = 'PN' / telemetry format
APER_TURE= 'UWFALL' / aperture name
APER_V2 = 1.565 / V2 aperture position in vehicle frame (arcsec)
APER_V3 = 7.534 / V3 aperture position in vehicle frame (arcsec)

/ SPACECRAFT DATA
ALTITUDE= 593.23 / average altitude during observation (km)
LOS_SUN = 106.08 / minimum line of sight to Sun (deg)
LOS_MOON= 77.11 / minimum line of sight to Moon (deg)
SHADOENT= '1994.133:05:11:29.00' / predicted Earth shadow last entry
SHADOEXT= '1994.133:05:42:45.00' / predicted Earth shadow last exit
LOS_SCV = 12.46 / minimum line of sight to S/C veloc. (deg)
LOS_LIMB= 58.0 / average line of sight to Earth limb (deg)

/ BACKGROUND LIGHT
ZODMOD = 22.3 / zodiacal light - model (V mag/arcsec2)
EARTHMOD= 20.2 / peak Earth stray light - model (V mag/arcsec2)
MOONMOD = 35.5 / moon stray light - model (V mag/arcsec2)
GALACTIC= -1.0 / diffuse galactic light - model (V mag/arcsec2)

/ POINTING CONTROL DATA
GUIDECMD= 'FINE LOCK' / commanded guiding mode
GUIDEACT= 'FINE LOCK' / actual guiding mode at end of GS acquisition
GSD_ID = '0084900235' / dominant guide star id
GSD_RA = 161.70720 / dominant guide star RA (deg)
GSD_DEC = 12.45407 / dominant guide star DEC (deg)

```

Figure C.2: Representative .jih or .cmh Header

```

GSD_MAG =          12.867 / dominant guide star magnitude
GSR_ID   = '0085201189'   / roll guide star id
GSR_RA   =          161.93314 / roll guide star RA (deg)
GSR_DEC  =          12.78141 / roll guide star DEC (deg)
GSR_MAG  =           12.977 / roll guide star magnitude
GSACQ    = '1994.133:06:31:02.92' / actual time of GS acquisition completion
PREDGSSEP=          1420.775 / predicted guide star separation (arcsec)
ACTGSSEP=          1421.135 / actual guide star separation (arcsec)
GSSEPRMS=           3.8 / RMS of guide star separation (milli-arcsec)
NLOSSES  =           0 / number of loss of lock events
LOCKLOSS=           0.0 / total loss of lock time (sec)
NRECENT  =           0 / number of recentering events
RECENTR  =           0.0 / total recentering time (sec)

                                     / LINE OF SIGHT JITTER SUMMARY
V2_RMS   =           4.5 / V2 axis RMS (milli-arcsec)
V2_P2P   =           51.6 / V2 axis peak to peak (milli-arcsec)
V3_RMS   =           20.9 / V3 axis RMS (milli-arcsec)
V3_P2P   =           267.3 / V3 axis peak to peak (milli-arcsec)
RA_AVG   =          161.85226 / average RA (deg)
DEC_AVG  =          12.58265 / average dec (deg)
ROLL_AVG =          293.01558 / average roll (deg)

                                     / PROBLEM FLAGS, WARNINGS and STATUS MESSAGES
                                     / (present only if problem exists)
ACQ2FAIL= '          ' T' / target acquisition failure
GSFAIL   = 'DEGRADED' ' / guide star acquisition failure (*1)
TAPEDROP= '          ' T' / possible loss of science data
TLM_PROB= '          ' ' / problem with the engineering telemetry
TM_GAP   =          404.60 / duration of missing telemetry (sec)
SLEWING  = '          ' T' / slewing occurred during this observation
TAKEDATA= '          ' F' / take data flag NOT on throughout observation
SIPROBnn= '          ' ' / problem with specified science instrument (*2)

END

```

notes

*1 - GSFAIL appears only once in a single header file.
The following table lists all current possible values for the GSFAIL keyword:

GSFAIL	DESCRIPTION	COMMENT
DEGRADED		/ guide star acquisition failure
IN PROGR		/ guide star acquisition failure
SSLEXP		/ guide star acquisition failure
SSLEXS		/ guide star acquisition failure
NOLOCK		/ guide star acquisition failure
SREXCS?		/ guide star acquisition failure
SREXCS1		/ guide star acquisition failure
SREXCS2		/ guide star acquisition failure
SREXCS3		/ guide star acquisition failure
SREXCP?		/ guide star acquisition failure
SREXCP1		/ guide star acquisition failure
SREXCP2		/ guide star acquisition failure
SREXCP3		/ guide star acquisition failure
UNKNOWN		/ guide star acquisition failure
VEHSAFE		/ guide star acquisition failure

*2 - The SIPROBnn keywords appear in the header file with nn = 01 - 99.
The following table lists all current possible values for the SIPROBnn keyword:

SIPROBnn	DESCRIPTION	COMMENT
DCF_NUM unchanged		/ This observation may not have been taken
FOS Safing!		/ This observation affected when FOS Safed!
HRS Safing!		/ This observation affected when HRS Safed!
WFII Safing!		/ This observation affected when WFII Safed!
FOC Safing!		/ This observation affected when FOC Safed!
Shut		/ FOS aperture door is not Open!
FAILED		/ FGS astrometry target acquisition failed

Appendix: C-8 ■ Observation Log Files

Table C.3: Contents of .jit or .cmi Table, Three-Second Averaging

Parameter	Units	Description
seconds	seconds	Time since window start
V2 dom	arcseconds	Dominant FGS V2 coordinate
V3 dom	arcseconds	Dominant FGS V3 coordinate
V2 roll	arcseconds	Roll FGS V2 coordinate
V3 roll	arcseconds	Roll FGS V3 coordinate
SI V2 AVG	arcseconds	Mean jitter in 3 seconds
SI V2 RMS	arcseconds	rms jitter in 3 seconds
SI V2 P2P	arcseconds	Peak jitter in 3 seconds
SI V3 AVG	arcseconds	Mean jitter in 3 seconds
SI V3 RMS	arcseconds	rms jitter in 3 seconds
SI V3 P2P	arcseconds	Peak jitter in 3 seconds
RA	degrees	Right ascension of aperture reference
DEC	degrees	Declination of aperture reference
Roll	degrees	Angle between North and +V3
LimbAng	degrees	Angle between earth limb and target
TermAng	degrees	Angle between terminator and target
LOS_Zenith	degrees	Angle between HST zenith and target
Latitude	degrees	HST subpoint latitude
Longitude	degrees	HST subpoint longitude
Mag V1,V2,V3	degrees	Magnetic field along V1, V2, V3
EarthMod	V Mag/arcsec ²	Model earth background light
SI_Specific	–	Special science instrument data
DayNight	0,1 flag	Day (0) or night (1)
Recenter	0,1 flag	Recentering status
TakeData	0,1 flag	Vehicle guiding status
SlewFlag	0,1 flag	Vehicle slewing status

C.1.3 Jitter File Contents (February 2003 Version)

The current format of the jitter files produced by the EDPS are similar to the Observation Logs (Obslogs) produced by the Observation Monitoring System (OMS) that no longer exist. The EDPS jitter files are limited to the engineering data that describes the performance of the Pointing Control System (PCS) including the Fine Guidance Sensors that are used to control the vehicle pointing. The jitter files report on PCS engineering data for the duration of the observation. The old name for these jitter files, observation logs (Obslogs), is now inappropriate, because of the absence of instrument monitoring. However the FITS header and extension formats of the jitter files are not radically different from the original Obslogs. FITS-based tools that access only the PCS related data should be able to handle both the old Obslogs and the new jitter files. The jitter files retain the same file naming conventions as the old Obslogs.

One way to distinguish the new jitter files from the old Obslogs files is to check the existence of the OPUS_VER keyword (system version id) that replaced the OMS_VER keyword. The EDPS software is part of the larger OPUS system that also supports the Science Data Processing System (SDPS) and the On-the-Fly Reprocessing (OTFR) system. Most of the format changes involve deletions of keywords. Six keywords have been added. There are some improvements in the new files; the accuracy of HST orbit-related statistics has been improved by ignoring the values in the telemetry that must be interpolated. Instead, an orbit model is now used to recalculate the orbit position and velocity at the exact time needed. The new jitter files have been generated since mid-February, 2003, but there is no specific date since the OMS ran in parallel with EDPS for about a week.

The old format is well documented by the Observatory Support Web page maintained by the Telescopes Branch of the Instrument Division at <http://www.stsci.edu/instruments/observatory/>

The differences between the new jitter files and the old observation log files are briefly described below. This jitter file information is supplemental to the Observation Log Documentation. For further details on the differences, along with sample file headers, see the following Web page that describes these files at length:

http://www.ess.stsci.edu/gsd/dst/edps/jitter_format.html

Changes in Image File

There are six new keywords for the jitter image file that has the file extension “.jif” or “.jwf”. Keywords that are in the primary header are assigned the extension (Ext) value “0”. Keywords in other extensions have the extension value “>0” since there can be more than one extension for associated products.

Keyword	Ext	Comment
ASN_PTYP	0	Association Product Type
GSD_FGS	0	1, 2 or 3: FGS ID of dominant FGS
GSR_FGS	0	1,2 or 3: FGS ID of subdominant (roll) FGS
GS_PRIM	0	DOMINANT or ROLL: GS that was acquired first
OPUS_VER	0	Build version: replaces OMS_VER
WCSAXES	>0	2: Number of axes for WCS parameters

There were 43 OMS Obslog keywords that were dropped for the new jitter image file.

Changes in Table File

The jitter table file that has the file extension “.jit” or “.jwt” has very few changes. There are still 28 columns of data in the table. Two of the columns have been replaced. Column 23 has changed from EarthMag, that was an estimate of stray light from the Earth based on a model that is no longer supported by EDPS, to BrightLimb, which is a short integer value, where 1 indicated that the limb position closest to the target is bright, and 0 indicates it is dark. Column 24 has changed from SI_Specific, that is not supported by EDPS, to FGS_flags, that provides status data for all three FGS. In addition, the comment for the DayNight flag has been corrected. The day value is 1 and the night value is 0. These changes affected the values of the keywords: TTYPE23, TUNIT23, TFORM23, TUNIT24, TTYPE24, and TUNIT24. The value of keyword NAXIS1 changed from 116 to 114 because the size of column 23 changed from a four-byte floating value to a two-byte short integer value. The keyword ASN_PTYP was added to the association keywords. It has the same value as that keyword in the image file header.

Changes in Association File Structure

In the old Obslogs, the jitter files for associated exposures were always collected into files with the same filename extension, “.jit”, “.jwt”, “.jif”, or “.jwf”, as the member exposures. The file name has the association rootname that is found in keyword ASN_ID. Each member exposure provides a FITS extension for a table or image, having the keyword ASN_MTYPE that describes the type of member. The primary header of the association image file has the following keywords updated to reflect statistics for the set of FITS extensions: TSTRTIME, TENDTIME, MGSSPRMS, TNLOSSES, TLOCKLOS, TNREENT, TRECENTR, TV2_RMS, MV2_P2P, TV3_RMS, MV3_P2P, TRA_AVG, TROLLAVG,

T_GDACT, T_ACTGSP, T_GSFAIL, T_SGSTAR, T_TLMPRB, T_NOTLM, T_NTMGAP, T_GSGAP, T_SLEWING, and T_TFDDWN.

For STIS associations, the wavecal members are separated from the science members by different file extensions: “.jwt” and “.jwf”. The structure for STIS associations has not changed. But for NICMOS and ACS, the file rootnames now match the rootnames of the association products. Only the members that are used to generate the association product are collected into FITS extensions for that product. The old Obslogs used only the association rootname. The new jitter-file associations may have multiple rootnames, ending in 0, 1, 2, etc. The statistical keywords in the primary header listed above are calculated for the subset of members collected for the specific association product. The ACS associations reserve the rootname of the ASN_ID for dithered data combined from multiple positions. At each position, there is an association product having the last character of its rootname being a sequence starting from 1. If there is no dithered product for ACS, the associated exposures at the single position have a product with a rootname ending in 1.

The Data Systems Branch maintains a Web page that allows access to detailed information on the most recent header formats and keyword definitions. This Web page can be accessed from the HST Keyword Dictionary entry found on the DSB home page at

<http://www.ess.stsci.edu/gsd/dst/>

C.2 Retrieving Observation Logs

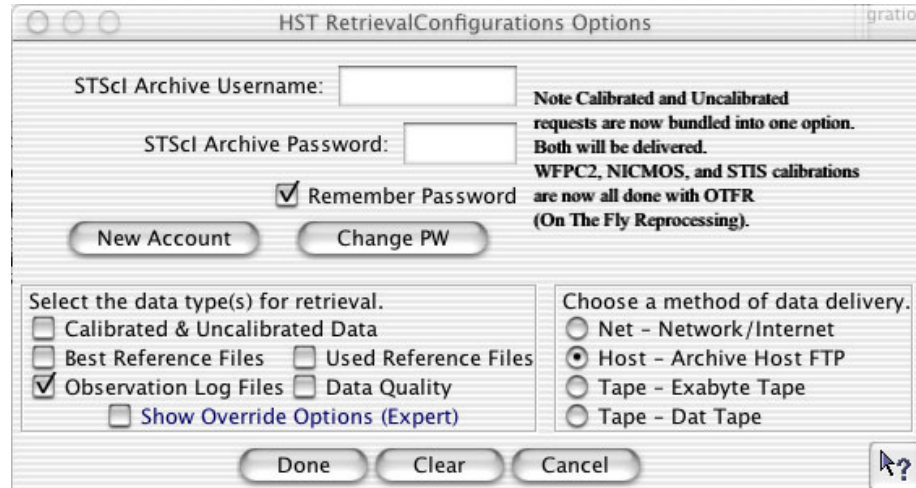
You can retrieve observation log files for data taken after October 20, 1994 from the HST Archive using StarView as described in chapter 1 of the HST Introduction. Unlike science data, which generally has a one-year proprietary period, observation log files become public as soon as they are archived.

The easiest way to get OMS files through StarView is to identify the observation of interest and proceed with your request as described in chapter 1 of the HST Introduction, until you reach the “HST Retrieval Configurations Options” screen, reproduced in Figure C.3. You can then check the Observation Log Files box, along with any other desired boxes, and continue with your request. StarView will then deliver the associated observation log files.

For observations logged between October 1994 to August 1995, you will be delivered the *cmi*, *cmj*, and *cmh* files in FITS form (e.g., *_cmi.fits*). Observations archived from August 1995 to February 1997 will return *_jif.fits* and *_jit.fits* files. These, and the earlier FITS files can be worked with as such, or converted to their GEIS counterparts via the STSDAS **strfits** task. However, as of February 1997, the

`_jif.fits` and `_jit.fits` files are standard FITS files with extensions and cannot be converted to GEIS.

Figure C.3: Choosing Observation Log Files in StarView



C.3 Using Observation Logs

Here are some simple examples of what can be learned from the observation log files. Note that for FITS format observation logs, current versions of STSDAS tools will handle the files with extensions properly. Keywords can be viewed with tools such as **imheader** or **hedit**, and data viewed, plotted, or displayed using the same tasks one might have for the GEIS files. For more information on FITS file structures, see chapter 2 of the HST Introduction.

C.3.1 Guiding Mode

Unless requested, all observations will be scheduled with FINE LOCK guiding, which may be one or two guide stars (dominant and roll). The spacecraft may roll slightly during an observation if only one guide star is acquired. The amount of roll depends upon the gyro drift at the time of the observation, the location during an orbit, and the lever arm from the guide star to the center of the aperture.

There are three commanded guiding modes: FINE LOCK, FINE LOCK/GYRO, and GYRO. OMS header keywords GUIDECMD (commanded guiding mode) and GUIDEACT (actual guiding mode) will usually agree. If there was a problem, they won't agree and the GUIDEACT value will be the guiding method actually used during the

exposure. If the acquisition of the second guide star fails, the spacecraft guidance, GUIDEACT, may drop from FINE LOCK to FINE LOCK/GYRO, or even to GYRO, which may result in a target rolling out of an aperture. Check the OMS header keywords to verify that there was no change in the requested guiding mode during the observation.



Until new flight software (version FSW 9.6) came online in September 1995, if the guide star acquisition failed, the guiding dropped to COARSE track. After September 1995, if the guide star acquisition failed, the tracking did not drop to COARSE track. Archival researchers may find older datasets that were obtained with COARSE track guiding.

The dominant and roll guide star keywords (GSD and GSR) in the OMS header can be checked to verify that two guide stars were used for guiding, or in the case of an acquisition failure, to identify the suspect guide star. The dominant and roll guide star keywords identify the stars that were scheduled to be used, and in the event of an acquisition failure, may not be the stars that were actually used. The following list of cmh keywords is an example of two star guiding.

```
GSD_ID = '0853601369      ' / Dominant Guide Star ID
GSD_RA =          102.42595 / Dominant Guide Star RA (deg)
GSD_DEC =         -53.41362 / Dominant Guide Star DEC (deg)
GSD_MAG =           11.251 / Dominant Guide Star Magnitude
GSR_ID = '0853602072      ' / Roll Guide Star ID
GSR_RA =          102.10903 / Roll Guide Star RA (deg)
GSR_DEC =         -53.77683 / Roll Guide Star DEC (deg)
GSR_MAG =           12.426 / Roll Guide Star Magnitude
```

If you suspect that a target has rolled out of the aperture during an exposure, you can quickly check the counts in each group of the raw science data. As an example, the following IRAF commands can be used to determine the counts in each group.

```
cl> grlist z2o4040dt.d0h 1-24 > groups.lis
cl> imstat @groups.lis
```

Some observations can span several orbits. If during a multiple orbit observation the guide star reacquisition fails, the observation may be terminated with possible loss of observing time, or switch to other less

desirable guiding modes. The GSACQ keyword in the cmh header will state the time of the last successful guide star acquisition.

```
GSACQ = '136:14:10:37.43' / Actual time of GS Acquisition Completion
```

C.3.2 Guide Star Acquisition Failure

The guide star acquisition at the start of the observation set could fail if the FGS fails to lock onto the guide star. The target may not be in the aperture, or maybe only a piece of an extended target is in the aperture. The jitter values will be increased because FINE LOCK was not used. The following list of cmh header keywords indicate that the guide star acquisition failed.

```
V3_RMS = 19.3 / V3 Axis RMS (milli-arcsec)
V3_P2P = 135.7 / V3 Axis peak to peak (milli-arcsec)
GSFAIL = 'DEGRADED' / Guide star acquisition failure!
```

The observation logs for all of the following observations in the observation set will have the “DEGRADED” guide star message. This is not a Loss of Lock situation but an actual failure to acquire the guide star in the desired guiding mode. For the example above, the guiding mode dropped from FINE LOCK to COARSE TRACK.

```
GUIDECMD= 'FINE LOCK' / Commanded Guiding mode
GUIDEACT= 'COARSE TRACK' / Actual Guiding mode at end of GS acquisition
```

If the observational dataset spans multiple orbits, the guide star will be re-acquired, but the guiding mode will not change from COARSE TRACK. In September 1995, the flight software was changed so that COARSE TRACK is no longer an option. The guiding mode drops from two guide star FINE LOCK to one guide star FINE LOCK, or to GYRO control.

C.3.3 Moving Targets and Spatial Scans

A type 51 slew is used to track moving targets (planets, satellites, asteroids, and comets). Observations are scheduled with FINE LOCK acquisition, i.e., with two or one guide stars. Usually, a guide star pair will stay within the pickle during the entire observation set, but if two guide stars are not available, a single guide star may be used, assuming the drift is small or the proposer says that the roll is not important for that particular observing program. An option during scheduling is to drop from FGS control to GYRO control when the guide stars move out of the FGS. Also, guide star handoffs (which are not a simple dropping of the guide stars to

GYRO control) will affect the guiding and may be noticeable when the jitter ball is plotted.

The jitter statistics are accumulated at the start of the observation window. Moving targets and spatial scan motion will be seen in the jitter data and image. Therefore, the OMS header keywords V2_RMS and V3_RMS values (the root mean square of the jitter about the V2 and V3 axis) can be quite large for moving targets. Also, a special anomaly keyword (SLEWING) will be appended to the OMS header stating movement of the telescope during the observation. This is expected for observing moving targets. The following list of .cmh header keywords is an example of expected values while tracking a moving target.

```

/ LINE OF SIGHT JITTER SUMMARY
V2_RMS =          3.2 / V2 Axis RMS (milli-arcsec)
V2_P2P =          17.3 / V2 Axis peak to peak (milli-arcsec)
V3_RMS =          14.3 / V3 Axis RMS (milli-arcsec)
V3_P2P =          53.6 / V3 Axis peak to peak (milli-arcsec)
RA_AVG = 244.01757 / Average RA (deg)
DEC_AVG = -20.63654 / Average DEC (deg)
ROLL_AVG= 280.52591 / Average Roll (deg)
SLEWING = '      T' / Slewing occurred during this observation

```

C.3.4 High Jitter

The spacecraft may shake during an observation, even though the guiding mode is FINE LOCK. This movement may be due to a micro-meteorite hit, jitter at a day-night transition, or for some other unknown reasons. The FGS is quite stable and will track a guide star even during substantial spacecraft motion. The target may move about in an aperture, but the FGS will continue to track guide stars and reposition the target into the aperture. For most observations, the movement about the aperture during a spacecraft excursion will be quite small, but sometimes, especially for observations with the spectrographs, the aperture may move enough that the measured flux for the target will be less than a previous group. Check the OMS header keywords (V2_RMS, V3_RMS) for the root mean square of the jitter about the V2 and V3 axis. The following list of .cmh header keywords is an example of typical guiding rms values.

```

/ LINE OF SIGHT JITTER SUMMARY
V2_RMS =          2.6 / V2 Axis RMS (milli-arcsec)
V2_P2P =          23.8 / V2 Axis peak to peak (milli-arcsec)
V3_RMS =          2.5 / V3 Axis RMS (milli-arcsec)
V3_P2P =          32.3 / V3 Axis peak to peak (milli-arcsec)

```

Recentering events occur when the spacecraft software decides that shaking is too severe to maintain lock. The FGS will release guide star control and within a few seconds reacquire the guide stars. It is assumed the guide stars are still within the FGS field of view. During the recentering

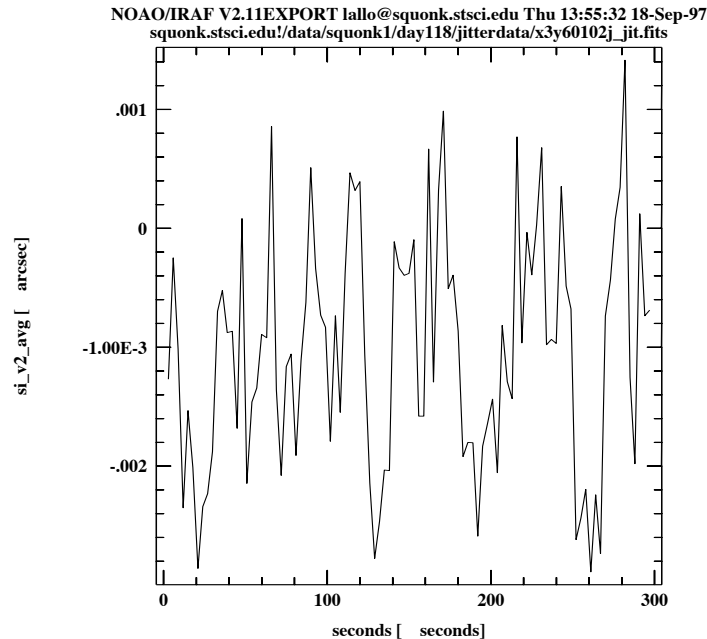
time, INDEF will be written to the OMS table. Recentering events are tracked in the OMS header file.

Be careful when interpreting “Loss of Lock” and “Recentering” events that occur at the very beginning or at the end of the OMS window. The OMS window is larger than the observation window. These events might not affect the observation since the observation start time will occur after the guide stars are acquired (or re-acquired), and the observation stop time may occur before the “Loss of Lock” or “Recentering” event that occurred at the end of an OMS window.

The **sgraph** command in the **stdas.graphics.stplot** package will plot time vs. jitter along the direction of HST’s V2 axis (see Figure C.4):

```
cl> sgraph "x3y60102j_jit.fits seconds si_v2_avg"
```

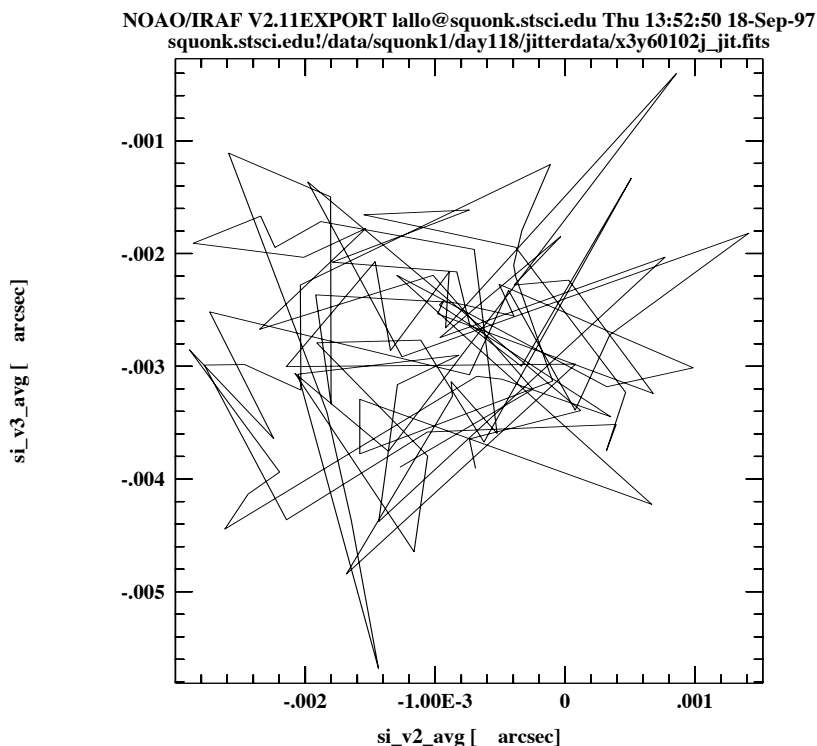
Figure C.4: Plotting Jitter Along V3 Axis



To get an idea of pointing stability, you can create a *jitter ball* by plotting jitter along the V2 axis vs. jitter along the V3 axis (see Figure C.5):

```
st> sgraph "x3660102j_jit.fits si_v2_avg si_v3_avg"
```

Figure C.5: Plotting V2 vs. V3 Jitter



The **tstatistics** task can be used to find the mean value of the `si_v3_avg` column—the amount of jitter (in arcseconds) in the direction of the V3. This value can be used to model jitter in a PSF. In this example, the mean jitter is ~ 3 mas, which is typical for post-servicing mission data:

Figure C.6: Averaging a Column with tstatistics

```
tt> tstat u26m0801j.cmi si_v3_avg
# u26m0801j.cmi si_v3_avg
#
nrows      mean      stddev      median      min      max
   11  -0.003006443888  0.00362533  -7.17163E-4  -0.00929515  0.00470988
```



Understanding and interpreting the meaning of the table columns and header keywords is critical to understanding the observation logs. Please read the available documentation and contact the STScI Help Desk (help@stsci.edu) if you have any questions about the files. Documentation is available via the Web at: <http://www.stsci.edu/hst/observatory/documents>

Index

A

accuracy
 astrometric, improving Intro:3—14
acquisition failure
 guide stars Appendix:C—14
ACS2D
 flow diagram ACS:3—9
 task, MAMA calibrations ACS:3—18
ACSCCD
 flow diagram ACS:3—8
 task, CCD calibrations ACS:3—10
ACSREJ
 task, cosmic-ray rejection ACS:3—15
ACSSUM
 task, image combination ACS:3—25
analog-to-digital conversion ACS:5—2
analysis
 images, general in STSDAS Intro:3—11
 spectra, general in STSDAS Intro:3—27
 spectra, general tasks in IRAF Intro:3—33
 spectra, STIS Intro:3—28
analysis package
 image analysis Intro:3—6
Aperture correction ACS:6—4
 encircled energies ACS:6—4
 red objects ACS:6—5
 scattered light halo ACS:6—4
Apertures
 detectors in V2/V3 frame ACS:1—6
apphot package
 aperture photometry Intro:3—19
archive
 file names Intro:2—1
arithmetic

 imset, msarith task Intro:3—17
 spectra, splot task Intro:3—35
array
 FITS table Intro:2—10
Association Tables
 applying user-defined shifts ACS:4—27
 creating custom tables ACS:4—22
 creating tables from scratch ACS:4—23
 merging tables ACS:4—22
 structure ACS:2—2
 use with PyDrizzle ACS:4—19
Astrometry
 Anderson and King correction ACS:6—24
 coordinate transformations ACS:6—23
 guide star failure impact ACS:6—25
astrometry
 basic, in STSDAS Intro:3—12
 improving accuracy Intro:3—14
 tasks in STSDAS Intro:3—14
B
background
 running tasks in Appendix:A—7
Bad Pixels
 data quality array ACS:3—11, ACS:3—19
 pixel flag values ACS:3—12
 reference file (BPIXTAB) ACS:3—11
bandpar Intro:3—21
Bias jump
 between quadrants ACS:5—4
 horizontal ACS:5—5
Bias subarrays ACS:5—6
Bias subtraction
 bias image (BIASFILE) ACS:3—13

- overscan subtraction table (OSCNTAB)
 - ACS:3—12
 - superbias ACS:5—2
- Blooming ACS:5—27
- Buffer memory ACS:1—8
- C**
- CALACS
 - flow diagram
 - CCD data ACS:3—8
 - general ACS:3—7
 - MAMA data ACS:3—9
 - gain correction ACS:3—21
 - individual tasks
 - ACS2D ACS:3—18
 - ACSCCD ACS:3—10
 - ACSREJ ACS:3—15
 - ACSSUM ACS:3—25
 - list of calibration steps ACS:3—6
 - memory requirements ACS:3—27
 - overview ACS:3—3
 - processing dithered exposures ACS:3—4
 - processing repeated exposures ACS:3—4
 - processing single exposures ACS:3—3
- Calibration
 - aXe software ACS:6—42
 - CCD ACS:3—10
 - coronagraphy ACS:6—32
 - flat field ACS:5—18
 - gain ACS:5—37
 - MAMA ACS:3—18
 - manual ACS:3—25
 - on-the-fly ACS:3—2
 - photometry ACS:6—1
 - pipeline ACS:3—9
 - polarimetry ACS:6—26, ACS:6—30
 - processing speed ACS:3—27
 - ramp filters ACS:6—35
 - shading correction ACS:5—46
 - software ACS:3—3
 - spectroscopy ACS:6—41
 - switch selection criteria ACS:3—26
- calibration
 - software, STSDAS Intro:3—1
- Capabilities
 - coronagraphy ACS:1—2
 - imaging ACS:1—1
 - polarimetry ACS:1—2
 - spectroscopy ACS:1—2
- Chip offset
 - WFC ACS:4—2
- Color transformation ACS:6—5
- command
 - see "task"
- commands
 - plot cursor Intro:3—36
- Context Image ACS:4—20, ACS:4—53
- conversion
 - counts to flux or magnitude Intro:3—19
 - flux to wavelength, resample task Intro:3—31
- Coronagraphy
 - flat fields ACS:6—33
 - ghosting ACS:6—35
 - observing considerations ACS:1—7
 - optical path ACS:1—4
 - photometry ACS:6—34
 - PSF ACS:6—34
- Cosmic-ray Rejection
 - ACSREJ task ACS:3—15
 - algorithm ACS:3—15
 - attached pixels ACS:5—15
 - DQ flags ACS:3—12
 - events ACS:5—13
 - fractional coverage ACS:5—13
 - reference file (CRREJTAB) ACS:3—17
- counts
 - flux conversion Intro:3—19
 - magnitude conversion Intro:3—19
- Cross talk
 - dependence on sky ACS:5—32
 - impact on object flux ACS:5—32
 - intensity ACS:5—30
 - quadrant dependence ACS:5—30
 - readout noise ACS:5—34
- CTE ACS:6—14
 - internal ACS:6—21
 - parallel ACS:6—16, ACS:6—19
 - PSF ACS:6—20
 - serial ACS:6—18, ACS:6—20

- trending ACS:6—19, ACS:6—20
- YCTE equation ACS:6—17, ACS:6—19
- cursor
 - splot commands Intro:3—36
- D**
- Dark Current
 - distribution tails ACS:5—9
 - features ACS:5—6
 - reference file (DARKFILE) ACS:3—20, ACS:5—6
 - subtraction, pipeline calibration ACS:3—20
 - superdark ACS:5—11
- Data
 - array size (arcsec) ACS:4—6
 - array size (pixels) ACS:2—8, ACS:4—6
 - associations ACS:2—2
 - file size (MB) ACS:2—8
 - file suffixes ACS:2—2
 - FITS file structure ACS:2—5
 - storage requirements ACS:2—7
- data
 - analysis software, STSDAS Intro:3—1
 - parameter types, IRAF Appendix:A—10
- Data Quality
 - array initialization ACS:3—11, ACS:3—19
 - FITS image extension ACS:2—5
 - flags
 - cosmic-rays ACS:3—12
 - hot pixels ACS:3—12
 - saturation ACS:3—12
 - pixel flag values ACS:3—12
- data quality
 - PDQ files Appendix:B—4
- Data Storage
 - buffer memory ACS:1—8
 - file size (MB) ACS:2—8
 - storage requirements ACS:2—7
- database
 - synphot Appendix:A—16
- dataset
 - see also "imset"
- Detectors
 - CCD ACS:1—3
 - MAMA ACS:1—3
- disconlab task
 - position display Intro:3—13
- display
 - display task Intro:3—7
 - image Intro:3—7
 - SAOimage Intro:3—9
 - spectra Intro:3—23
- display task
 - images in STSDAS Intro:3—7
- Distortion
 - see "Geometric Distortion"
- Dithered data
 - combining ACS:4—14
- documentation
 - IRAF Intro:3—40
 - STSDAS Intro:3—40
- Drizzling
 - ACS software tools ACS:4—16
 - applying user-defined shifts ACS:4—24
 - introduction ACS:4—15
 - kernel ACS:4—15
 - MultiDrizzle software ACS:4—34
 - pixfrac ACS:4—15
 - PyDrizzle ACS:4—17
 - scale ACS:4—15
- Dust motes ACS:5—25
- E**
- echelle spectra
 - echplot task Intro:3—24
- echplot task
 - plot echelle spectra Intro:3—24
- engineering data
 - OMS logs Appendix:C—1
- environment variable
 - IRAF Appendix:A—11
- eparam task
 - editing parameters Appendix:A—9
- Error Array
 - FITS image extension ACS:2—5
 - initialization ACS:3—10, ACS:3—19
- extension
 - FITS file Intro:2—2
 - FITS, appending Intro:2—7

F

file

PostScript, creating Intro:3—26

files

data formats Appendix:A—13

data quality (PDQ) Appendix:B—4

FITS, working with Intro:2—4

naming conventions Intro:2—1

observation log Appendix:B—4

observer comments (OCX) Appendix:B—4

rootname Appendix:B—3

specifying, STIS Intro:3—24

trailer Appendix:B—5

Filter wheel position accuracy ACS:5—26

FINE LOCK

guidance Appendix:C—12

FITS

files, working with Intro:2—2—Intro:2—11

format, described Intro:2—2

GEIS files in Intro:2—12

table Intro:2—8

table, array in cell Intro:2—10

waiver Intro:2—17

FITS file structure ACS:2—5

fitting package

fit spectra Intro:3—37

tasks in Intro:3—37

Flat field

coronagraphy ACS:6—33

correction, pipeline calibration ACS:3—21

earth flat ACS:5—24

ground flat ACS:5—17

L-flat correction ACS:5—18

pipeline flats ACS:5—21

pixel area map ACS:6—5

polarimetry ACS:6—30

ramp filters ACS:6—36

reference file (PFLTFILE) ACS:3—21

sky flat ACS:5—24

spectroscopy ACS:6—41

flux

combine with wavelength, mkmultispec Intro:3—31

from counts Intro:3—19

format

IRAF and STSDAS files Appendix:A—13

FOS

display spectra Intro:3—23

Full well depth

HRC ACS:5—39

WFC ACS:5—39

fwplot task

spectra display Intro:3—25

G

Gain calibration

HRC ACS:5—38

WFC ACS:5—38

Gain correction ACS:3—21

GEIS format

described Appendix:A—13

header file Intro:2—14

working with Intro:2—14

Generic Conversion

OPUS processing ACS:3—2

Geometric Distortion

calibration program ACS:4—5

detector orientations ACS:1—5

distortion model ACS:4—2

HRC ACS:4—8

overview ACS:1—4, ACS:4—2

reference file (IDCTAB) ACS:4—12

SBC ACS:4—10

velocity aberration correction ACS:4—3

WFC ACS:4—5

geometric distortion

correction Intro:3—13

GHRS

display spectra Intro:3—23

group

number in image Intro:2—14

working with Intro:3—11

grspec task

plot groups Intro:3—26

guidance mode

observation log Appendix:C—12

guide stars

acquisition Appendix:C—13

acquisition failure Appendix:C—14

dominant roll Appendix:C—13

number used Appendix:C—12

H

hardcopy

see "print" or "paper products"

Header

extension header keywords ACS:2—14

primary header keywords ACS:2—9

header

file, GEIS Intro:2—14

keyword, inheritance in FITS Intro:2—6

header data unit

FITS file Intro:2—2

help

STSDAS and IRAF tasks Intro:3—4, Appendix:A—7

HLSP Intro:1—19

Hot columns ACS:5—5

Hot pixels ACS:3—3, ACS:5—9

"Warm" pixels ACS:5—9

anneal rate ACS:5—12

DQ flags ACS:3—12

growth ACS:5—7

trending ACS:5—11

HRC

capabilities ACS:1—2

description ACS:1—2, ACS:1—3

distortion ACS:1—4, ACS:4—8

I

IDCTAB

distortion table ACS:4—12

IDL (Interactive Data Language) Intro:3—2

igi

plotting with Intro:3—27

printing plots Intro:3—26

image

display Intro:3—7

GEIS file Intro:2—15

plot data, implot Intro:3—14

section Intro:3—11

see also "FITS"

STSDAS tasks Intro:3—4

working with Intro:3—4

Image Reduction and Analysis Facility

see "IRAF"

image set

see "imset"

imcopy task

FITS files Intro:2—7

imexamine task

image display and plot Intro:3—15

imgtools package

multigroup GEIS images Intro:3—4

implot task

plot image data Intro:3—14

imset

combination, msjoin task Intro:3—18

extraction, mssplit task Intro:3—18

statistics, msstatistics task Intro:3—17

STSDAS tasks for Intro:3—16

imtab task

header to table Intro:3—32

IPPSSOOOT

see "files, naming conventions"

IRAF

basics Appendix:A—1

described Intro:3—1, Appendix:A—1

documentation Intro:3—40

obtaining Appendix:A—15

parameter, data type Appendix:A—10

pipng Appendix:A—6

psikern, PostScript Intro:3—26

setup Appendix:A—2

spectra analysis Intro:3—33

tasks available Appendix:A—1

J

Java Intro:3—3

jitter

effect on target lock Appendix:C—15

images from OMS Appendix:C—1

plotting Appendix:C—16

K

Keywords

extension header ACS:2—14

primary header ACS:2—9

keywords

FITS header Intro:2—6

see also "header"

L

Linearity

HRC ACS:5—39, ACS:5—43

WFC ACS:5—40

Linearity Correction

MAMA ACS:3—19

reference file (MLINTAB) ACS:3—19

lparam task

viewing parameters Appendix:A—9

M

magnitude

from counts Intro:3—19

math

see "arithmetic"

mkiraf command

IRAF setup Appendix:A—3

mkmultispec task Intro:3—31

moving target

acquisition Appendix:C—14

msarith task

imset arithmetic Intro:3—17

mscombine task

combine imset Intro:3—17

msjoin task

combine imset Intro:3—18

mssplit task

extract imset Intro:3—18

msstatistics task

imset statistics Intro:3—17

mstools package

FITS image extensions Intro:3—6

MultiDrizzle

data products ACS:4—19, ACS:4—52

example ACS:4—37

list of tasks performed ACS:4—34

removing cosmic-rays ACS:3—6

system requirements ACS:4—35

use with PyDrizzle ACS:3—6

multispec format

described Intro:3—31

N

naming conventions

files, HST data Intro:2—1

nfit1d task Intro:3—37

ngaussfit task Intro:3—37

NICMOS

imset, STSDAS tasks Intro:3—16

O

observation log

files Appendix:B—4

observer comment file

described Appendix:B—4

Observing Considerations

coronagraph ACS:1—7

dithering ACS:1—7

parallel observing ACS:1—8

pattern ACS:1—7

target acquisitions ACS:1—7

OCX file

observer comments Appendix:B—4

OMS

observation log files Appendix:B—4

Optical Ghosts ACS:5—26

Optics

coronagraph ACS:1—4

HRC/SBC ACS:1—4

WFC ACS:1—4

OPUS

see "pipeline"

OTFR Intro:1—2

on-the-fly reprocessing ACS:3—2

Overscan Region Table

reference file (OSCNTAB) ACS:3—13

Overscan subtraction ACS:3—12, ACS:5—3

P

package

IRAF concept Appendix:A—5

STSDAS, structure Intro:3—4, Intro:3—5

parameter

data types Appendix:A—10

- see also "eparam" and "lparam"
 - setting, IRAF Appendix:A—9
 - PDQ file
 - described Appendix:B—4
 - Photometry ACS:6—1
 - aperture correction ACS:6—4
 - bypassing PHOTCORR during recalibration ACS:3—27
 - color correction ACS:6—5
 - coronagraphy ACS:6—34
 - header keywords ACS:3—24
 - photometric systems ACS:6—1, ACS:6—2
 - pixel area map ACS:6—8
 - red leaks ACS:6—21
 - reference files (GRAPHTAB, COMPTAB) ACS:3—24
 - UV contamination ACS:6—22
 - zeropoints ACS:6—2
 - photometry
 - basic, in STSDAS Intro:3—18
 - synthetic Intro:3—21
 - pipe
 - IRAF task Appendix:A—6
 - Pipeline
 - CALACS image calibration ACS:3—3
 - generic conversion ACS:3—2
 - PyDrizzle dither combining ACS:3—5
 - pipeline
 - files produced by Intro:2—1
 - Pixel area map ACS:6—5
 - extended sources ACS:6—8
 - geometric distortion ACS:6—11
 - photometry ACS:6—8
 - point sources ACS:6—10
 - pixel coordinate
 - converting to RA and Dec Intro:3—13
 - pixel data
 - GEIS file Intro:2—15
 - Pixel Scale
 - average scale ACS:4—6
 - default drizzled image scale ACS:4—13
 - plot
 - igi task Intro:3—27
 - pointing stability Appendix:C—15
 - Polarimetry
 - calibration ACS:6—26, ACS:6—30
 - diattenuation ACS:6—27
 - flat fields ACS:6—30
 - geometric distortion ACS:6—29
 - instrumental ACS:6—27
 - phase retardance ACS:6—27
 - zeropoint ACS:6—30, ACS:6—32
 - position
 - RA and Dec in STSDAS Intro:3—12
 - Post-Flash Correction
 - reference file (FLSHFILE) ACS:3—14
 - PostScript
 - psikern, IRAF Intro:3—26
 - print
 - plots, igi Intro:3—26
 - PSF
 - CTE ACS:6—20
 - red halo ACS:6—12
 - UV ACS:6—13
 - variation ACS:6—11
 - psikern
 - PostScript IRAF kernel Intro:3—26
 - PyDrizzle
 - association tables ACS:4—19
 - data products
 - context image ACS:4—20, ACS:4—53
 - science image ACS:4—19, ACS:4—52
 - weight image ACS:4—20, ACS:4—53
 - memory requirements ACS:3—27
 - overview ACS:4—17
 - PyRAF environment ACS:4—20
 - use with MultiDrizzle ACS:3—6
 - PyFITS Intro:3—2
 - PyRAF Intro:3—3
 - Python Intro:3—2
- Q**
- quantization noise ACS:5—2
- R**
- Ramp filters
 - apertures ACS:6—35
 - flat fields ACS:6—36
 - ground calibration ACS:6—35
 - Readout noise ACS:5—2
 - Recalibration
 - bypassing PHOTCORR ACS:3—27
 - manual ACS:3—25

- on-the-fly ACS:3—2
 - removing cosmic rays ACS:3—6
 - requirements ACS:3—25
 - see also "Calibration"
 - setting switches ACS:3—26
 - recentering
 - jitter Appendix:C—15
 - Red leaks
 - HRC ACS:6—21
 - SBC ACS:6—22
 - Reference files
 - bad pixel table ACS:3—11
 - bias image ACS:3—13
 - calibration switches ACS:3—26
 - CCD characteristics table ACS:3—10
 - cosmic-ray rejection table ACS:3—17
 - dark image ACS:3—20, ACS:5—6
 - flat field ACS:3—21
 - header keywords ACS:2—11
 - MAMA linearity table ACS:3—19
 - overscan table ACS:3—12
 - post-flash image ACS:3—14
 - shutter shading image ACS:3—22
 - throughput component tables ACS:3—24
 - Repeat Observations
 - summing ACS:3—25
 - resample task
 - flux to wavelength Intro:3—31
 - rootname
 - see "files, naming conventions"
- S**
- SAOimage
 - display image Intro:3—9
 - Saturation
 - DQ flags ACS:3—12
 - Saturation limit
 - digital ACS:5—2
 - SBC
 - capabilities ACS:1—2
 - description ACS:1—3
 - distortion ACS:1—4, ACS:4—10
 - Scale
 - average pixel scale ACS:4—6
 - default drizzled pixel scale ACS:4—13
 - Scattered earth light ACS:5—36
 - section
 - image Intro:3—11
 - sgraph task
 - plot group Intro:3—25
 - plot STIS spectra Intro:3—23
 - Shading Correction
 - reference file (SHADFILE) ACS:3—22
 - Shift file ACS:4—24
 - Shutter operation ACS:5—45
 - Shutter stability
 - revised exposure times ACS:5—46
 - software
 - IRAF Appendix:A—1, Appendix:A—15
 - STSDAS Intro:3—1, Appendix:A—15
 - South Atlantic Anomaly ACS:5—13
 - Space Telescope Science Data Analysis System
 - see "STSDAS"
 - specfit task
 - fit models to spectrum Intro:3—40
 - spectra
 - analysis tasks,IRAF Intro:3—33
 - analysis, STSDAS Intro:3—27
 - analysis, tasks in STSDAS Intro:3—34
 - display Intro:3—23
 - display, STIS Intro:3—23
 - fitting Intro:3—37
 - specfit task Intro:3—40
 - Spectroscopy
 - AUTOIMAGE ACS:6—38
 - aXe software ACS:6—42
 - background ACS:6—40
 - bright stars ACS:6—39
 - calibration ACS:6—41
 - calibration keywords ACS:6—41
 - crowding ACS:6—40
 - dispersion ACS:6—37
 - dithering ACS:6—47
 - extracted spectra ACS:6—44
 - extra-field objects ACS:6—40
 - flat field ACS:6—41
 - flat fielding ACS:6—42
 - flux calibration ACS:6—48
 - resolution ACS:6—40

- wavelength calibration ACS:6—48
- zeroth order ACS:6—40
- splot task
 - cursor commands Intro:3—36
 - plot spectra Intro:3—35
- StarView
 - cross-qualification Appendix:C—11
 - data retrieval Intro:1—9
 - downloading Intro:1—4
 - observation logs Appendix:C—11
- Statistics
 - image ACS:3—24
- STIS
 - analysis, preparing Intro:3—28
 - display spectra Intro:3—23
 - echelle spectra, plotting Intro:3—24
 - imset, STSDAS tasks for Intro:3—16
- STSDAS Intro:3—1
 - astrometry Intro:3—14
 - astrometry in Intro:3—12
 - described Appendix:A—1
 - documentation Intro:3—40
 - image analysis tasks Intro:3—16
 - image display Intro:3—7
 - image, display Intro:3—11
 - image, section Intro:3—11
 - images Intro:3—4
 - imset tasks Intro:3—16
 - obtaining Appendix:A—15
 - organization of Intro:3—4, Intro:3—5
 - photometry in Intro:3—18
 - spectra analysis tasks Intro:3—34
 - synphot, database Appendix:A—16
 - synthetic photometry Intro:3—21
 - tables Intro:3—6
- Suffix
 - pipeline suffixes ACS:2—2, ACS:3—4
- suffix
 - see "files, naming conventions"
- synphot
 - database, obtaining Appendix:A—16
- synphot package
 - synthetic photometry Intro:3—21
- synthetic photometry
 - see "synphot"

T

- table
 - FITS Intro:2—8
 - STSDAS Intro:3—6
- target acquisition
 - moving target Appendix:C—14
- Target Acquisitions ACS:1—7
- task
 - IRAF concept Appendix:A—6
- tomultisecc task
 - extract STIS spectral orders Intro:3—28
- trailer file
 - described Appendix:B—5
- Trailer Files
 - comments from PyDrizzle ACS:4—18
 - creation by CALACS ACS:2—4
- transformation
 - pixel coordinates to RA and Dec Intro:3—13
- ttools package
 - STSDAS tables Intro:3—6
- txtable task
 - extract arrays Intro:3—30

U

- unit
 - counts to flux/mag, conversion Intro:3—19
- Units
 - calibrated data ACS:3—27
 - drizzled data ACS:3—27
 - raw data ACS:3—27

V

- variable
 - IRAF, environment Appendix:A—11
- Velocity Aberration
 - correction to distortion model ACS:4—3

W

- waiver FITS format Intro:2—17
- Warm pixels ACS:5—9
- wavelength
 - combine with flux, mkmultispec Intro:3—31
- Weight Image ACS:4—20, ACS:4—53

WFC

- capabilities ACS:1—1
- chip offsets ACS:4—2
- description ACS:1—2, ACS:1—3
- distortion ACS:1—4, ACS:4—5

WFPC2 associations Intro:1—19

World Coordinate System

- mkmultispec task Intro:3—31

X

xy2rd task

- pixel coordinates to RA and Dec Intro:3—13

Z

Zeropoint

- determining ACS:6—2
- photometric ACS:6—3