November 2022

Cosmic Origins Spectrograph Data Handbook

Version 5.1

Also available online at
https://hst-docs.stsci.edu/cosdhb

STScI | SPACE TELESCOPE SCIENCE INSTITUTE

3700 San Martin Drive
Baltimore, MD 21218
https://hsthelp.stsci.edu

COS Data Handbook

Version 5.1 - November 2022

PDF version

Cosmic Origins Spectrograph Data Handbook

Table of Contents

Expand all   Collapse all

User Support

Please contact the *HST* Help Desk for assistance. We encourage users to access the new web portal where you can submit your questions directly to the appropriate team of experts.

- **Website:** http://hsthelp.stsci.edu
- **E-mail:** help@stsci.edu

Additional Resources

Information and other resources are available on the COS website:

- http://www.stsci.edu/hst/instrumentation/cos

Revision History

Version	Date	Editor
5.1	November 2022	David Soderblom
5.0	August 2021	David Soderblom
4.0	May 2018	Marc Rafelski Senior Technical Editors: Susan Rose and Sharon Toolan
3.0	October 2015	Andrew Fox Senior Technical Editor: Susan Rose
2.0	January 2013	Derck Massa, Brian York and Svea Hernandez
1.0	March 2009	Brittany Shaw, Derck Massa, and Mary Elizabeth Kaiser

Authorship

This document is written and maintained by the COS Team in the Instruments Division of STScI with the assistance of associates in the Operations and Engineering Division. Contributions to the current edition were also made by COS team members John Debes, Serge Dieterich, Travis Fischer, William Fischer, Elaine Mae Frazer, David French, Sierra Gomez, Sten Hasselquist, Alec Hirschauer, Nick Indriolo, Bethan James, Robert Jedrzejewski, Christian Johnson, Darshan Kakkad, Nathaniel Kerman, Lauren Miller, Rachel Plesha, Marc Rafelski, Julia Roman-Duval, Kate Rowlands, David Sahnow, and Ravi Sankrit.

Citation

In publications, please refer to this document as

- Soderblom, D., et al. 2022, *COS Data Handbook*, Version 5.1, (Baltimore: STScI).

Introduction

Chapter Contents

How to Use this Handbook

This handbook is designed to help users manipulate, process, and analyze data from the Cosmic Origins Spectrograph (COS), which was installed on the *Hubble Space Telescope* (*HST*) during the 2009 servicing mission (SM4). It is designed for users familiar with *HST* data but new to COS.

The current edition of the *COS Data Handbook* was published in August 2021. It is presented as an independent and self-contained document, referred to as the *COS Data Handbook*.

For detailed information on the capabilities of the instrument, and how to plan observations, users should refer to the *COS Instrument Handbook*. For further information and timely updates, users should consult the COS Web page (http://www.stsci.edu/hst/instrumentation/cos), especially the Document Archive link. In particular, the STScI Analysis Newsletters (STANs) highlight changes in code and calibration procedures and provide other instrument-related news. The Instrument Science Reports (ISRs) present in-depth characterizations of the instrument and detailed explanations of calibration code and procedures.

Handbook Structure

The COS Data Handbook is organized in five chapters, which discuss the following topics:

- Chapter 1: COS Overview provides a brief overview of the instrument and its operational capabilities.
- Chapter 2: COS Data Files describes the contents of COS data files, the meanings of selected header keywords, and the relationship of the data products to the original Phase II proposal.
- Chapter 3: COS Calibration describes how the calibration pipeline processes observations, the content of COS reference files used during calibration, and how to run the calibration pipeline locally.
- Chapter 4: COS Error Sources describes the sources of uncertainty and limiting accuracies of COS data. COS observers should read this chapter to acquaint themselves with the limitations of the data that may remain after pipeline calibration.
- Chapter 5: COS Data Analysis describes how to analyze COS data with **Python** using **stenv**, as well as other **Python** libraries useful for optimizing data products and analyzing the data. It describes how to analyze target acquisitions and guide star tracking. It provides descriptions of different kinds of data and gives detailed instructions on how to work with them; specifically: extracted spectra, and `TIME-TAG` data.

Additional help with *HST* data is always available via the STScI Help Desk at http://hsthelp.stsci.edu or by email to help@stsci.edu.
Readers are advised to consult the COS web pages (http://www.stsci.edu/hst/instrumentation/cos) for the latest news and updates on COS performance.

Typographic Conventions

To help you understand the material in this Data Handbook, we will use a few consistent typographic conventions.

Visual Cues

The following typographic cues are used:

- **bold words** identify a **Python** library or function
- `typewriter-like` words identify a file name, system command, or response that is typed or displayed.
- *italic type* indicates a new term, an important point, a mathematical -variable, or a task parameter.
    - SMALLER CAPS

  identifies a header keyword.
- ALL CAPS identifies a table column.

Comments

Occasional side comments point out three types of information, each identified by an icon in the left margin.

> ⓘ *Warning: You could corrupt data, produce incorrect results, or create some other kind of severe problem.*

> ⚠ *Heads Up: Here is something that is often done incorrectly or that is not -obvious.*

> ✓ *Tip: No problems... just another way to do something or a suggestion that might make your life easier.*

> ⓘ *Information especially likely to be updated on the COS Web site is indicated by this symbol.*

Chapter 1: COS Overview

Chapter Contents

1.1 Instrument Capabilities and Design

The Cosmic Origins Spectrograph (COS) is a fourth generation *HST* spectrometer, designed to enhance the spectroscopic capabilities of *HST* at ultraviolet (UV) wavelengths. COS was built by Ball Aerospace Corporation to the specifications of Dr. James Green, the Principal Investigator (PI), at the University of Colorado at Boulder in conjunction with the COS Investigation Definition Team (IDT). Designed to primarily observe faint point sources, COS is optimized for maximum throughput, and provides moderate and low resolution spectroscopy in the UV and limited imaging in the NUV.

COS is a slitless spectrograph that employs two circular 2.5 arcsec diameter science apertures, the Primary Science Aperture (PSA) and the Bright Object Aperture (BOA). The PSA is an open aperture and the BOA contains a neutral density filter to attenuate the flux of bright objects. COS also contains two calibration apertures, the Wavelength Calibration Aperture (WCA) and the Flat-Field Calibration Aperture (FCA). Light from external sources does not reach these apertures. Instead they are illuminated by internal calibration lamps. The FCA is not available for observers, but the WCA can be used by observers to obtain wavelength calibration spectra. The WCA can be illuminated by one of two Pt-Ne wavelength calibration lamps. Similarly, the FCA can be illuminated by one of two deuterium flat-field calibration lamps.

The instrument has two channels: a far-ultraviolet (FUV) channel that is sensitive across the 900–2150 Å wavelength range and a near-ultraviolet (NUV) channel that provides wavelength coverage from 1650 to 3200 Å. The COS optical design achieves its high performance, particularly in the FUV, by minimizing the number of reflections in the optical path and the use of large format detectors which maximize the wavelength coverage per exposure. Each channel has its own photon-counting detector and a selection of gratings (Table 1.1). The FUV channel resolution varies with Lifetime Position (LP; see Appendix A) as shown in Figure 1.1. The NUV channel also has a mirror that can be used in two modes for imaging. The FUV channel uses a single reflection system where a high-efficiency, first-order, aspheric holographic grating corrects the beam in the dispersion direction but has low spatial resolution perpendicular to dispersion. *Only one channel may be used at a time.*

Table 1.1: COS Spectroscopic Modes.

Grating	Wavelength range (Å)	Bandpass per exposure and FUV Gap[1] (Å)	Resolution R = λ/FWHM[2]	Dispersion (mÅ pixel^{-1})
		FUV Channel		
`G130M`	900–1236	295/16	up to 11,500[3]	9.97
	1065–1365	296/15.7	10,000–15,000[4]	9.97
	1150–1450	292/14.3	12,000–24,000[4]	9.97
`G160M`	1405–1775	360/18.1	13,000–24,000[4]	12.23

G140L	~900–2150[3]	>1150/112	1,500–4,000[4]	80.3
NUV Channel				
G185M	1700–2100	3 × 35	16,000–20,000	37
G225M	2100–2500	3 × 35	20,000–24,000	33
G285M	2500–3200	3 × 41	20,000–24,000	40
G230L	1700–3200[5]	(1 or 2) × 400	2,100–3,200	390

[1] Width of gap between FUV detector segments.

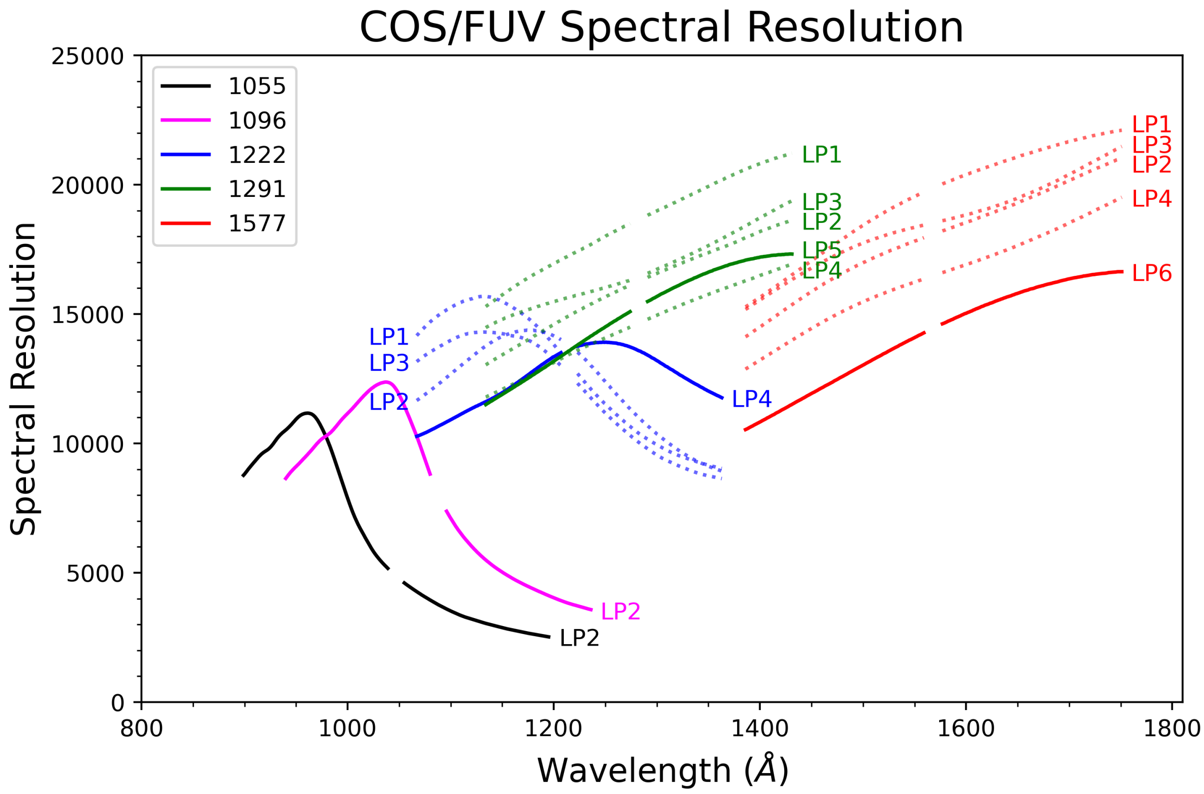
[2] Note that the resolution R increases approximately linearly with wavelength and is determined from the Line Spread Function (LSF) of COS, which is non-gaussian.

[3] R falls with increasing wavelength for the shortest wavelength settings of G130M: R=8,500–11,500 between 940 and 1080 Å.

[4] Resolution is dependent on wavelength, cenwave, and Lifetime Position. For more detail of the dependence of resolution on Lifetime Position, see Figure 1.1 and COS2025 policies.

[5] Some shorter wavelengths are recorded in second-order light.

Figure 1.1: COS FUV Spectral Resolution Depends on Lifetime Position.



This figure shows the spectral resolution *R* = / versus wavelength for five central wavelengths ("cenwaves"), four from the `G130M` grating (1055, 1096, 1222, 1291), and one from the `G160M` grating (1577). The two blue mode settings (1055 and 1096) remain at Lifetime Position 2 (LP2), and 1222 remains at LP4. In Cycle 29, cenwave 1291 moved from LP4 to LP5, and in Cycle 30, cenwave 1577 moved from LP4 to LP6. Compared to LP4, the resolution of LP5 is similar, and that of LP6 is slightly worse. The resolution curves shown are generated by the Code V optical models, which have been validated through on-orbit observations.

1.1.1 FUV Spectroscopy

The FUV channel employs a large format cross delay line (XDL) detector consisting of two 16384 × 1024 pixel segments, referred to as FUV segments A and B. The segments are separated by a physical gap of 9 mm, which makes it impossible to obtain a continuous spectrum across the two segments with a single setting. The supported central wavelength positions were selected to enable full wavelength coverage of the gap. Table 1.2 shows the wavelength ranges of both segments for all possible FUV grating and central wavelength combinations for `FP-POS` = 3 (for a discussion on FP-POS and its usage, see the COS Instrument Handbook and "Grating Offset Positions" 1.1.3 below).

Table 1.2: Wavelength Ranges for FUV Gratings for `FP-POS` = `3`.

Grating	Central wavelength setting (Å)[1]	Recorded wavelengths[2]	
		Segment B	Segment A

`G130M`	1055	899–1040	1055–1196
	1096	940–1080	1096–1236
	1222	1067–1207	1223–1363
	1291	1134–1274	1291–1431
	1300	1144–1283	1300–1441
	1309	1154–1294	1309–1450
	1318	1163–1303	1319–1460
	1327	1172–1313	1328–1469
`G160M`	1533[3]	1342–1515	1533–1707
	1577	1386–1559	1577–1751
	1589	1397–1571	1589–1762
	1600	1409–1581	1601–1774
	1611	1420–1594	1612–1786
	1623	1432–1606	1625–1798
`G140L`	800[3]	N/A[4]	815–1948
	1105	N/A[4]	1118–2251
	1280	<900–1165	1280–2391

[1] The central wavelength (cenwave) is (approximately) the shortest wavelength recorded on Segment A.

[2] All wavelengths recorded here are nominal, due to the uncertainties in the position of the OSM1 mechanism.

[3] Cenwaves 800 and 1533 were implemented in Cycle 26.

[4] The G140L grating in the 800 and 1105 central wavelength settings moves the zero-order image onto segment B. Therefore, only segment A is available for these settings.

1.1.2 NUV Spectroscopy

To provide maximum wavelength coverage on the square format of the NUV detector, three mirrors simultaneously image three, fully aberration-corrected, spectra onto a single 1024 × 1024 Multi-Anode Micro-channel Array (MAMA) detector. Consequently, three separate regions of the spectrum are imaged onto the detector. These spectral regions, referred to as stripes A, B, and C, each span the physical length of the detector in the dispersion direction—but are not contiguous in wavelength space. The allowable grating positions were defined with two objectives: the capability of obtaining full spectral coverage over the NUV bandpass and maximizing scientific return with a minimum number of grating positions. As a result, several of the supported central wavelength positions were selected to maximize the number of diagnostic lines on the detector in a single exposure. Table 1.3 shows the wavelength ranges of the three stripes for all possible NUV grating and central wavelength combinations.

Table 1.3: Wavelength Ranges for NUV Gratings for `FP-POS` = 3.

Grating	Central wavelength setting (Å)[1]	Recorded wavelengths		
		Stripe A	Stripe B	Stripe C
`G185M`	1786	1670–1705	1769–1804	1868–1903
	1817	1701–1736	1800–1835	1899–1934
	1835	1719–1754	1818–1853	1916–1951
	1850	1734–1769	1833–1868	1931–1966
	1864	1748–1783	1847–1882	1945–1980
	1882	1766–1801	1865–1900	1964–1999
	1890	1774–1809	1872–1907	1971–2006
	1900	1783–1818	1882–1917	1981–2016
	1913	1796–1831	1895–1930	1993–2028
	1921	1804–1839	1903–1938	2002–2037
	1941	1825–1860	1924–1959	2023–2058
	1953	1837–1872	1936–1971	2034–2069
	1971	1854–1889	1953–1988	2052–2087
	1986	1870–1905	1969–2004	2068–2103
	2010	1894–1929	1993–2028	2092–2127
G225M	2186	2070–2105	2169–2204	2268–2303
	2217	2101–2136	2200–2235	2299–2334
	2233	2117–2152	2215–2250	2314–2349
	2250	2134–2169	2233–2268	2332–2367
	2268	2152–2187	2251–2286	2350–2385
	2283	2167–2202	2266–2301	2364–2399
	2306	2190–2225	2288–2323	2387–2422
	2325	2208–2243	2307–2342	2406–2441
	2339	2223–2258	2322–2357	2421–2456
	2357	2241–2276	2340–2375	2439–2474
	2373	2256–2291	2355–2390	2454–2489
	2390	2274–2309	2373–2408	2472–2507

	2410	2294–2329	2393–2428	2492–2527
`G285M`	2617	2480–2521	2596–2637	2711–2752
	2637	2500–2541	2616–2657	2731–2772
	2657	2520–2561	2636–2677	2751–2792
	2676	2539–2580	2655–2696	2770–2811
	2695	2558–2599	2674–2715	2789–2830
	2709	2572–2613	2688–2729	2803–2844
	2719	2582–2623	2698–2739	2813–2854
	2739	2602–2643	2718–2763	2837–2878
	2850	2714–2755	2829–2870	2945–2986
	2952	2815–2856	2931–2972	3046–3087
	2979	2842–2883	2958–2999	3073–3114
	2996	2859–2900	2975–3016	3090–3131
	3018	2881–2922	2997–3038	3112–3153
	3035	2898–2939	3014–3055	3129–3170
	3057	2920–2961	3036–3077	3151–3192
	3074	2937–2978	3053–3094	3168–3209
	3094	2957–2998	3073–3114	3188–3229
`G230L`	2635	1334–1733[2]	2435–2834	1768–1967[3]
	2950	1650–2050	2750–3150	1900–2100[3]
	3000	1700–2100	2800–3200	1950–2150[3]
	3360	2059–2458[4]	3161–3560[5]	2164–2361[3]

[1] The central wavelength setting (cenwave) corresponds to the approximate midpoint of stripe B.

[2] For central wavelength 2635 Å, the stripe A wavelengths are listed for completeness only (and in case a bright emission line falls onto the detector). The NUV detector's sensitivity at these wavelengths is extremely low. To obtain a low-resolution spectrum at wavelengths below ~1700 Å we recommend the FUV grating `G140L`.

[3] The values in shaded cells are wavelength ranges observed in second-order light. Their dispersion is twice that of the first-order spectrum. First-order flux, from wavelengths twice those of the listed range, will be present at the ~5% level.

[4] Lyman α may be present in second-order light.

[5] Longward of 3200 Å, second-order light may be present. At these wavelengths, the flux calibration applied by **calcos** is unreliable.

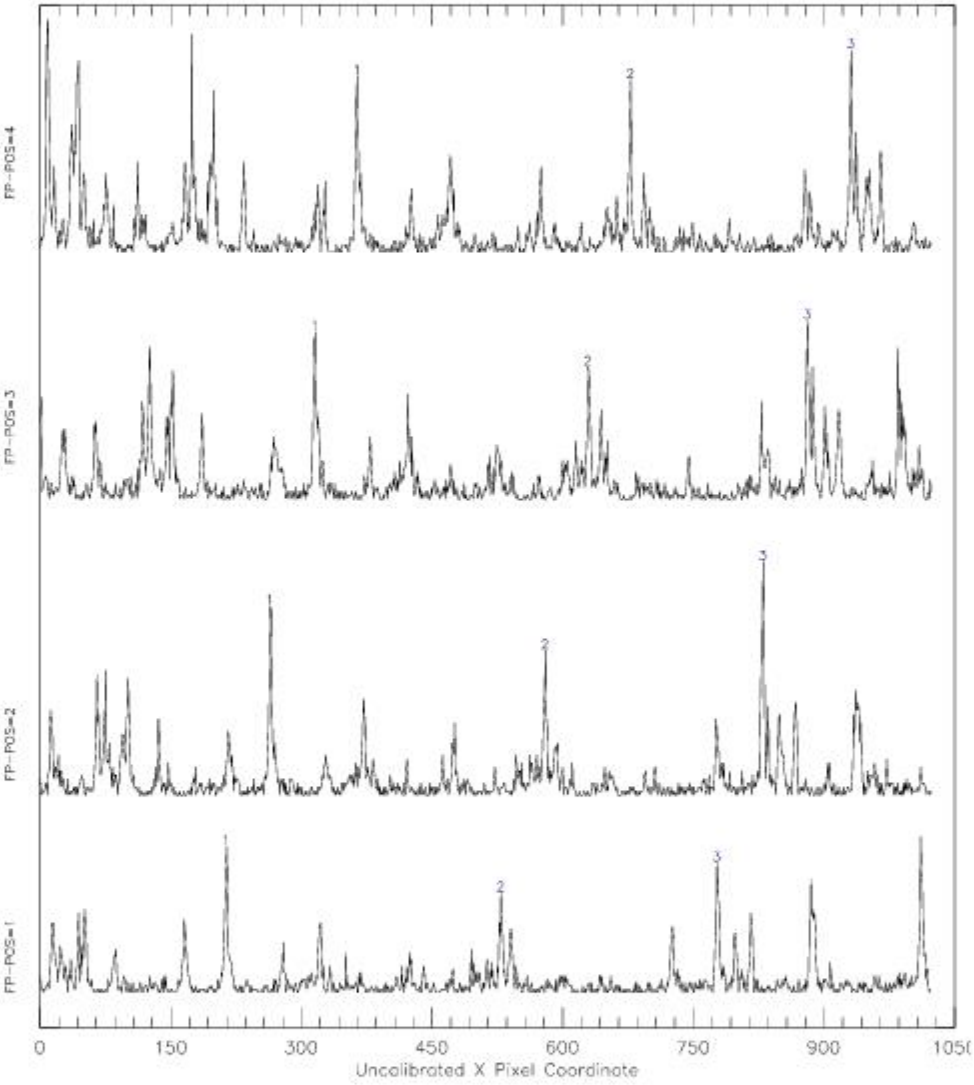
1.1.3 Grating Offset Positions (`FP-POS`)

For each NUV and FUV central wavelength setting there are four grating offset positions (`FP-POS=1–4`) available to move the spectrum slightly in the dispersion direction. This allows the spectrum to fall on different areas of the detector to minimize the effects of small scale fixed pattern noise in the detector. Figure 1.2 shows an example of the shifts in uncalibrated x-pixel coordinates of the NUV stripe B spectra for all four `FP-POS` positions.

1.1.4 NUV Imaging

COS imaging may only be done with the NUV channel and the spectral coverage includes the entire NUV bandpass from ~1650–3200 Å. This mode utilizes a flat mirror with two available mirror settings, MIRRORA and MIRRORB. The first setting uses a primary reflection off the mirror surface, and the second setting provides an attenuated reflection. MIRRORB and/or the BOA may be used to obtain images of brighter objects, but MIRRORB produces a secondary image and the BOA produces an image with coma that degrades the spatial resolution (Figure 5.2 and Figure 5.3).

Figure 1.2: Grating Offset Positions (FP-POS).



This figure shows spectra of an emission-line source obtained at all four `FP-POS` positions using the `G185M` grating with a central wavelength setting of 1850. The individual plots show the collapsed counts from the stripe B spectra versus the uncalibrated x-pixel coordinates. Note that the three features marked 1, 2, and 3, shift slightly for each `FP-POS` position.

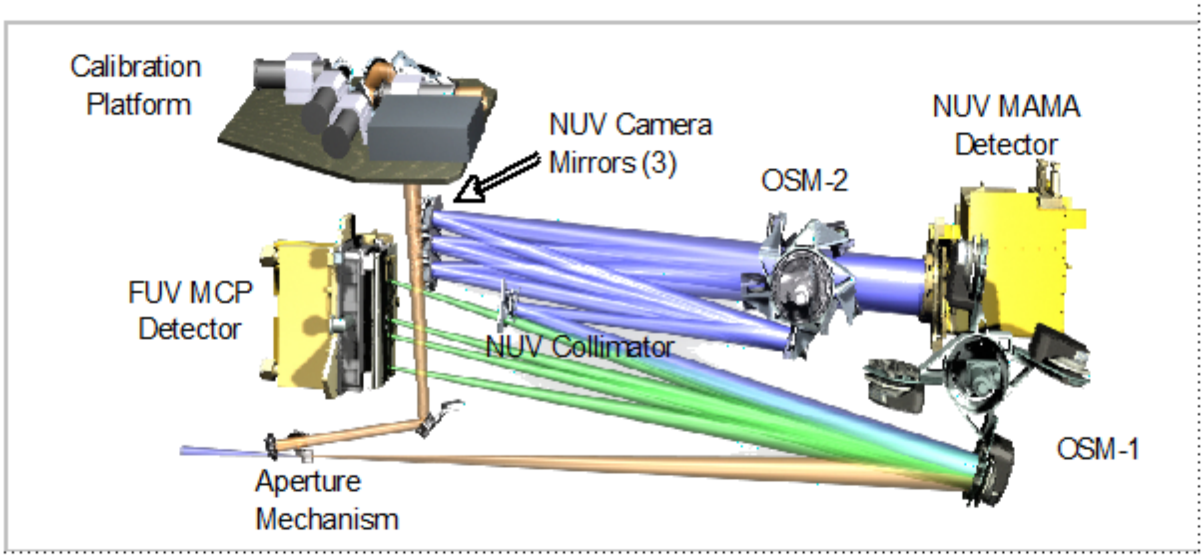
While the spatial resolution of COS NUV MIRRORA (Section 1.2) images can be good, the field of view is very small. Furthermore, because COS uses the aberrated PSF from the optical telescope assembly (OTA), and because the optics image the sky onto the detector, not the aperture, the image includes some light from sources out to a radius of about 2 arcsec. However, only point sources within about 0.5 arcsec of the aperture center have essentially all their light imaged, and so the photometric interpretation of a COS image can be inherently complex.

1.1.5 Data Collection Modes

COS has two modes of data collection, `TIME-TAG` and `ACCUM`, and only one mode can be used for a given exposure. In `TIME-TAG` mode the position, time, and for FUV, pulse height of each detected photon are tabulated into an events list, while in `ACCUM` mode the photon events are integrated onboard into an image. `TIME-TAG` data have a time resolution of 32 ms, and can be screened as a function of time during the post-observation pipeline processing to modify temporal sampling and exclude poor quality data. COS is optimized to perform in `TIME-TAG` mode, although `ACCUM` mode is fully supported in the pipeline processing. Note that in `ACCUM` mode no "walk" correction is made (see Section 3.4). `ACCUM` mode should be used primarily for UV bright targets that can not be observed in `TIME-TAG` mode due to high count rates. Users should note that FUV data taken in `ACCUM` mode store only a portion of the full detector since the 18 MB of onboard memory cannot hold a complete FUV image (containing both detector segments). `ACCUM` mode omits only the wavecal region and unused detector space, therefore the FUV `ACCUM` subarrays contain all of any external spectrum. The FUV `ACCUM` subarrays, whose sizes are 16384 × 128, are shown in Figure 2.2 for LP3.

1.2 COS Physical Configuration

Figure 1.3: The COS Optical Path and the Locations of the Mechanisms.

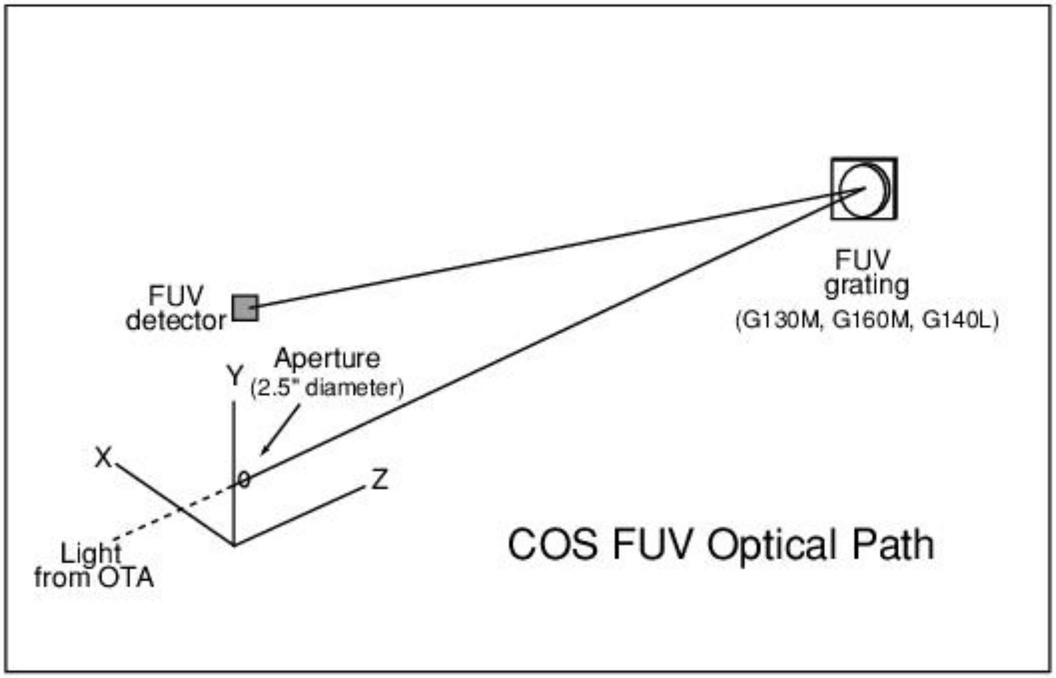


Scaled with all elements shown in their correct relative locations.

The COS optical design includes an external shutter, two science apertures, two calibration apertures, two Optics Select Mechanisms (OSM1 and OSM2), and separate NUV and FUV detectors. COS also has an independent calibration lamp assembly containing two Pt-Ne and two deuterium lamps, which can illuminate the detectors with an emission line or a continuum spectrum, respectively. The COS optical design and elements are displayed in Figure 1.3.

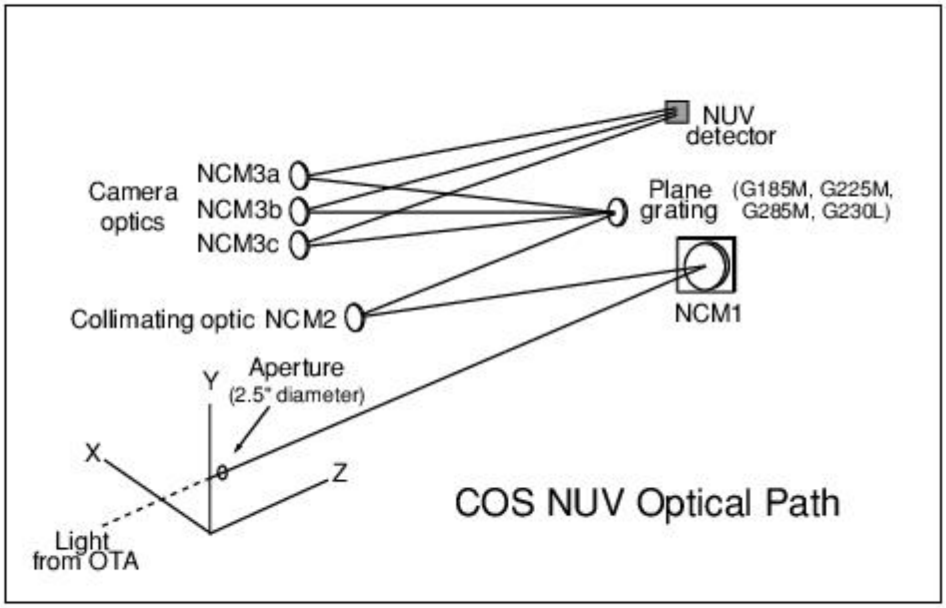
External light enters the aperture mechanism through either the PSA or the BOA and illuminates OSM1, which contains the three FUV gratings and a mirror. Each grating can be set to one of several positions, to obtain different wavelength ranges. The positioning of the OSM1 mechanism is not precisely repeatable, and this can cause small, but significant, variations in how the spectrum or image is projected onto the detector. This non-repeatability can be corrected in post-observation data processing using separate or concurrent (`TAGFLASH`) calibration lamp exposures (wavecals). The COS FUV channel optical path is illustrated in Figure 1.4.

Figure 1.4: The COS FUV Optical Path.



If the OSM1 is set to the mirror position, incoming light is directed to a collimating mirror, and then to OSM2, which contains a mirror for imaging and the four NUV gratings. Each grating offers multiple positions. As is the case with OSM1, the positioning of OSM2 does not repeat exactly, and the data need to be corrected in post-observation data processing via either separate or concurrent wavecals. If a grating is in place on OSM2, the dispersed light is imaged onto the NUV detector by three separate parallel camera mirrors (NCM3a, b, c). This results in three spectra, or stripes, covering different wavelength ranges. Full wavelength coverage may be obtained through multiple observations with different grating positions. Alternatively, if the plane mirror is in place on OSM2, the undispersed light is sent to the middle camera mirror (NCM3b) and then imaged onto the NUV detector. The plane mirror on OSM2 may be used in either of two settings, designated as MIRRORA and MIRRORB. The MIRRORA setting employs a direct reflection from the plane mirror. For the MIRRORB setting, the plane mirror is slightly offset to provide primary reflection off the order-sorting filter and hence an attenuation factor of approximately 25 compared to the MIRRORA setting. The COS NUV channel optical path is illustrated in Figure 1.5.

Figure 1.5: The COS NUV Optical Path (spectroscopic only).



A series of beam-splitters and fold mirrors direct light from the calibration lamp assembly (see Figure 1.3), through either the WCA or FCA and into the optical path. The calibration lamp assembly can provide continuum illumination with its deuterium lamps and emission line illumination with its Pt-Ne lamps to both the NUV and FUV spectrographs. The Pt-Ne lamps may be operated during `TIME-TAG` science exposures in order to produce concurrent wavelength calibrations (`TAGFLASH` mode).

1.2.1 The COS Detectors

COS uses two detectors, an FUV XDL and an NUV MAMA. Table 1.4 gives an overview of their characteristics.

Table 1.4: COS Detector Characteristics.

Detector Characteristic	FUV XDL	NUV MAMA
Photocathode	CsI (opaque)	Cs_2Te (semi-transparent)
Window	None	MgF_2 (re-entrant)
Wavelength range	<900–2150 Å	1650–3200 Å
Active area	85 × 10 mm[1]	25.6 × 25.6 mm
Pixel format (full detector)	16384 × 1024[1]	1024 × 1024
Image size recorded per spectrum	16384 × 128 (`ACCUM`)[1] 16384 × 1024 (`TIME-TAG`)[1]	1024 × 1024

Pixel size	6 × 24 μm 0.023 × 0.092 arcsec	25 × 25 μm 0.0235 × 0.0235 arcsec
Spectral resolution element size (= "resel")	6 × 10 pix[2]	3 × 3 pix
Plate scale: Along dispersion (per resel)	0.13 arcsec	0.075 arcsec
Plate scale: Cross dispersion (per resel)	0.92 arcsec	0.075 arcsec
Plate scale: Imaging (per resel)	N/A	0.075 arcsec
Quantum efficiency	~26% at 1335 Å ~12% at 1560 Å	~10% at 2200 Å ~8% at 2800 Å
Typical Dark count rate (away from SAA)[3]	1.13 cnt s^{-1} cm^{-2} 1.63×10^{-6} cnt s^{-1} pix^{-1} 9.77×10^{-5} cnt s^{-1} resel^{-1}	140 cnt s^{-1} cm^{-2} 8.77×10^{-4} cnt s^{-1} pix^{-1} 7.89×10^{-3} cnt s^{-1} resel^{-1}

[1] Sizes given are for an individual FUV segment.

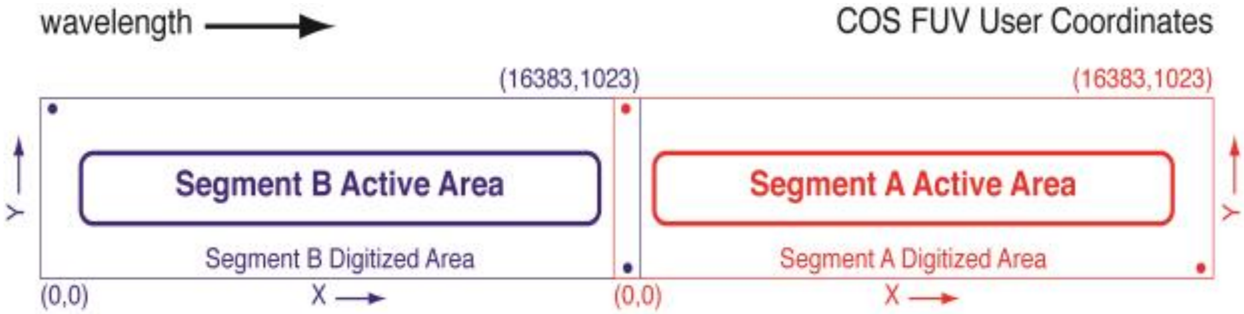
[2] For most modes.

[3] NUV dark rate is time dependent. NUV and FUV dark rates are for 2017. For updated information on NUV and FUV dark rates please see the COS performance monitoring pages.

FUV Channel

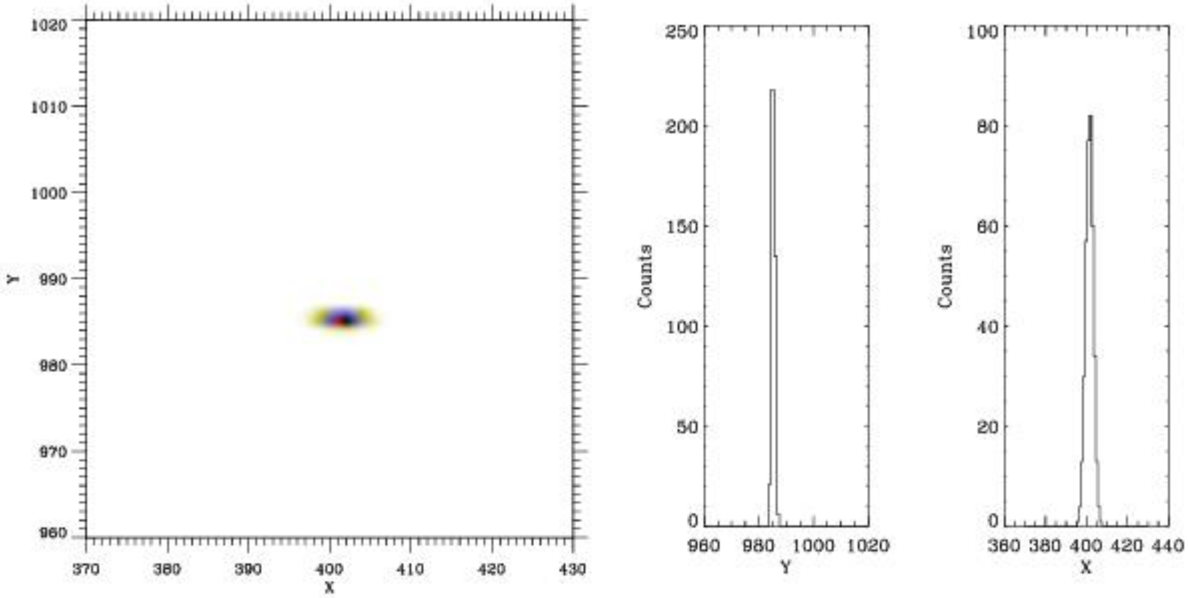
The FUV channel uses a large-format, windowless solar-blind cross delay line (XDL) detector. This is a two-segment photon-counting detector with microchannel plates feeding an XDL anode. The data are digitized to a 16384 × 1024 pixel format for each segment; however the active area is only 14191 × 439 for Segment A (FUVA) and 14374 × 426 for Segment B (FUVB). Because there are no physical pixels, fiducial electronic pulses are recorded at specific times throughout an observation to permit alignment of data to a standard reference frame. These electronic pulses are referred to as "stim pulses." Figure 1.6 schematically shows the COS FUV XDL segments with the locations of the active areas and stim pulses. The stim pulses emulate counts located near the edges of the anode, beyond the illuminated portions of the detector. A zoomed-in image of one of the FUV stim pulses on Segment B is shown in Figure 1.7. An example of an FUV external science spectrum taken with Segment B is shown in Figure 1.8, with a simultaneous wavelength calibration spectrum.

Figure 1.6: The FUV XDL Detector.



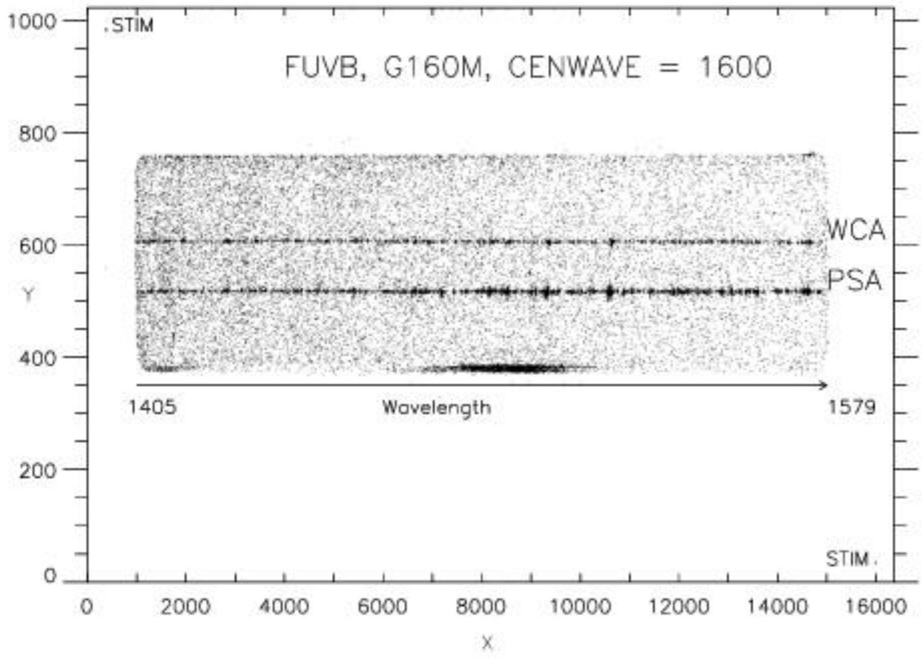
Layout of the FUV detectors (drawn to scale). The slight curvature at the corners of the active areas is also present on the flight detectors. The red and blue dots show the approximate locations of the stim pulses. The numbers in parentheses are the pixel coordinates at the corners of the segment's digitized area.

Figure 1.7: COS FUV Stim Pulse.



Left: **A portion of an image in the FUV detector with a typical stim pulse is shown.** ***Right:*** **A histogram of the stim pulse profile in the x and y direction. The electronic stim pulses are used to remove thermal distortions and to map the XDL detector elements to a standard reference frame.**

Figure 1.8: Example of a COS FUV Spectrum.



Wavelength calibration spectra for FUV Segment B with G160M/1600 obtained during ground testing. The upper spectrum is from the internal wavelength calibration lamp obtained through the WCA. The lower spectrum is from an external lamp obtained through the PSA. The bright streak at the bottom is due to an area of enhanced background on the detector segment. Note that the size of the active area is smaller than the overall digitized area. The stim pulses are also visible in the upper left and lower right corners.

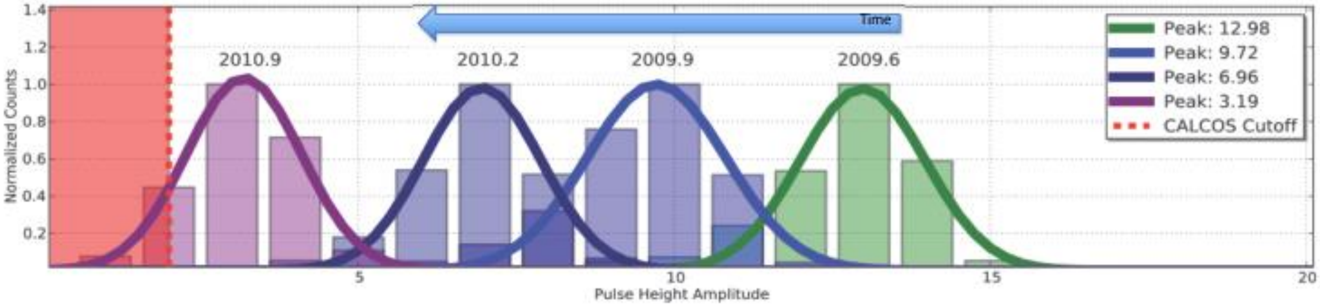
With each recorded event on the XDL detector, the total charge in the associated electron cloud incident on the anode is recorded. For FUV `TIME-TAG` data this pulse height amplitude (PHA) is sent to the ground along with the position of the event and can be used during data analysis to help identify non-photon events, see Figure 1.9. For FUV XDL `ACCUM` mode data, only an integrated pulse height distribution (a histogram of the PHA data) for the entire segment is available.

A photon landing on an FUV detector segment creates an event (a cascade of electrons) at the backside of the detector which is characterized by a PHA that is detected by the electronics. The detector electronics distinguishes between real and background noise events by the value of the PHA, with background events having low PHAs and real events large PHAs. However, as a region of the detector is exposed to more and more light, the PHAs that it produces become smaller, an effect called "gain sag." Gain sag can cause two effects: *the mis-registration of event positions* and *localized sensitivity loss*.

Mis-registration of event positions as a function of PHA is termed "walk." Walk has been identified in both the dispersion (X) and cross-dispersion (Y) directions. A walk correction is made by the COS calibration pipeline for TIME-TAG data (see Section 3.4.5).

Localized sensitivity loss occurs when the PHAs for some pixels become too small to be distinguished from background events, causing events to be missed or filtered out. This results in a localized region of low sensitivity. Eventually, the gain of all of the pixels in a region becomes so small that photons landing on that location no longer create events with valid PHAs, see Figure 1.9. In that case, no events are registered and the region is termed a "dead spot." When this occurs, it is necessary to either increase the high voltage applied to the detector (which increases the PHAs of all the pixels), or to move the aperture so that the science spectra land on a different portion of the detector (which has not been exposed to as much light). The COS FUV detectors have already experienced localized gain sag on regions of the FUVB and FUVA detectors exposed to the bright Ly-α airglow line when the G130M and G140L are used respectively. As a result, the detector high voltage has been raised on several occasions (see Appendix B.1). Furthermore, the default location for the science spectra and target acquisitions is periodically moved to an unsagged region of the detector. For more information on lifetime positions consult Appendix A.1.

Figure 1.9: Example of a COS FUV Pulse Height Distribution at different times.

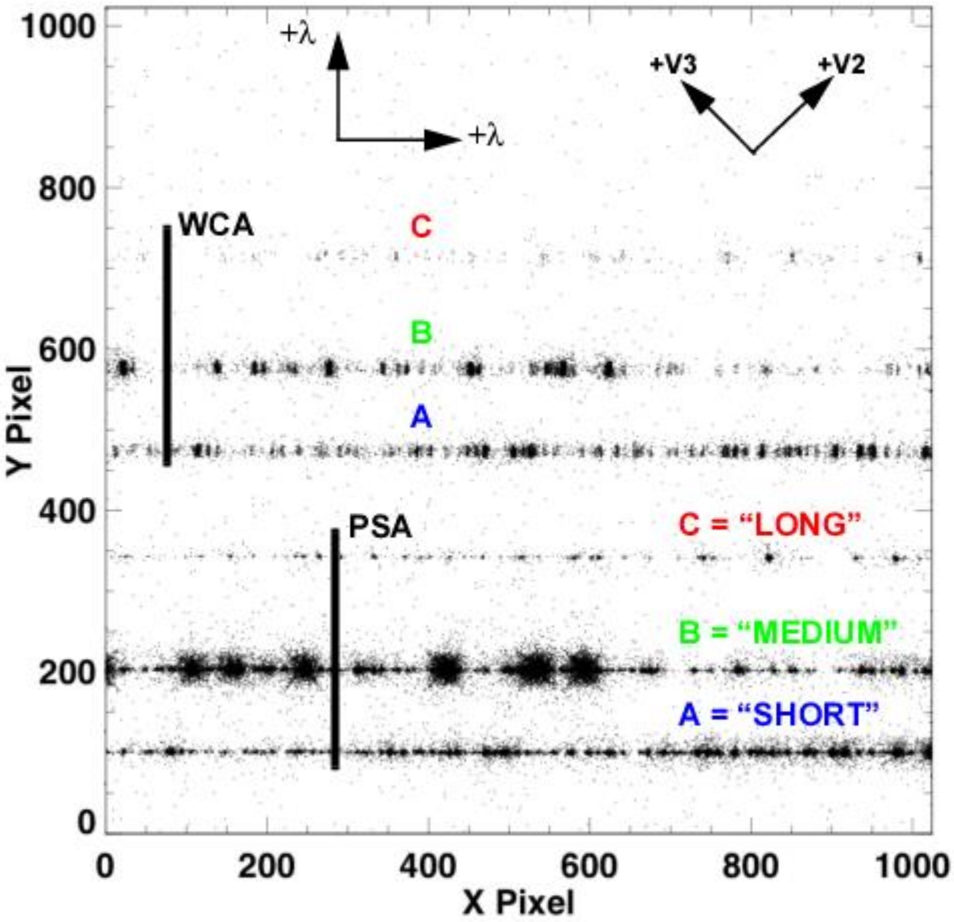


The PHAs for photon events falling in a given region shifts to lower values as charge is extracted from the detector. The normalized distributions from a single area are shown over 4 different periods from 2009.6 to 2010.9. Over this period, the peak in the pulse height distribution (modal gain), as measured by a gaussian fit, has decreased from nearly 13 to just above 3. The red region on the left shows the calcos cutoff value. Any events falling in this area are screened out by calcos processing as noise, and the events will be lost to the final spectrum.

NUV Channel

The NUV channel uses a 1024 × 1024 pixel Multi-Anode Micro-channel Array (MAMA) detector. This detector has a semi-transparent cesium telluride photocathode on a magnesium fluoride window, which allows detection of photons with wavelengths from ~1150 to ~3200 Å. The NUV MAMA provides no pulse-height information, but may be used in both `ACCUM` and `TIME-TAG` mode. The NUV channel creates three spectrum stripes on the MAMA detector, resulting in three separate stripes for the science data and three for wavelength calibration data as shown in Figure 1.10.

Figure 1.10: Example of a COS NUV Spectrum.



Wavelength calibration spectra obtained from the internal source through the WCA (upper three stripes) and an external source through the PSA (lower three stripes). The stripes are designated A, B, and C, in going from bottom to top for each source. Wavelength increases from left to right in each stripe and from bottom to top (hence the SHORT, MEDIUM, and LONG designations).

1.3 Basic Instrument Operations

1.3.1 Target Acquisitions

The details of acquiring objects with COS are described in Chapter 8 of the *COS Instrument Handbook*. In brief, the COS flight software[1] provides several methods for acquiring and centering a target in the aperture in both imaging and dispersed light modes. The simplest and fastest method uses the `ACQ/IMAGE` command to obtain a direct NUV image of the target field, and then moves the telescope so that the aperture is at the centroid of the measured light. This is the preferred method, but the target coordinates must be accurate enough to ensure that it falls within the aperture after the initial pointing of the telescope such that centering algorithms can be used. If high-accuracy coordinates were not available, a spiral search (`ACQ/SEARCH`) would be performed with either detector prior to other acquisition methods to ensure the target will fall within the aperture. The other COS acquisition methods (`ACQ/PEAKXD` and `ACQ/PEAKD`) use dispersed light from the target, and could also be performed with either detector.

1.3.2 Routine Wavecals

Routine wavelength calibration exposures, or wavecals, are needed by the COS calibration pipeline, **calcos**, to compensate for the effects of OSM drifts. All wavelength calibration exposures are taken in `TIME-TAG` mode. They may be obtained in either the `TAGFLASH` mode, where `FLASH=YES` for `TIME-TAG` science observations, or in separate wavelength calibration exposures that are either automatic or user-specified.

For `TAGFLASH` exposures, the wavecal lamp is turned on briefly at the start of an externally targeted exposure, and again at predefined intervals throughout the exposure. In this mode, photons from the external science target and the internal wavelength calibration source are recorded simultaneously on different portions of the detector; see Figure 1.8 and Figure 1.10.

For `TIME-TAG` exposures not done in `TAGFLASH` mode, a separate wavecal exposure will be automatically performed (AUTO wavecal) for each set of external spectrographic science exposures using the same spectral element, central wavelength, and `FP-POS` value. These automatic wavecals are performed after the first such science exposure and after each subsequent science exposure if more than 40 minutes of visibility time has elapsed since the previous wavecal and the same spectrograph set-up has been in use over that time.

For `TIME-TAG` exposures taken at LP6, a non-concurrent wavelength calibration exposure will be automatically performed for each set of external spectrographic science exposures using the same spectral element, central wavelength, and FP-POS. Unlike AUTO or GO wavecals, these SPLIT wavecals are taken after moving the aperture to a different location on the detector. As described more in Section 3.4.12, LP6 exposures of length > 960s receive an additional wavecal shift to compensate for the lack of concurrent wavecal flashes that occur for more typical `TIME-TAG` exposures at other LPs.

Observers also have the ability to insert additional wavecals by specifying `TARGET=WAVE` (GO wavecal). These exposures will use the same calibration lamp configurations and exposure times as the automatic wavecals. The only way to tell the difference between GO and automatic wavecal data is to look at the `MEMTYPE` header keyword, which will be discussed later in Table 2.6 of the "Association Tables (ASN)" Section.

1.3.3 Typical COS Observing Sequence

For most observations, the following sequence of events occurs:

- Acquire the object using `ACQ/IMAGE` with the NUV detector. This may be preceded by an `ACQ/SEARCH` if needed to scan a larger area of sky. If the target is bright enough, the `ACQ/PEAKXD`, `ACQ/PEAKD` sequence can be used.
- Obtain a spectrum in `TIME-TAG` mode using `TAGFLASH` mode so that the data can be corrected for any OSM drifts, and with different `FP-POS` positions to enhance the signal-to-noise.
- Obtain more spectra during additional orbits as needed to achieve the desired signal-to-noise. Alternatively, one could specify additional cenwaves that include the features of interest.

The typical COS observing sequence depends greatly on the type of observation specified. Typical COS observations use `TIME-TAG` mode and the PSA, with simultaneous wavelength calibrations taken via `TAGFLASH`. Multiple exposures are often used to cover the FUV detector gap, or to produce full wavelength coverage from the NUV wavelength stripes.

---

[1] The flight software minor revision number in the SPT file header is a hexadecimal number identified as a decimal. Therefore, to obtain the correct flight software number used for a given observation, use the 'FLTSWVER' header keyword instead.

1.4 COS Coordinate System

References to multiple coordinate systems appear in the headers of COS data. These are tied to the OTA frame, the User frame, and the `POS-TARG` frame. The following is a brief explanation of how these systems (shown in Figure 1.11) are related, and a more thorough explanation can be found in the Phase II Instructions.

The three coordinate systems of interest are the:

- OTA or "V" Frame (V_1, V_2, V_3): the common coordinate system for Scientific Instruments and the FGSs. It is a distortion-free frame whose unit is arc seconds.
- User Frame (X_{user}, Y_{user}): the frame associated with a pipeline science image. It is aligned with the detector, where the wavelength increases along $+X_{user}$.
- Detector Frame ($X_{detector}$, $Y_{detector}$): the frame associated with the detector, which is related to the User Frame and the size of the detector. The relation between the two is provided below.
- POS-TARG Frame (X_{POSTARG}, Y_{POSTARG}): a distortion-free frame with units of arc seconds. Its origin coincides with the science aperture and its axes are closely aligned with the user frame.

The angles associated with these frames that appear in the headers of COS data files are:

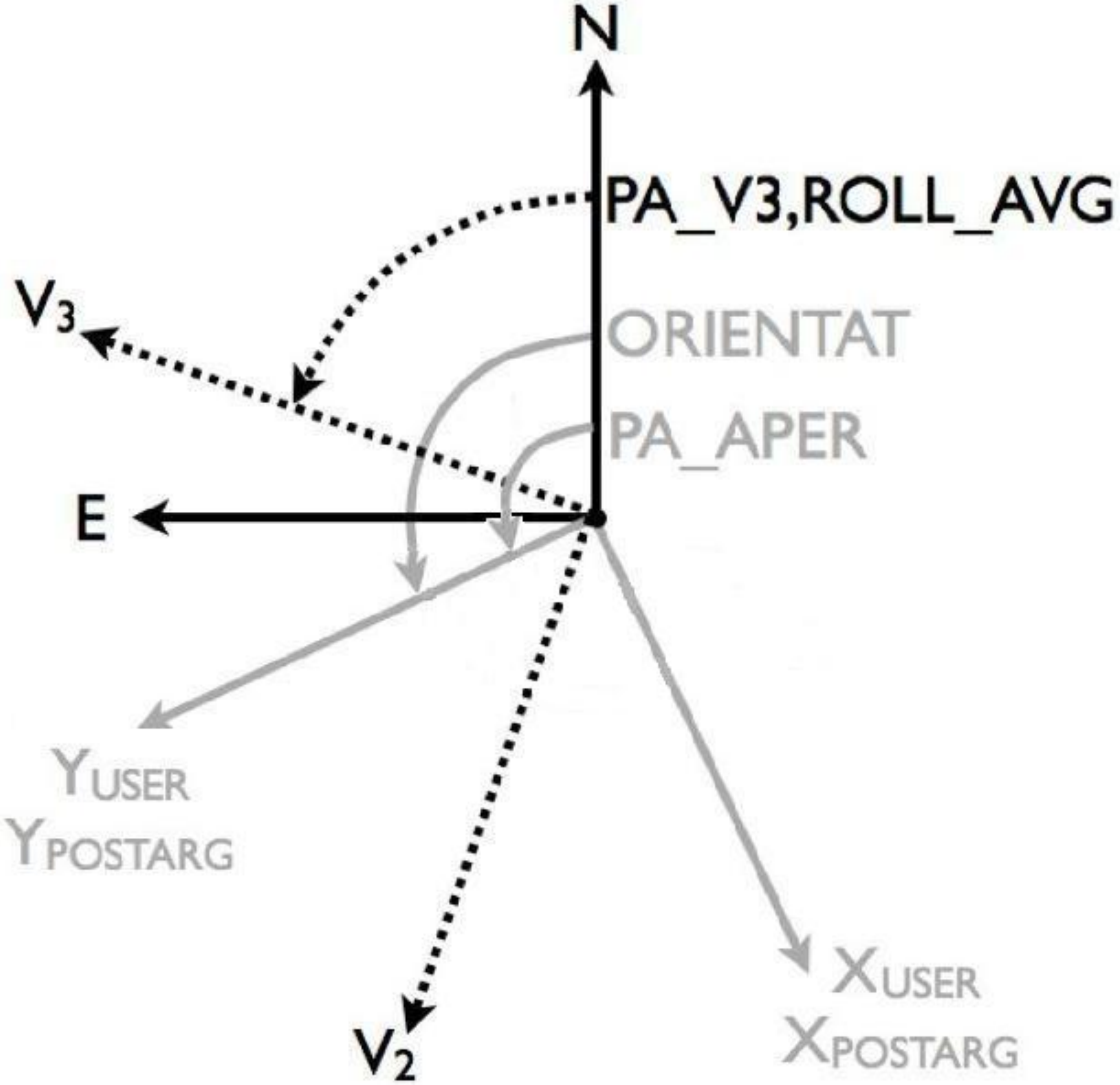
- `PA_V3`: The position angle of the V_3 axis; the angle from North, towards East, to V_3, measured at the center of the *HST* focal plane (in the `SPT` header).
- `ROLL_AVG`: The average angle from North towards East to V_3, measured at the position of the COS field in the *HST* focal plane (in the `JIT` header, computed).
- `PA_APER`: The angle from North through East to Y_{POSTARG} measured at the aperture reference (in the science header).
- `ORIENTAT`: The angle from North through East to Y_{user} measured at the aperture reference (in science header). For COS, `PA_APER` and `ORIENTAT` are equal, i.e., $Y_{\mathrm{POSTARG}} = Y_{\mathrm{USER}}$. *Note that this is not the same angle as the `ORIENT` specified in Phase II, which gives the position angle of the U_3 axis, where $U_3 = -V_3$.*

Refer to ISR TEL 2008-02 for a complete discussion of the COS reference frame geometry.

Detector coordinates are related to user coordinates as follows:

NUV channel: $X_{user} = 1023 - Y_{detector}$; $Y_{user} = 1023 - X_{detector}$
FUV channel: $X_{user} = 16383 - X_{detector}$; $Y_{user} = Y_{detector}$

Figure 1.11: COS Coordinate Systems.



Chapter 2: COS Data Files

Chapter Contents

2.1 Overview

Raw COS telescope data are processed through the STScI data processing (**DP**) pipeline. The **DP** pipeline first processes the data through Generic Conversion, where the data bits from individual exposures are unpacked and combined into files containing raw, uncalibrated data. Next, the data are processed through the COS calibration pipeline, **calcos**, which performs image and spectroscopic reduction to produce output files that can be used directly for scientific analysis (see Chapter 3 for a detailed description of **calcos**). Finally, the data are ingested into the Hubble Data Archive (HDA) through the Data Archive and Distribution System (DADS). This system populates a database containing header keywords which is accessible to users via the Mikulski Archive at STScI (MAST). The data (both calibrated and uncalibrated) are then available for distribution by MAST to the user.

Since COS reference files are frequently updated, the *HST* data archive may use different versions depending on the date the datasets are retrieved. In the event of an updated reference file or calibration software, users may re-calibrate their data in one of two ways. The preferred method is for the user to re-retrieve the data from the HDA using the MAST interface. This will provide the most recent cached reductions produced by the standard pipeline with the default settings. Alternatively, users can reprocess the data themselves through **calcos** using the most recent reference files and software (see "Run calcos" in Section 3.6.1). *The second option will not include any changes in the data due to Generic Conversion updates, but will allow a customized calibration through the use of modified reference files or keyword switches.* Also, the user will need to manually edit the header keywords stating which reference files should be used by **calcos** (Table 2.18 and Section 3.6.1). The most recent reference files can be obtained either from the CRDS website, or via MAST.

Once you have retrieved your data, you will need to understand:

- The naming conventions and file suffixes of the individual files (Section 2.2).
- The basic format in which the COS data are stored (Section 2.3).
- The structure and content of the individual files (Section 2.4).
- The size of the COS data files (Section 2.5).
- How to use the header keywords to identify the principal parameters of an observation and how to determine the calibration processing steps that were performed on a dataset (Section 2.6).
- The meanings of the error and data quality arrays, which are propagated through the pipeline for each COS science observation (Section 2.7).

2.2 COS File Names

The naming convention for COS files is `rootname_*.fits`, where rootname follows the ipppssoot naming convention (see Chapter 2 of the *Introduction to HST Data Handbooks*), and * is a three to nine character file suffix. The suffix identifies the type of data within the file. All FUV data files with the exception of the `lampflash`, `x1d`, and `x1dsum` files will have an additional suffix of `_a` or `_b` (e.g., `rootname_*_[a,b].fits`) to denote the detector segment. However, if `segment=A` is specified in the Phase II proposal there will be no corresponding `_b` files and vice versa. The FUV `lampflash`, `x1d`, and `x1dsum` files will always be segment combined and therefore will not have the additional suffix.

Table 2.1 lists the file suffixes for the COS data files and indicates which files are produced by the different types of observations. Depending on the type of observation, and the path it has taken through the calibration pipeline (see calibration flow charts; Figure 3.1–Figure 3.6), there will be an appropriate subset of these files in a given dataset. Note, the format of some of the COS files can be different depending on the observing mode; see Section 2.3 for more details.

> *COS data utilize a modified naming convention from other HST instruments. In, particular COS FUV files can have TWO suffixes. The first suffix identifies the filetype and the second suffix if present identifies the FUV detector segment. For the remainder of this document the use of "suffix" will refer to the first suffix which identifies the filetype and will always include filetypes with the additional FUV segment suffix if they exist.*

Table 2.1: Data Types and File Naming Conventions for Different COS Observation Types

Long Suffix	Data Format	Spectroscopic				Imaging		Contents
		FUV		NUV		NUV		
		`TIME-TAG`	**`ACCUM`**	**`TIME-TAG`**	**`ACCUM`**	**`TIME-TAG`**	**`ACCUM`**	
	Uncalibrated Science Data							
`rawtag`	table			x		x		Raw NUV `TIME-TAG` events list
`rawtag_a`, `rawtag_b`	table	x						Raw FUV `TIME-TAG` events list
`rawaccum`	image				x		x	Raw NUV `ACCUM` image
`rawaccum_a`, `rawaccum_b`	image		x					Raw FUV `ACCUM` image
`rawacq`	table or image						x	Raw acquisition file
`pha_a`, `pha_b`	image		x					Pulse height distribution

	Uncalibrated Support Data							
`asn`	table	x	x	x	x	x	x	Association file
`jit`	table	x	x	x	x	x	x	Spacecraft pointing data averaged over 3 s intervals
`jif`	image	x	x	x	x	x	x	2-D histogram of the _`jit` file
`spt`	image	x	x	x	x	x	x	Support, planning and telemetry information
`trl`	table	x	x	x	x	x	x	Trailer file with a historical record of generic conversion processing
	Intermediate Data Products							
`trl`	table	x	x	x	x	x	x	The raw trailer file is updated with a historical record and errors log of calibration pipeline processing[1]
`tra, log`	txt	x	x	x	x	x	x	Same as _trl file but in text format
`corrtag`[2]	table			x				NUV `TIME-TAG` events list with calibrated values
`corrtag_a`,[2] `corrtag_b`[2]	table	x						FUV `TIME-TAG` events list with calibrated values
`flt`	image			x	x	x	x	NUV flat-fielded science image
`flt_a`, `flt_b`	image	x	x					FUV flat-fielded science image
`counts`	image			x	x	x	x	NUV not flat-fielded science image
`counts_a`, `counts_b`	image	•	•					FUV not flat-fielded science image
`lampflash`	table	x[3]		x[3]				1-D extracted `TAGFLASH` (`FLASH=yes`) spectra
`x1d`	table	x	x	x	x			1-D extracted spectra for a single exposure
`x1dsum<n>`[4]	table	x	x	x	x			Averaged 1-D extracted spectra for multiple exposures with the same grating, central wavelength, aperture and `FP-POS=<n>`

	Final Data Products							
`fltsum`	image					x	x	Summed flat-fielded image (imaging only). *Final calibrated association product for all COS imaging datasets.*
`x1dsum`	table	x	x	x	x			Final combined 1-D extracted spectra for multiple exposures with the same grating, central wavelength and aperture combining all `FP-POS`. ***Final calibrated association product for all COS spectroscopic datasets***.

---

[1] Only updated during processing and ingestion by the HDA. When reprocessing data in a user's home environment the `trl` file will not be updated. Instead reprocessing will generate an `ASCII` `tra` file.

[2] For `ACCUM` data the time stamps in the first extension are set to the median value in the `corrtag` files; each count in the `rawaccum` file becomes an event. See Section 2.4.2.

[3] Only for `TIME-TAG` with `FLASH=yes` (`TAGFLASH` mode).

[4] <n> can be 1,2,3,4 and denotes the `FP-POS` number.

2.3 COS File Structures

All COS data products are Multi-Extension FITS (MEF) format files and begin with a primary data unit which includes only a header with no data extension. The **info()** function in the **astropy.io.fits Python** module can be used to list the complete set of extensions and their data formats for the COS data files. For more information on using **astropy.io.fits**, please refer to Chapter 5.

2.3.1 COS `FITS` Table Extension Files

Tabular COS information, such as extracted one-dimensional spectra or the `TIME-TAG` mode event series, are stored as FITS binary tables. The tables can be accessed directly in the **stenv** environment using the **astropy.table.Table** module as described in Section 5.1.2 of this document, or with other standard FITS tools.

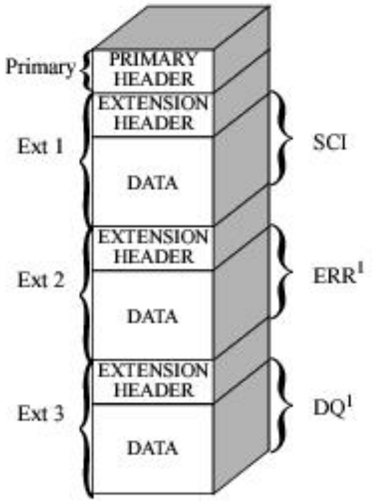
2.3.2 COS `FITS` Image Extension Files

COS images and two-dimensional spectroscopic data are stored in FITS image extension files, which can be directly manipulated, without conversion, in the **stenv** environment. Accessing images in the FITS image extension files in **Python** requires use of the **astropy.io.fits** module and the images can be displayed with the **matplotlib** package. Figure 2.1 illustrates the structure of a COS `FITS` image extension file, which contains:

- A primary header that stores keyword information describing the global properties of the exposure in the file (e.g., the target name, target coordinates, exposure type, optical element, aperture, detector, calibration switches, reference files used).
- A set of image extensions, each containing header keywords with information specific to the given exposure (e.g., exposure time, world coordinate system) and a data array.

Note that not all COS image extension files will contain the ERR and DQ extensions.

Figure 2.1: FITS Image Extension File for COS.



The following file types are stored in `FITS` image extension files with the particular format shown in Figure 2.1: `rawaccum`, flt, counts, pha and rawacq.[1] Each COS readout can generate one `FITS` image SCI extension or three `FITS` image extensions (`SCI`, ERR, and `DQ`) as explained below:

- The first extension type, `SCI`, stores the science values.
- The second extension type, `ERR`, contains the statistical errors, which are propagated through the calibration process. It is unpopulated in raw data files.
- The third extension type, `DQ`, stores the data quality values, which flag suspect pixels in the corresponding `SCI` data.

The error arrays and data quality values are described in more detail in Section 2.7. The value of the EXTNAME keyword in the extension header identifies the type of data the extension contains; the value of this keyword may be determined using the convenience function **getval()** in the **astropy.io.fits** module using **Python**. For example, to see the value of `EXTNAME` of extension 1 of `ldel05ivq_rawtag_a.fits`:

```
> from astropy.io import fits
> fits.getval('ldel05ivq_rawtag_a.fits', 'EXTNAME', ext=1)
```

---

[1] Only `ACQ/IMAGE` files use the exact format shown in Figure 2.1. For more details on acquisition file formats see "Acquisition Files (`RAWACQ`)" in Section 2.4.4.

2.4 COS Data Products

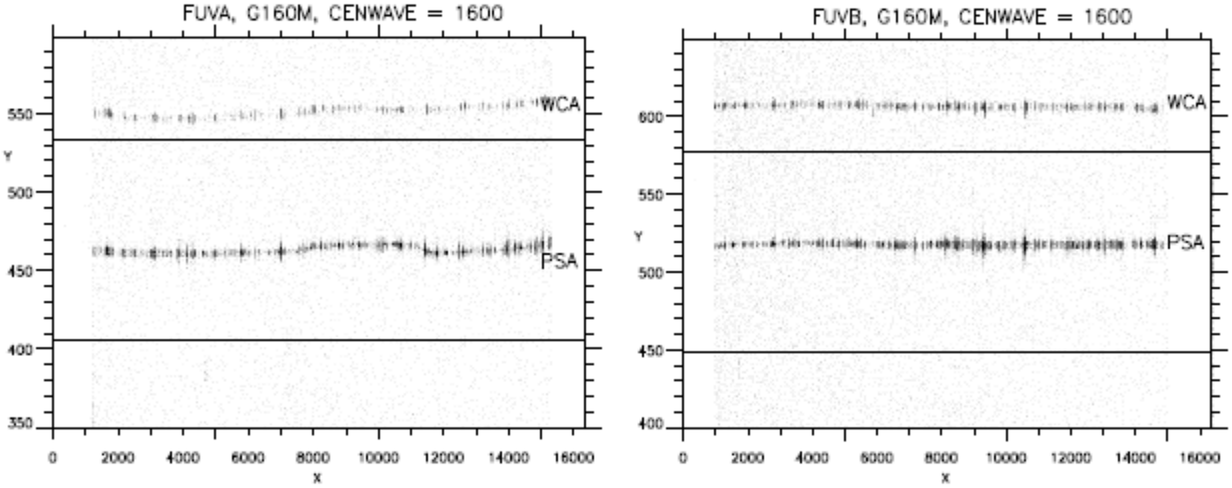
The following sections discuss the COS raw science data files, intermediate calibration products, final calibration products, and auxiliary data files. Uncalibrated science data include all raw science data generated during Generic Conversion that have not been processed through the calibration pipeline. These raw files are the input files to the **calcos** pipeline, usually as part of an association file (see "Association Tables (ASN)" in Section 2.4.4). The pipeline produces both individual calibrated exposure files and, when appropriate, a final combined product file. Note that the final combined product files (x1dsumN, x1dsum) are only produced if calcos is run on the ASN file.

2.4.1 Uncalibrated Science Data Files

Raw ACCUM Images (`rawaccum`)

For `ACCUM` data, the raw files contain a set of images, as shown in Figure 2.1, and have filenames with the suffix `rawaccum` for NUV data, or `rawaccum_a` and `rawaccum_b` for the two segments of the FUV detector. The SCI extension contains an image of the total accumulated counts during an exposure. For NUV data the ERR and DQ extensions have only a header with no data. For FUV data the ERR extension has only a header with no data, and the DQ extension is populated with data quality information only for pixels that are outside the subarray boundaries (defined below). The DQ extensions will be populated in the `flt` files, after calibration pipeline processing. Even though FUV `rawaccum_a[b]` data are 16384 × 1024 images, only portions of them contain actual data. These portions are called subarrays. Typically, three subarrays are used for each segment of an FUV `ACCUM` image. Two are centered on the stim pulse positions and the third is a stripe 128 pixels high which is centered on the wavecal spectrum of the object. Figure 2.2 shows these spectral region subarrays superimposed on two FUV rawtag images. As Figure 2.2 shows, the wavecal spectrum falls outside of the subarray. Consequently, wavecals must be taken separately for `ACCUM` data.

Figure 2.2: Overlay of FUV ACCUM Subarrays on FUV TIME-TAG Data.



The above figures show FUV `TAGFLASH` data for both segments with the corresponding `ACCUM` subarrays noted by the dark lines. The data plotted here are the raw event locations prior to calibration processing. The distortion in the data, particularly for segment A, is very noticeable and discussed further in Section 3.4.5. Note that the full active area in Y is not shown.

Raw `TIME-TAG` Events Lists (`rawtag`)

Raw events tables contain the locations and arrival times of individual photon events collected in `TIME-TAG` mode. These files have the suffix `rawtag` for NUV or `rawtag_a[b]` for the two FUV segments. Figure 2.3 shows the format of a `rawtag` table. The first extension contains the events list, in which each row of the table corresponds to a single event in the data stream and the columns of the table contain scalar quantities that describe the event. The second extension contains the good time intervals (GTI) table, where an uninterrupted period of time is considered as one good time interval. Interruptions in the data taking due to memory overflow could result in more than one GTI. Table 2.2 shows the columns of a `rawtag` table.

Figure 2.3: FITS File Format for Raw and corrected `TIME-TAG` Tables.

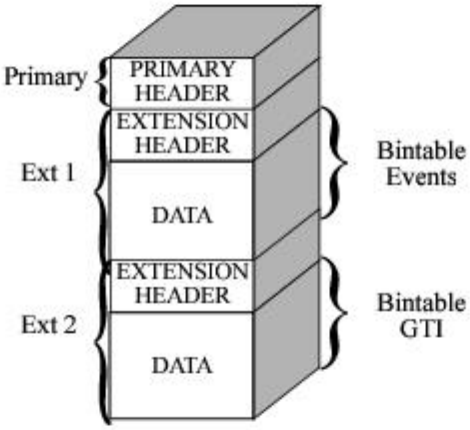


Table 2.2: Columns of a Raw `TIME-TAG` Data Table.

Extension 1			
Column Name	**Units**	**Data Type**	**Description**
TIME	sec	float	Elapsed time in seconds since the exposure start time
RAWX	pixel	integer	Pixel coordinate along the dispersion axis
RAWY	pixel	integer	Pixel coordinate along the cross-dispersion axis
PHA[1]		byte	Pulse height amplitude (0–31)
Extension 2			
Column Name	**Units**	**Data Type**	**Description**
START	sec	float	Start good time interval since exposure start
STOP	sec	float	End good time interval

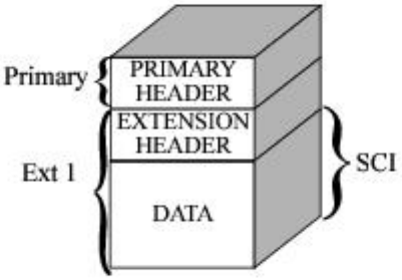
[1] The PHA column is present in the NUV data *only* for symmetry with the FUV data columns. For NUV data the values in this column are set to 0, since no pulse height amplitudes are available.

For more information on working with `TIME-TAG` data see Section 5.4.

Pulse Height Amplitude Files (`pha`)

For FUV `ACCUM` data only, a 7-bit pulse height amplitude histogram is accumulated in the onboard detector electronics. This information is placed in a file with the suffix `pha`. The pulse-height histogram files contain a primary header with no data and a single FITS image SCI extension containing a histogram of the pulse-height distribution during the exposure. The pulse height amplitude files do not contain an ERR or DQ extension, as shown in Figure 2.4. The pulse height distribution is an image array of length 128, corresponding to the number of photons with pulse height values from 0 to 127, corresponding to the pulse heights of 0–31 available in `TIME-TAG` data.

Figure 2.4: FITS Array Extension File for COS.



2.4.2 Intermediate Science Data Files

Corrected Events Lists (`corrtag`)

The COS pipeline produces corrected `TIME-TAG` events lists and stores them in binary tables with suffix `corrtag`. These files have a main header and three extensions: a corrected events list extension, a good time interval extension, and a timeline table extension, with a format similar to the one shown in Figure 2.3. The first extension of the `corrtag` file is the events table (see Table 2.3) which includes X and Y event locations that have been corrected for thermal and geometric distortions and for walk (see Section 3.4), Doppler shift (from HST's orbital motion), and offsets due to OSM motions in both the dispersion and cross-dispersion directions. It also includes wavelengths associated with events that occur within the active area of the detectors and a data quality (DQ) flag for each event (see Table 2.19). The second extension gives the start and stop times of the good time intervals (as in the `rawtag` file), and the third extension is the timeline table. The timeline table includes second by second values for spacecraft position, solar and target altitude above the horizon, and count rates for the most prominent airglow lines and the background. These observed rates might include counts from other external sources in addition to the ones from the airglow line itself. The data in this extension can be useful for reprocessing `TIME-TAG` data to exclude, for example, daytime data using the **Python** tool **costools.timefilter**, described in Section 5.4.2.

For `ACCUM` data, the `corrtag` files are somewhat different. All of the time stamps in the first extension are set to the median value of the observation. Each count in the `rawaccum` file becomes an event so, for example, a pixel in the `rawaccum` that had 100 counts would have 100 entries in the `corrtag` file. The `RAWX`, `XCORR` and `XDOPP` entries are all the same for NUV data, but can be different for FUV. In addition, `RAWY` and `YCORR` entries will have the same values. However, `XFULL` and `YFULL` can be different. In the timeline extension, the `SHIFT1`, airglow and `DARKRATE` entries are fixed.

Table 2.3: Columns of a COS `corrtag` Table.

Column Name	Units	Data Type	Description
Extension 1			
`TIME`	sec	float	Elapsed time in seconds since the exposure start time
`RAWX`	pixel	integer	Pixel coordinate along dispersion axis (same as in `rawtag` file)

`RAWY`	pixel	integer	Pixel coordinate along cross-dispersion axis (same as in `rawtag` file)
`XCORR`[1]	pixel	float	`RAWX` corrected for thermal and geometric distortion and for walk[1]
`XDOPP`	pixel	float	`XCORR` corrected for HST's Doppler shift and for FUV only distortion
`YCORR`[1]	pixel	float	`RAWY` corrected for thermal and geometric distortion and for walk[1]
`XFULL`	pixel	float	`XDOPP` corrected for offset in the dispersion direction, based on the wavecal spectrum
`YFULL`[2]	pixel	float	`YCORR` corrected for offset in the cross-dispersion direction, based on the wavecal spectrum
`WAVELENGTH`	Angstrom	float	Only events in the active area are assigned wavelengths
`EPSILON`		float	Event weight based on flat field and deadtime
`DQ`		integer	Data quality flag
`PHA`[3]		byte	Pulse height amplitude
Extension 2			
`START`	sec	float	Start good time interval since exposure start
`STOP`	sec	float	End good time interval
Extension 3			
`TIME`	sec	float	Time in 1 sec intervals from first entry
`LONGITUDE`	degrees	float	Earth based longitude
`LATITUDE`	degrees	float	Earth based latitude
`SUN_ALT`	degrees	float	Altitude of the sun above the geometric horizon
`SUN_ZD`	degrees	float	Angle between *HST* and the Sun, seen from the center of Earth
`TARGET_ALT`	degrees	float	Altitude of the target above the geometric horizon
`RADIAL_VEL`	km/s	float	Instantaneous *HST* radial velocity toward the target
`SHIFT1`	pixels	float	Instantaneous dispersion direction shift (stripe B for NUV)
`LY_ALPHA`	counts/s	float	Total counts/sec in a box across the aperture at Ly-alpha
`OI_1304`	counts/s	float	Total counts/sec in a box across the aperture at O I 1304
`OI_1356`	counts/s	float	Total counts/sec in a box across the aperture at O I 1356
`DARKRATE`	counts/s	float	Counts/sec/pixel averaged over both background regions

[1] The `XCORR` and `YCORR` columns are present in the NUV data only for symmetry with FUV data. Currently no distortion correction is applied to NUV data, so for NUV data the `XCORR` and `YCORR` columns are identical to the `RAWX` and `RAWY` columns.

[2] For FUV data extracted with the TWOZONE method, `YFULL` is now also corrected for the spectrum trace and offset from the template profile (see Section 3.4.14).

[3] The `PHA` column is present in the NUV data only for symmetry with the FUV data columns. For NUV data this column is set to a default value of 0, since no pulse height amplitudes are available for NUV.

Lampflash Files (`lampflash`)

For `TAGFLASH` data, **calcos** produces an events list with suffix `lampflash`, that contains the extracted wavecal lamp flashes. Each row in the events list corresponds to a different segment or stripe and flash number (the first flash is number 1, the second is number 2, etc.). The `lampflash` files have the format shown in Figure 2.5 . The contents of the columns in a `lampflash` events list are listed in Table 2.4. Columns `TIME`, `LAMP_ON`, and `LAMP_OFF` have the same temporal zero point as the `TIME` column of the `rawtag` and `corrtag` tables and the same unit (seconds). The shifts contained in the `SHIFT_DISP` and `SHIFT_XDISP` columns of the lampflash table are applied to the `XDOPP` and `YCORR` columns of the corrtag file to produce the `X[Y]FULL` entries. When multiple `TAGFLASHES` are present, the shifts are interpolated in time for events occurring between each set of flashes. Events occurring before the first flash are shifted by a value extrapolated using the slope defined by the first two flashes; events beyond the last flash are given the shift determined by the last flash. As a result, the difference between the `X[Y]FULL` and `X[Y]CORR` entries in the `corrtag` file can be a function of time.

As noted in the table below, the time column provides a median value. Lamps are flashed for a fixed length, let's say 12s. As an example, if you have flash 1 that starts at 0 seconds and lasts 12 seconds, the median time is 5.5. The next flash in the exposure then executes at 600 seconds and lasts 12 seconds, so the median time of the flash is 605.5. So your time array would be (5.5, 605.5, 5.5, 605.5), repeated because one is for FUVA and one is for FUVB.

Figure 2.5: FITS File Format for Lampflash Table.

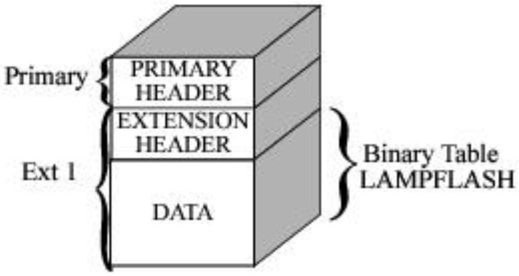


Table 2.4: Columns of a COS Lampflash Table.

Column Name	Units	Data Type	Description
`SEGMENT`		String	FUV segment(s) or NUV stripe(s) corresponding to the extracted tagflash wavecal
`TIME`	sec	double	Median time of each flash
`EXPTIME`	sec	double	Duration of each flash in seconds
`LAMP_ON`	sec	double	Lamp turn on time for each flash, counting from start of exposure
`LAMP_OFF`	sec	double	Lamp turn off time for each flash, counting from start of exposure
`NELEM`		integer	Length of the `WAVELENGTH`, `GROSS`, `NET`, `BACKGROUND`, and `DQ`, `DQ_WGT` arrays
`WAVELENGTH`	Å	double [nelem]	Wavelengths of each extracted tagflash wavecal spectrum(s)
`GROSS`	counts s^{-1}	float [nelem]	Gross count rate of each extracted tagflash wavecal spectrum(s)
`NET`	counts s^{-1}	float [nelem]	Net count rate of each extracted tagflash wavecal spectrum(s)
`BACKGROUND`	counts s^{-1}	float [nelem]	Background count rate of each extracted tagflash wavecal spectrum (s)
`SHIFT_DISP`	pixel	float	Dispersion direction shift(s) determined by comparing each tagflash wavecal with a wavecal template
`SHIFT_XDISP`	pixel	float	Cross-dispersion direction shift(s) determined by comparing each tagflash wavecal with a wavecal template
`CHI_SQUARE`		float	Chi square of comparison between tagflash wavecal and wavecal template

`N_DEG_FREEDOM`		integer	Number of degrees of freedom in chi square comparison
`SPEC_FOUND`		boolean	T (true) or F (false), if each tagflash wavecal spectrum was found or not

Counts Files (`counts`)

The `counts` images are an intermediate calibrated output product for both imaging and spectroscopic data with suffix `counts`. These files contain three extensions (SCI, ERR, and DQ) as shown in Figure 2.1. These files are constructed by summing up the events from each pixel using the `XFULL` and `YFULL` coordinates. The data are in units of counts per pixel. For FUV data the images are 16384 columns in the x (dispersion) direction by 1024 rows in the y (cross-dispersion) direction. The NUV images are 1274 columns in the x direction by 1024 rows in the cross-dispersion direction for spectroscopic data, and 1024 × 1024 for data obtained in imaging mode. The NUV spectroscopic files have more pixels in the dispersion direction than the actual NUV detector. This is because the `counts` files (and `flt` files) have been corrected for Doppler shift and OSM shift (including `FP-POS` offset), so the width was increased to accommodate those shifts. The FUV images are not extended since the active area is less than the size of the detector, so these effects can be incorporated into the images without the need to extend them. The FUV data are also corrected for walk and geometric distortions.

Flat-Fielded Image Files (`flt`)

For spectroscopic data a flat-fielded image is an intermediate calibrated data file. These files have a suffix, `flt`, and contain three extensions (`SCI`, `ERR`, and `DQ`) as shown in Figure 2.1. These files are constructed by summing up the values in the EPSILON column for each pixel using the `XFULL` and `YFULL` coordinates. The data are in units of the count rate. For FUV data the images are 16384 × 1024, and, like the `counts` images, the NUV images are 1274 × 1024 for spectroscopic data and 1024 ×1024 for data obtained in imaging mode. The `flt` images are corrected for deadtime effects. The NUV images are corrected for all flat-field effects and the FUV data are currently corrected for only the largest fixed-pattern features; the `XDL` grid-wire shadows, low-order flat-field variations (L-flats), and large geometric distortion artifacts.

2.4.3 Final Science Data Files (and Product Files)

The initial input files to **calcos** are the association tables with suffix `asn`. These files provide the calibration pipeline with information about how the data files are associated. In general, only exposures taken in sequence with the same spectral element, central wavelength (if applicable), and aperture at any `FP-POS` will be associated. For more information on COS association files see the "Association Tables (ASN)" portion of Section 2.4.4.

Processing of each individual exposure in the association produces a final calibrated result named with exposure rootname and suffix `x1d` (spectroscopy) or `flt` (imaging).

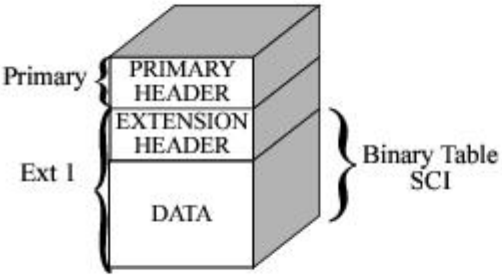
Next, for each `FP-POS` position <n> (where <n>=1, 2, 3, or 4), if there are *multiple spectroscopic* exposures in the association that use the same `FP-POS` position, **calcos** will combine their respective `x1d` into a file named with the association rootname and suffix `x1dsum`<n>, where <n> is the integer `FP-POS` value and the DQ_WGT is applied to the final, combined spectrum. If there is a *single* exposure with a given `FP-POS` value in the association, the `x1dsum`<n> file contains the `x1d` spectrum to which the DQ_WGT is applied (see Section 3.4.22).

Lastly, a final association product file is produced with association rootname and suffix `x1dsum` (spectroscopy) or `fltsum` (imaging) by combining *all science exposures* in the association.

One-Dimensional Extracted Spectra (`x1d`, `x1dsum`)

The COS pipeline produces extracted one-dimensional spectra and stores them in binary tables with suffix `x1d`, `x1dsum`<n>, or `x1dsum`. Figure 2.6 shows the format of the 1-D extracted spectra table.

Figure 2.6: FITS File Format for 1-D Extracted Spectrum Table.



These COS extracted spectra tables can be 1- to 3-Dimensional, with one row for each unique segment or stripe. For FUV data there are typically two rows which correspond to segments A and B distinguished by "FUVA" and "FUVB" in the SEGMENT column respectively. For NUV data there are three rows, "NUVA," "NUVB," and "NUVC" corresponding to stripes A, B, and C, respectively. Each table column can contain either a scalar value or an array of values, such as WAVELENGTH or FLUX. For example, NELEM will contain a scalar number, while the WAVELENGTH column will contain an array of wavelengths. Table 2.5 shows the contents of the different columns in an extracted spectrum table. A discussion of the data in COS extracted spectra is provided in Section 3.4.18.

Table 2.5: Columns of a COS Extracted Spectrum Table.

Column Name	Units	Data Type	Description
SEGMENT		string	FUV segments or NUV stripe names
EXPTIME	seconds	float	Corrected exposure times for each segment
NELEM		integer	Length of the array fields, such as the WAVELENGTH and GROSS arrays[1]
WAVELENGTH[1]	Å	double [nelem]	Wavelengths corresponding to fluxes
FLUX[1]	erg s^{-1} cm^{-2} $Å^{-1}$	float [nelem]	Flux calibrated NET spectrum
ERROR[1]	erg s^{-1} cm^{-2} $Å^{-1}$	float [nelem]	Upper bound of internal error estimate
ERROR_LOWER	erg s^{-1} cm^{-2} $Å^{-1}$	float [nelem]	Lower bound of internal error estimate
VARIANCE_FLAT		float [nelem]	Term used for calculating the internal error due to the flat-field error.

VARIANCE_COUNTS		float [nelem]	Term used for calculating the internal error due to the source counts.
VARIANCE_BKG		float [nelem]	Term used for calculating the internal error due to the background counts.
GROSS	counts s^{-1}	float [nelem]	Gross extracted spectrum count rate
NET[1]	counts s^{-1}	float [nelem]	Difference of GROSS and BACKGROUND[1] arrays
BACKGROUND	counts s^{-1}	float [nelem]	Background count rate
GCOUNTS	counts	float [nelem]	Gross counts
DQ_WGT		float [nelem]	Number of sub-exposures that contribute to each pixel in the combined spectrum.
DQ		short [nelem]	Logical OR of data quality flags in extraction region
DQ_OUTER		short [nelem]	Data quality flag in outer extraction region for spectra extracted with TWOZONE
BACKGROUND_PER_PIXEL	counts s^{-1}	float [nelem]	Average background per pixel
NUM_EXTRACT_ROWS		integer	Number of extracted rows
ACTUAL_EE		double [nelem]	Actual energy enclosed between outer zone boundaries
Y_LOWER_OUTER		double [nelem]	Index of lower outer extraction zone boundary
Y_LOWER_INNER		double [nelem]	Index of lower inner extraction aperture boundary
Y_UPPER_OUTER		double [nelem]	Index of upper outer extraction zone boundary
Y_UPPER_INNER		double [nelem]	Index of upper inner extraction zone boundary

[1] Note that in the x1dsum & x1dsumN files, the DQ_WGT column is applied, while in the x1d files it is not.

Flat-Fielded Image Files (`flt`, `fltsum`)

For NUV imaging observations, the `flt` and `fltsum` images are the final data products, with the latter being a simple sum of the individuals when several exposures are processed together. They are fully linearized and flat-field corrected images. Unlike the `flt` files produced for the spectroscopic data (which are intermediate data products with a format of 1274 × 1024, see Section 2.4.2), the formats of the `flt` and `fltsum` files for imaging data are 1024 × 1024, since Doppler and OSM motions are not applied.

2.4.4 Auxiliary Data Files

Association Tables (`ASN`)

An association file is created for all COS observation sets, and has the suffix asn (e.g., `lcwj01010_asn.fits`). This file holds a single binary table extension, which can be displayed with the **astropy.table.Table** module.

Calcos calibrates raw data from multiple science exposures and any contemporaneously obtained line lamp calibration exposures through the pipeline as an associated unit. Each individual science exposure in an *association* is fully calibrated in the process. The information within an association table shows how a set of exposures are related, and informs the COS calibration pipeline how to process the data.

An example association table is shown below. Note that all related COS exposures will be listed in an association table, with the exception of acquisitions, darks, and flats. It is possible to have an association which contains only one exposure. The association file lists the rootnames of the associated exposures as well as their membership role in the association. The exposures listed in an association table directly correspond to individual raw FITS files. For example, the association table can describe how wavecal exposures are linked to science exposures. Table 2.6 summarizes the different exposure membership types (`MEMTYPES`) used for COS association tables.

Table 2.6: Member Types in COS Associations.

`MEMTYPE`	Description
`EXP-AWAVE`	Input automatic wavelength calibration exposure
`EXP-FP`	Input science exposure
`EXP-GWAVE`	Input GO wavelength calibration exposure
`PROD-FP`	Output science product

The table below illustrates the contents of the association table for a sequence of spectroscopic exposures for four `FP-POS` positions.

Sample Association Table l9v221010_asn. To display the association table for **ldel05050_asn.fits**:

```
> from astropy.io import fits
> fits.info('ldel05050_asn.fits')
Filename: ldel05050_asn.fits
No. Name  Ver Type     Cards Dimensions Format
  0 PRIMARY 1 PrimaryHDU  43 ()
  1 ASN     1 BinTableHDU 25 5R × 3C    [14A, 14A, L]
> from astropy.table import Table
> t = Table.read('ldel05050_asn.fits', hdu=1)
> print(t)

MEMNAME   MEMTYPE MEMPRSNT
--------- ------- --------
LDEL05JYQ EXP-FP  True
LDEL05K0Q EXP-FP  True
LDEL05K2Q EXP-FP  True
LDEL05K4Q EXP-FP  True
LDEL05050 PROD-FP True
```

In the above example, MEMTYPE describes the exposure membership type or role in the association. The column MEMPRSNT lists whether the member is present or not. The association file can be modified to not include a member during processing by changing the MEMPRSNT to ‘false.’

The association table above lists the names of the four associated exposures that are calibrated and combined to create the various association products which will have a rootname of `ldel05050`. This particular association is created from a single `TIME-TAG` spectroscopic APT specification with `FP-POS=ALL` specified in the Phase II file, which leads to a science exposure taken at each `FP-POS` location. For example, the first entry in the table, `ldel05jyq`, is the rootname of a single external science exposure taken with `FP-POS=1`. This exposure corresponds to the following `rawtag` files: `ldel05jyq_rawtag_a.fits`, `ldel05jyq_rawtag_b.fits`. The memtype of this exposure is `EXP-FP` which shows that it is an external exposure. Similar files correspond to the remaining three entries in the association file for data taken with the remaining three `FP-POS` positions. The pipeline will calibrate the members of an association as a unit, producing the calibrated data products for each individual exposure as well as the final combined association data product. For this particular association, the pipeline will produce a final combined association product, `ldel05050_x1dsum.fits`, which contains the final `FP-POS` combined, calibrated spectrum.

Trailer Files (`TRL`)

When COS data are processed in the HDA, the output messages from generic conversion and the different calibration steps are stored in a FITS `ASCII` table known as the trailer file, with suffix `trl`. Files ending with `_log.txt` are identical to files ending in `trl`, but are instead in TXT format.Each time the archive processes data, the old trailer file is erased and a new one created using the results of the most recent processing performed. The archive will produce a trailer file for each individual exposure and association product. Association product trailer files contain the appended information from all the exposures in the association, in order of processing. The order of processing is the same as the order of exposures in the association table, with the exception of `AUTO` or `GO` wavecals which are always processed first.

In the trailer files from the HDA, the output messages from generic conversion appear first in the file. This section contains information relevant to the selection of the best reference files and the population of some of the header keywords. The second part of this file contains information from **calcos** processing. Each task in the **calcos** pipeline creates messages during processing which describe the progress of the calibration, and appear in the order in which each step was performed. These messages are relevant to understand how the data were calibrated, and in some cases, to determine the accuracy of the products.

⚠ ***It is highly recommended to always examine the trailer files for errors or warnings.***

In the last section of the `_trl` file, the **calcos** steps are indicated by their module name. The **calcos** messages provide information on the input and output files for each step, the corrections performed, information regarding the reference files used, and in the case of FUV data, messages about the location of the stim pulses, or shift correction applied to the data. **Calcos** also gives warnings when the appropriate correction to the data could not be applied. For more detailed information on the calibration steps and structure of **calcos**, please refer to Chapter 3.

Calcos Trailer Files (`TRA`)

When **calcos** is run on a personal machine, **calcos** redirects the output of its steps to the STDOUT and an `ASCII` file with name `rootname.tra`. Note, the level of detail included in the output messages can be modified when running **calcos** (see about "verbosity" in "Run **calcos**"). So, when run on a personal machine, **calcos** will *not* overwrite the `trl` file but rather will direct the output to STDOUT and an `ASCII tra` file. The `tra` file is formatted like the `trl` file but with two exceptions: the `tra` file will not contain the output messages from generic conversion, and the `tra` file is not converted to FITS format. Each time **calcos** is run on a file, the STDOUT messages will be appended to the `tra` file if it already exists. Also, when running **calcos** on a personal machine there will be no `tra` created for the association products (ASN files). Instead, the **calcos** messages for association products will be sent only to STDOUT.

Support Files (SPT)

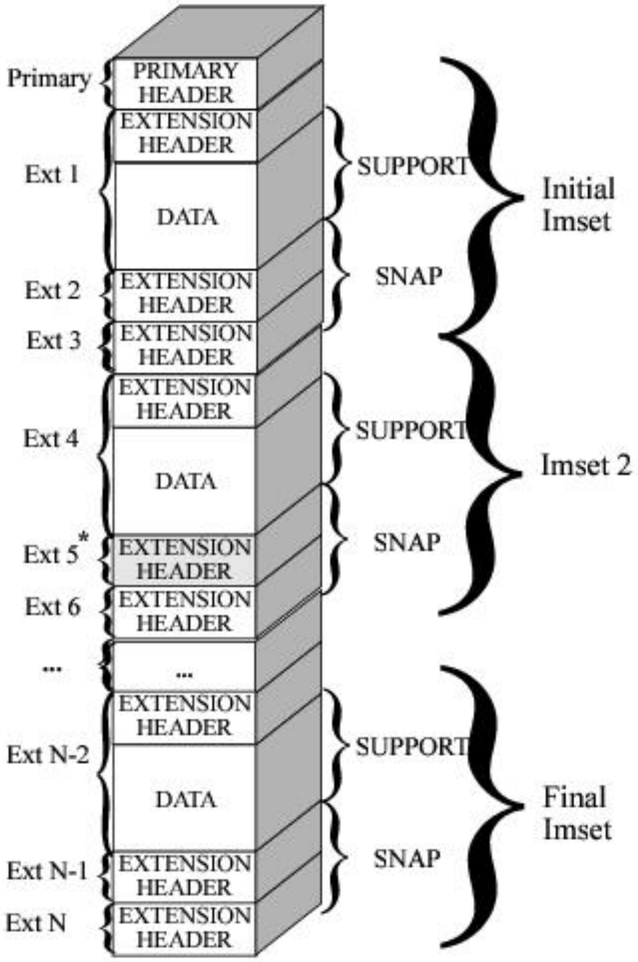
The support files contain information about the observation and engineering data from the instrument and spacecraft that were recorded at the time of the observation. A COS support file contains a primary header and a variable number of image extensions. Depending on the length of the exposure, the support file will contain one or more "imsets," each of which includes a support extension (EXTNAME = 'SUPPORT") and two snap extensions (EXTNAME = 'SNAP1' and 'SNAP2'):

The SUPPORT extension contains a header with the proposal information and an (16-bit) image array containing the data which populate the SPT header keyword values. The image array element values are used to populate the header keywords.

Following the support extension in each imset, there are two engineering snapshot extensions. These extensions contain samples of many instrument and telescope parameters from telemetry data during the course of the exposure. The SNAP1 extension in the first imset will always contain telemetry information collected immediately before the exposure begins. Subsequent SNAP1 extensions in all imsets will repeat the information contained in the first one. The SNAP2 extension in each imset will be populated with telemetry data taken during the course of the exposure. The final SNAP2 extension will be populated with data taken immediately after the completion of the science exposure.

Figure 2.7 depicts the structure of an N extension COS support file. With several snapshots of COS telemetry values, one may track the instrument status periodically throughout an exposure. For a schematic listing of the spt headers with detailed information about the spt header keywords, see:
http://stdatu.stsci.edu/keyword/cgi-bin/kdct-header.cgi?i=COS&s=20.1&db=Operational.

Figure 2.7: COS Support File.



* Extension 5, is not populated, and therefore all header keywords in this extension will be set to a default. Every other snapshot extension from extension 5 through N-4, will also not be populated.

COS support file with N extensions. The initial imset contains telemetry values at the start of the exposure. For subsequent imsets, only the second snap extension contains valid data.

Acquisition Files (`RAWACQ`)

All COS acquisition exposures will produce a single raw data file with suffix `rawacq`. Almost all COS spectroscopic science exposures are preceded by an acquisition sequence or exposure to center the target in the aperture. Keywords in the header of COS science data identify the exposure names of relevant acquisition exposures in each visit. In addition, there are several other useful keywords in the COS acquisition exposures that describe the acquisition parameters used, as well as the calculated centroid positions and slew offsets. Table 2.7 lists all the relevant acquisition keywords.

Table 2.7: ACQ/IMAGE Header Keywords.

Keyword Name	Description
`ACQSNAME`[1]	Rootname of first acquisition search exposure
`ACQINAME`[1]	Rootname of first acquisition image exposure
`PEAKXNAM`[1]	Rootname of first cross-dispersion (XD) peakup exposure
`PEAKDNAM`[1]	Rootname of first along-dispersion (AD) peakup exposure
`ACQ_NUM`[1]	Total number of exposures in acquisition sequence
`LAMPSTAT`	Status of Wavecal lamp exposure (LTAIMCAL)
`LAMPTIME`	Lamp exposure integration time(s)
`LAMPMXCR`	Measured centroid of lamp exposure in X (AD)
`LAMPMYCR`	Measured centroid of lamp exposure in Y (XD)
`LAMPEVNT`	Number of events in the lamp exposure
`LAMPCNTR`	Lamp Centering method
`LMPSUBX1`	X coordinate of the left of the lamp subarray (pixels)
`LMPSUBX2`	X coordinate of the right of the lamp subarray (pixels)
`LMPSUBY1`	Y coordinate of the top of the lamp subarray (pixels)
`LMPSUBY2`	Y coordinate of the bottom of the lamp subarray (pixels)
`ACQSTAT`	Status of the acquisition exposure (LTAIMAGE)
`TARGTIME`	Acquisition exposure integration time(s)
`ACQCENTX`	Measured target centroid in X (AD) direction
`ACQCENTY`	Measured target centroid in Y (XD) direction
`WCA2SCIX`	WCA to science Aperture offset in X (AD)
`WCA2SCIY`	WCA to Science aperture offset in Y (XD)
`ACQPREFX`	Desired target X (AD) position
`ACQPREFY`	Desired target Y (XD) position
`ACQSLEWX`	Slew offset in X (AD) (arcseconds)
`ACQSLEWY`	Slew offset in Y (XD) (arcseconds)
`TRGSUBX1`	X coordinate of the left of the target subarray (pixels)

`TRGSUBX2`	X coordinate of the right of the target subarray (pixels)
`TRGSUBY1`	Y coordinate of the left of the target subarray (pixels)
`TRGSUBY2`	Y coordinate of the right of the target subarray (pixels)

[1] These keywords are also found in the COS science headers in addition to being in the acquisition headers.

Table 2.8: ACQ/SEARCH Header Keywords.

Keyword Name	Description
`ACQSNAME`[1]	Rootname of first acquisition search exposure
`ACQINAME`[1]	Rootname of first acquisition image exposure
`PEAKXNAM`[1]	Rootname of first cross-dispersion (XD) peakup exposure
`PEAKDNAM`[1]	Rootname of first along-dispersion (AD) peakup exposure
`ACQ_NUM`[1]	Total number of exposures in acquisition sequence
`ACQSTAT`	Status of target exposure
`TARGTIME`	Integration time per dwell (s)
`CENTER`	Centering method used by the search
`ACQFLOOR`	Threshold Floor used (for FLUX-WEIGHT-FLOOR centering method)
`SCANSIZE`	Number of dwells per side of the square pattern
`ACQNPOS`	Total number of dwells
`STEPSIZE`	Scan step size between dwells (arcsec)
`ENDSLEWX`	Commanded X-direction (AD) slew from the final dwell point (arcsec)
`ENDSLEWY`	Commanded Y-direction (XD) slew from the final dwell point (arcsec)
`ACQSLEWX`	Commanded X-direction (AD) slew from the center of the search pattern (arcsec)
`ACQSLEWY`	Commanded Y-direction (XD) slew from the center of the search pattern (arcsec)
`SEGMENT`[2]	FUV Segment used
`TRGSUBX1`[3]	X coordinate of the left of the target subarray (pixels)
`TRGSUBX2`[3]	X coordinate of the right of the target subarray (pixels)
`TRGSUBY1`[3]	Y coordinate of the top of the target subarray (pixels)
`TRGSUBY2`[3]	Y coordinate of the bottom of the target subarray (pixels)

`TRGAS1X1`[2]	X coordinate of the left of the first segment A target subarray
`TRGAS1X2`[2]	X coordinate of the right of the first segment A target subarray
`TRGAS1Y1`[2]	Y coordinate of the top of the first segment A target subarray
`TRGAS1Y2`[2]	Y coordinate of the bottom of the first segment A target subarray
`TRGBS1X1`[2]	X coordinate of the left of the first segment B target subarray
`TRGBS1X2`[2]	X coordinate of the right of the first segment B target subarray
`TRGBS1Y1`[2]	Y coordinate of the top of the first segment B target subarray
`TRGBS1Y2`[2]	Y coordinate of the bottom of the first segment B target subarray
`TRGAS2X1`[2]	X coordinate of the left of the second segment A target subarray
`TRGAS2X2`[2]	X coordinate of the right of the second segment A target subarray
`TRGAS2Y1`[2]	Y coordinate of the top of the second segment A target subarray
`TRGAS2Y2`[2]	Y coordinate of the bottom of the second segment A target subarray
`TRGBS2X1`[2]	X coordinate of the left of the second segment B target subarray
`TRGBS2X2`[2]	X coordinate of the right of the second segment B target subarray
`TRGBS2Y1`[2]	Y coordinate of the top of the second segment B target subarray
`TRGBS2Y2`[2]	Y coordinate of the bottom of the second segment B target subarray

[1] These keywords are also found in the COS science headers in addition to being in the acquisition headers.

[2] FUV only.

[3] NUV only.

Table 2.9: ACQ/PEAKXD Header Keywords.

Keyword Name	Description
`ACQSNAME`[1]	Rootname of first acquisition search exposure
`ACQINAME`[1]	Rootname of first acquisition image exposure
`PEAKXNAM`[1]	Rootname of first cross-dispersion (XD) peakup exposure
`PEAKDNAM`[1]	Rootname of first along-dispersion (AD) peakup exposure
`ACQ_NUM`[1]	Total number of exposures in acquisition sequence
`LAMPSTAT`	Status of lamp exposure (LTACAL)

`LAMPTIME`	Integration time of lamp exposure(s)
`LAMPMYCR`	Measured centroid of lamp exposure in Y (AD)
`LAMPEVNT`	Number of events in lamp exposure
`LAMPCNTR`	Lamp Centering Method
`LSTRIPE`[2]	NUV Lamp Stripe used for target acquisition
`LMPSUBX1`[2]	X coordinate of the left of the lamp subarray (pixels)
`LMPSUBX2`[2]	X coordinate of the right of the lamp subarray (pixels)
`LMPSUBY1`[2]	Y coordinate of the top of the lamp subarray (pixels)
`LMPSUBY2`[2]	Y coordinate of the bottom of the lamp subarray (pixels)
`LMPAS1X1`[3]	X coordinate of the left of the first segment A lamp subarray
`LMPAS1X2`[3]	X coordinate of the right of the first segment A lamp subarray
`LMPAS1Y1`[3]	Y coordinate of the top of the first segment A lamp subarray
`LMPAS1Y2`[3]	Y coordinate of the bottom of the first segment A lamp subarray
`LMPBS1X1`[3]	X coordinate of the left of the first segment B lamp subarray
`LMPBS1X2`[3]	X coordinate of the right of the first segment B lamp subarray
`LMPBS1Y1`[3]	Y coordinate of the top of the first segment B lamp subarray
`LMPBS1Y2`[3]	Y coordinate of the bottom of the first segment B lamp subarray
`LMPAS2X1`[3]	X coordinate of the left of the second segment A lamp subarray
`LMPAS2X2`[3]	X coordinate of the right of the second segment A lamp subarray
`LMPAS2Y1`[3]	Y coordinate of the top of the second segment A lamp subarray
`LMPAS2Y2`[3]	Y coordinate of the bottom of the second segment A lamp subarray
`LMPBS2X1`[3]	X coordinate of the left of the second segment B lamp subarray
`LMPBS2X2`[3]	X coordinate of the right of the second segment B lamp subarray
`LMPBS2Y1`[3]	Y coordinate of the top of the second segment B lamp subarray
`LMPBS2Y2`[3]	Y coordinate of the bottom of the second segment B lamp subarray
`ACQSTAT`	Status of target exposure (LTAPKXD)
`TARGTIME`	Acquisition exposure integration time(s)
`ACQMEASY`	Measured target centroid in Y (XD) direction

`ACQPREFY`	Desired computed Y position
`ACQSLEWY`	Slew offset in Y (XD) (arcsec)
`TARGEVNT`	Number of events in the acquisition exposure
`STRIPE`[2]	NUV Stripe used for target acquisition
`SEGMENT`[3]	FUV detector segment name (FUVA or FUVB or BOTH)[4]
`TRGSUBX1`[2]	X coordinate of the left of the target subarray (pixels)
`TRGSUBX2`[2]	X coordinate of the right of the target subarray (pixels)
`TRGSUBY1`[2]	Y coordinate of the top of the target subarray (pixels)
`TRGSUBY2`[2]	Y coordinate of the bottom of the target subarray (pixels)
`TRGAS1X1`[3]	X coordinate of the left of the first segment A target subarray
`TRGAS1X2`[3]	X coordinate of the right of the first segment A target subarray
`TRGAS1Y1`[3]	Y coordinate of the top of the first segment A target subarray
`TRGAS1Y2`[3]	Y coordinate of the bottom of the first segment A target subarray
`TRGBS1X1`[3]	X coordinate of the left of the first segment B target subarray
`TRGBS1X2`[3]	X coordinate of the right of the first segment B target subarray
`TRGBS1Y1`[3]	Y coordinate of the top of the first segment B target subarray
`TRGBS1Y2`[3]	Y coordinate of the bottom of the first segment B target subarray
`TRGAS2X1`[3]	X coordinate of the left of the second segment A target subarray
`TRGAS2X2`[3]	X coordinate of the right of the second segment A target subarray
`TRGAS2Y1`[3]	Y coordinate of the top of the second segment A target subarray
`TRGAS2Y2`[3]	Y coordinate of the bottom of the second segment A target subarray
`TRGBS2X1`[3]	X coordinate of the left of the second segment B target subarray
`TRGBS2X2`[3]	X coordinate of the right of the second segment B target subarray
`TRGBS2Y1`[3]	Y coordinate of the top of the second segment B target subarray
`TRGBS2Y2`[3]	Y coordinate of the bottom of the second segment B target subarray

[1] These keywords are also found in the COS science headers in addition to being in the acquisition headers.

[2] NUV only.

[3] FUV only.

Table 2.10: ACQ/PEAKD Header Keywords.

Keyword Name	Description
`ACQSNAME`[1]	Rootname of first acquisition search exposure
`ACQINAME`[1]	Rootname of first acquisition image exposure
`PEAKXNAM`[1]	Rootname of first cross-dispersion (XD) peakup exposure
`PEAKDNAM`[1]	Rootname of first along-dispersion (AD) peakup exposure
`ACQ_NUM`[1]	Total number of exposures in acquisition sequence
`ACQSTAT`	Status of acquisition (LTAPKD)
`TARGTIME`	Acquisition exposure integration time(s)
`CENTER`	Centering method used
`ACQFLOOR`	Threshold floor value
`ACQNPOS`	Number of dwells in the acquisition
`STEPSIZE`	Peakup scan stepsize (arcsec)
`ACQMEASX`	Measured target centroid in X (AD) direction
`ACQPREFX`	Desired computed X (AD) position
`ENDSLEWX`	X (AD) slew from final dwell position (arcsec)
`ACQSLEWX`	Slew offset from center in X (AD) (arcsec)
`SEGMENT`[2]	FUV detector segment name (FUVA or FUVB or BOTH)
`TRGSUBX1`[3]	X coordinate of the left of the target subarray (pixels)
`TRGSUBX2`[3]	X coordinate of the right of the target subarray (pixels)
`TRGSUBY1`[3]	Y coordinate of the top of the target subarray (pixels)
`TRGSUBY2`[3]	Y coordinate of the bottom of the target subarray (pixels)
`TRGAS1X1`[2]	X coordinate of the left of the first segment A target subarray
`TRGAS1X2`[2]	X coordinate of the right of the first segment A target subarray
`TRGAS1Y1`[2]	Y coordinate of the top of the first segment A target subarray
`TRGAS1Y2`[2]	Y coordinate of the bottom of the first segment A target subarray
`TRGBS1X1`[2]	X coordinate of the left of the first segment B target subarray

TRGBS1X2[2]	X coordinate of the right of the first segment B target subarray
TRGBS1Y1[2]	Y coordinate of the top of the first segment B target subarray
TRGBS1Y2[2]	Y coordinate of the bottom of the first segment B target subarray
TRGAS2X1[2]	X coordinate of the left of the second segment A target subarray
TRGAS2X2[2]	X coordinate of the right of the second segment A target subarray
TRGAS2Y1[2]	Y coordinate of the top of the second segment A target subarray
TRGAS2Y2[2]	Y coordinate of the bottom of the second segment A target subarray
TRGBS2X1[2]	X coordinate of the left of the second segment B target subarray
TRGBS2X2[2]	X coordinate of the right of the second segment B target subarray
TRGBS2Y1[2]	Y coordinate of the top of the second segment B target subarray
TRGBS2Y2[2]	Y coordinate of the bottom of the second segment B target subarray

[1] These keywords are also found in the COS science headers in addition to being in the acquisition headers.

[2] FUV only.

[3] NUV only.

PEAKD and SEARCH Acquisitions

Acquisition peakups in the dispersion direction (ACQ/PEAKD) and acquisition spiral searches (ACQ/SEARCH) both use the flux from exposures taken at different dwell points to center the target. For more information on these types of COS acquisitions see Sections 8.6 and 8.3 respectively of the *COS Instrument Handbook*. Data for these acquisitions contain one binary table extension which describes the acquisition search pattern dwell point locations and counts as shown in Table 2.11 and Figure 2.8.

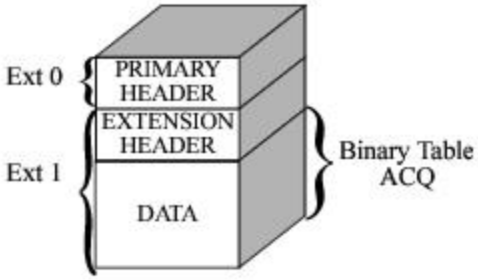
Table 2.11: Columns of an ACQ/SEARCH or ACQ/PEAKD Table.

Column Name	Units	Description
DWELL_POINT		Dwell point number in search pattern
DISP_OFFSET[1]	arcsec	Offset in dispersion direction from the initial target pointing
XDISP_OFFSET	arcsec	Offset in the cross-dispersion direction from the initial target pointing
COUNTS	counts	Raw counts value at dwell point

[1] DISP_OFFSET is only present in tables for ACQ/SEARCH and PEAKD.

[2] DISP_OFFSET is not present in tables for ACQ/SEARCH and PEAKD.

Figure 2.8: FITS File Format for ACQ/SEARCH and ACQ/PEAKD Data.



PEAKXD Acquisition

Acquisition peakups in the cross-dispersion direction (`ACQ/PEAKXD`) use a `TIME-TAG` spectrum to center the target in the cross-dispersion direction. For more information on the `ACQ/PEAKXD` algorithm see Section 8.5 of the *COS Instrument Handbook*. An `ACQ/PEAKXD` exposure includes only a primary header and extension header. Note that in almost all cases a PEAKXD is done first, then a PEAKD.

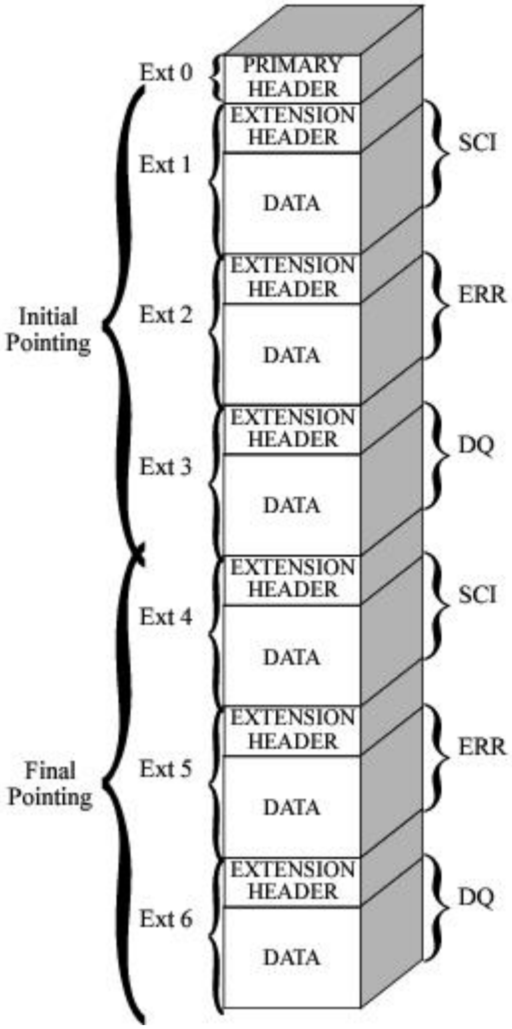
IMAGE Acquisition

Acquisition images (`ACQ/IMAGE`) use an NUV image to center the target in the aperture. For more information on the `ACQ/IMAGE` algorithm see Section 8.4 of the *COS Instrument Handbook*. An `ACQ/IMAGE` exposure produces a raw data file containing two science image extensions corresponding to the initial and final pointing:

- [SCI,1] is an image of the initial target pointing.
- [SCI,2] is a confirmation image after the acquisition procedure has been performed.

See Figure 2.9 for the FITS file format for `ACQ/IMAGE` data.

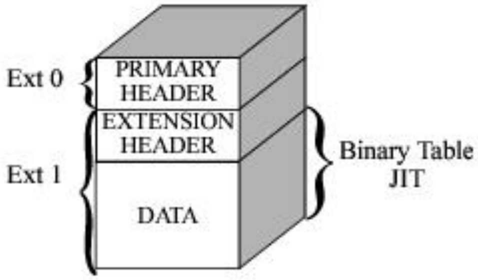
Figure 2.9: FITS File Format for `ACQ/IMAGE` Data.



Jitter Files (`jit`)

The COS jitter files include engineering data that describe 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 during the duration of the observation. The support files contain information about the observation and engineering data from the instrument and spacecraft that were recorded at the time of the observation. COS jitter files utilize the file format shown in Figure 2.10 for all science observations, excluding acquisitions.

Figure 2.10: FITS File Format for `JITTER` Data.



The jitter tables contain PCS data for each three-second interval during the observation, as listed in Table 2.12. For more information on jitter files refer to Chapter 5 of the *Introduction to HST Data Handbooks*.

Table 2.12: Columns of a jitter Table.

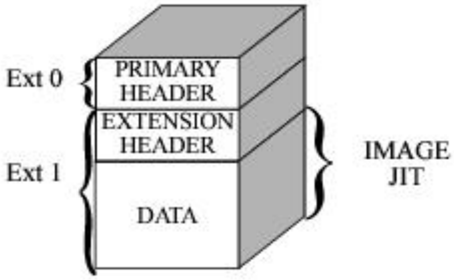
Column Name	Data Type	Units	Description
`SECONDS`	float	seconds	'Seconds' three-second intervals from start
`V2_DOM`	float	arcsec	Dominant FGS V2 Coordinate
`V3_DOM`	float	arcsec	Dominant FGS V3 Coordinate
`V2_ROLL`	float	arcsec	Roll FGS V2 Coordinate
`V3_ROLL`	float	arcsec	Roll FGS V3 Coordinate
`SI_V2_AVG`	float	arcsec	Mean jitter in V2 over 3 seconds
`SI_V2_RMS`	float	arcsec	Peak jitter in V2 over 3 seconds
`SI_V2_P2P`	float	arcsec	RMS jitter in V2 over 3 seconds
`SI_V3_AVG`	float	arcsec	Mean jitter in V3 over 3 seconds
`SI_V3_RMS`	float	arcsec	Peak jitter in V3 over 3 seconds
`SI_V3_P2P`	float	arcsec	RMS jitter in V3 over 3 seconds
`RA`	double	degrees	Right Ascension of aperture reference
`DEC`	double	degrees	Declination of aperture reference
`ROLL`	doublet	degree	Position angle between North and +V3
`LIMBANG`	float	degree	Position angle between V1 axis and Earth limb
`TERMANG`	float	degree	Angle between V1 axis and terminator
`LOS_ZENITH`	float	degree	Angle between *HST* Zenith and target

`LATITUDE`	float	degree	*HST* subpoint latitude
`LONGITDUE`	float	degree	*HST* subpoint longitude
`MAG_V1`	float	Gauss	Magnetic field along V1
`MAG_V2`	float	Gauss	Magnetic field along V2
`MAG_V3`	float	Gauss	Magnetic field along V3
`BRIGHTLIMB`	integer		Earth limb of LimbAng is bright (1 or 0) t
`FGS_FLAGS`	float		FGS status flags
`DAYNIGHT`	string		Observation taken during the day (0) or night (1)
`RECENTER`	string		Recentering status flag, event in progress = 1
`TAKEDATA`	string		Vehicle guiding status, nominal GS tracking = 1
`SLEWFLAG`	string		Vehicle slewing status, slewing = 1

2-D Spacecraft Pointing Histogram (`jif`)

The COS jif files are a 2-D histogram of the corresponding `jit` file (see Jitter Files) and have the file format shown in Figure 2.11 for all science observations excluding acquisitions.

Figure 2.11: FITS File Format for `jif` Data.



2.5 Data Storage Requirements

Users are reminded to consider the large size of `counts` and `flt` files when allocating disk space for storing and reprocessing COS data. Additionally, `corrtag` files with a large number of events can be quite large. These images serve as intermediate or final calibration products from the pipeline and have the file sizes given in Megabytes in Table 2.13. *Note that these sizes are per exposure, and an associated observation set may have several exposures.*

Table 2.13: COS Pipeline Data Volumes per Exposure.

File Type	FUVA	FUVB	Total FUV	Total NUV	Calibrated File
`rawtag`	9 bytes per photon	9 bytes per photon	9 bytes per photon (20 MB per buffer dump)	8 bytes per photon (18 MB per buffer dump)	
`corrtag`	39 bytes per photon	39 bytes per photon	39 bytes per photon (87 MB per buffer dump)	26 bytes per photon (58 MB per buffer dump)	•
`rawaccum`	64 MB	64 MB	128 MB	2 MB	
`flt`	160 MB	160 MB	320 MB	10 MB	•
`x1d`	0.5 MB[1]	0.5 MB[1]	1 MB[2]	<1 MB	•
`fltsum`	N/A	N/A	N/A	10 MB	•
`counts`	160 MB	160 MB	320 MB	10 MB	•
`x1dsum`	0.5 MB	0.5 MB	1 MB[2]	<1 MB	•
`lampflash`	N/A	N/A	<1 MB	<1 MB	•

[1] Values pertain to `x1d_a` or `x1d_b` files only. These files are temporary output products from **calcos** processing.

[2] Values are in addition to amounts given for each segment.

Similarly, users are reminded of the large cumulative size of calibrated COS spectroscopic datasets. Table 2.14 provides volume estimates for calibrated COS datasets.

Table 2.14: COS Pipeline Data Volumes per Calibrated Exposure.

Detector	FUV		NUV	
Observation Mode	`TIME-TAG`	`ACCUM`	`TIME_TAG`	`ACCUM`
Pipeline-processed volume per exposure	650 MB + 48 bytes per photon	775 MB + 39 bytes per photon	25–35 MB + 34 bytes per photon	25-35 bytes

Standard calibrated files[1]	325 MB + 39 bytes per photon[2]	325 MB	15–25 MB + 36 bytes per photon[3]	15–25 MB

[1] Minimum volume delivery option over the internet.

[2] Approximately 70 MB per `BUFFER-TIME`.

[3] Approximately 52 MB per `BUFFER-TIME`.

2.6 Headers, Keywords, and Relationship to Phase II

As with previous *HST* instruments, the FITS header keywords in COS data files store important information characterizing the observations and telemetry received during the observations, and describe the post-observation processing of your dataset. Each keyword follows FITS conventions and is no longer than eight characters. Values of keywords can be integer, real (floating-point), boolean, and character strings. Several keywords are *HST* and COS specific. Knowledge of the keywords and where to find them is an important first step in understanding your data. By examining your file headers using the **astropy.io.fits** and **astropy.table** modules in the **stenv** environment, you will find detailed information about your data including:

- Target name, coordinates, proposal ID, and other proposal level information.
- Observation and exposure time information such as observation start and duration.
- Instrument configuration information such as detector, grating, central wavelength setting, and aperture.
- Readout definition parameters such as subarray parameters.
- Exposure-specific information such as more detailed timing, world coordinate system information, and Fine Guidance Sensor identification.
- Calibration information such as the calibration switches and reference files used by the pipeline and parameters derived from the calibration, such as image statistics and wavelength shifts.

The keywords relevant for one COS data type will not necessarily be relevant to another. Accordingly, you will find that the header in a particular file type contains a unique combination of keywords appropriate for that type of observation. Long definitions for the keywords can also be accessed from the following Web page, which provides detailed explanations of the contents and algorithm for populating the keywords. This site also provides sample headers for different COS file types:
http://stdatu.stsci.edu/keyword/.

Keywords that deal with a particular topic, such as the instrument configuration, are grouped together logically throughout the headers. Table 2.15 lists a useful subset of these groups of keywords, indicates the name of the grouping, and where applicable, shows their relationship to the corresponding information from the Phase II proposal.

Table 2.16 (spectroscopy) and Table 2.17 (imaging) summarize the possible calibration switch keywords and indicate whether they are present for particular observations; Table 2.18 indicates reference file keywords corresponding to particular calibration steps. A calibration switch keyword is initially populated with values of `OMIT`, `PERFORM` or `N/A` in the raw uncalibrated science data (with the exception of XTRACTALG, which has values of 'BOXCAR' or 'TWOZONE' to select the extraction algorithm to be used, depending on lifetime position (LP)). After each calibration step is executed in the COS calibration pipeline, **calcos** will set the keyword switch to `COMPLETE`.

Table 2.15: Selected Header Keywords and Relationship to Phase II Parameters.

Header Keyword	Phase II Equivalent	Description
	Gerneral File Information (Primary Header)	
`FILENAME`		Name of file
`FILETYPE`		Type of data found in the file (SCI, ACQ, APT, ASN_TABLE)
`NEXTEND`		Number of extensions in the file

`DATE`		Date file was created
Program Information (Primary Header)		
`PROPOSID`		4 or 5 digit program number

`PR_INV_L`
`PR_INV_F`
`PR_INV_M`
	PI Last Name
	PI First Name
	PI Middle Initial
		Last name of principal investigator
		First name of principal investigator
		Middle name initial of principal investigator

`LINENUM`	Visit_Number, Exposure_Number	Indicates the visit and exposure number from the Phase II proposal: Visit_Number, Exposure_Number
Target Information (Primary Header)		
`TARGNAME`	TargetName	Name of target
`RA_TARG` `DEC_TARG`	RA DEC	Right ascension of the target (deg) (J2000) Declination of the target (deg) (J2000)
`POSTARG1` `POSTARG2`	POSTARG POSTARG	Postarg in axis 1 direction Postarg in axis 2 direction
Science Instrument Configuration (Primary Header)		
`OBSTYPE`		Observation type (`IMAGING` or `SPECTROSCOPIC`)
`OBSMODE`	Opmode	Operating mode (`ACCUM`, `TIME-TAG`)
`EXPTYPE`	Opmode	Exposure type (`EXTERNAL/SCI`, `WAVECAL`, `PHA`, `DARK`, `FLAT`, `ACQ/IMAGE`, `ACQ/SEARCH`, `ACQ/PEAKD`, `ACQ/PEAKXD`, `ENG DIAG`, or `MEM DUMP`)
`DETECTOR`	Config	Detector in use (`NUV` or `FUV`)
`SEGMENT`	SEGMENT	FUV detector segment in use (`FUVA`, `FUVB`, `BOTH`, or `N/A`)
`DETECTHV`		FUV detector high voltage state (`NomAB`, `NomA`, `NomB`, `Off`, `Low`)
`SUBARRAY`		Data from a subarray (`T`) or full frame (`F`)
`LAMPUSED`		Lamp status, `NONE` or name of lamp which is on (`P1`, `D1`, `P2`, or `D2`)
`LAMPSET`		Spectral calibration lamp current value (milliamps)
`LIFE_ADJ`	LIFETIME-POS	Detector life time adjustment position
`OPT_ELEM`	SpElement	Optical element in use (grating or mirror name)
`CENWAVE`	Wavelength	Central wavelength for grating settings
`APERTURE`	Aperture	Aperture name
`PROPAPER`	Aperture	Proposed aperture name
`APER_FOV`		Aperture field of view description in mm

APMPOS		Aperture Mechanism Position from telemetry
APERXPOS		Aperture Block Pos. cross dispersion (telemetry)
FPPOS	FPPOS	Grating offset index (1–4) for spectrum dithers (FPPOS)
TAGFLASH	FLASH	Type of flashed exposures in TIME-TAG (NONE, AUTO, or UNIFORMLY-SPACED)
EXTENDED	Extended	Is the target extended (Yes or No)
NRPTEXP	NumberOfIterations	Number of repeat exposures in dataset: Default = 1
EXP_NUM		Exposure number for repeated observations
SHUTTER		External shutter position (OPEN or CLOSED)
Engineering Parameters (Primary Header)		
FPOFFSET		FP offset from nominal, in motor steps
DEVENTA		Digital event counter, FUV segment A (counts s^{-1})
DEVENTB		Digital event counter, FUV segment B (counts s^{-1})
FEVENTA		Fast event counter, FUV segment A (counts s^{-1})
FEVENTB		Fast event counter, FUV segment B (counts s^{-1})
MEVENTS		NUV MAMA event counter (counts s^{-1})
Target Acquisition Dataset Identifiers (Primary Header)		
ACQSNAME		Rootname of first acquisition search exposure
ACQINAME		Rootname of first acquisition image exposure
PEAKXNAM		Rootname of first x-dispersion peakup exposure
PEAKDNAM		Rootname of first dispersion peakup exposure
ACQ_NUM		Total number of exposures in acquisition sequence
Archive Search Keywords (Primary Header)		
BANDWID		Bandwidth of the data
SPECRES		Approximate resolving power at central wavelength
CENTRWV		Central wavelength of the data
MINWAVE		Minimum wavelength in spectrum
MAXWAVE		Maximum wavelength in spectrum

PLATESC		Plate scale (arcsec/pixel)
Association Keywords (Primary Header)		
ASN_ID		Unique identifier assigned to association
ASN_MTYP		Role of the member in the association
ASN_TAB		Name of the association table
Exposure Information (in Extension header 1 or greater)		
DATE-OBS		UT date of start of observation (yyyy-mm-dd)
TIME-OBS		UT time of start of observation (hh:mm:ss)
EXPTIME		Corrected exposure time (seconds). For FUV exposures, the largest of EXPTIMEA and EXPTIMEB
EXPTIMEA		Corrected FUV Segment A exposure time (seconds)
EXPTIMEB		Corrected FUV Segment B exposure time (seconds)
RAWTIME		Exposure time of an individual raw exposure (seconds)
EXPSTART		Exposure start time (Modified Julian Date)
EXPEND		Exposure end time (Modified Julian Date)
EXPSTRTJ		Exposure start time (Julian Date)
EXPENDJ		Exposure end time (Julian Date)
PLANTIME	TimePerExposure	Planned exposure time (seconds)
NINTERPT		Number of exposure interrupts
ORIENTAT		Position angle of image y-axis (degrees)
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)
GYROMODE		Number of gyros scheduled for observation
REFFRAME		Guide star catalog version
Aperture Information (in Extension header 1 or greater)		
RA_APER DEC_APER		RA of reference aperture center Declination of reference aperture center
PA_APER		Position angle of reference aperture center

SHIFT1A		Wavecal shift determined for FUV segment A or NUV stripe A in dispersion direction (pixels)
SHIFT1B		Wavecal shift determined for FUV segment B or NUV stripe B in dispersion direction (pixels)
SHIFT1C		Wavecal shift determined for NUV stripe C in dispersion direction (pixels)
SHIFT2A		Offset in cross-dispersion direction for FUV segment A or NUV stripe A (pixels)
SHIFT2B		Offset in cross-dispersion direction for FUV segment B or NUV stripe B (pixels)
SHIFT2C		Offset in cross-dispersion direction for NUV stripe C (pixels)
SP_LOC_A		Cross-dispersion location at which **calcos** extracted the FUV segment A or NUV stripe A spectrum
SP_LOC_B		Cross-dispersion location at which **calcos** extracted the FUV segment B or NUV stripe B spectrum
SP_LOC_C		Cross-dispersion location at which **calcos** extracted the NUV stripe C spectrum
SP_NOM_A		Nominal location of the spectral extraction region for FUV segment A or NUV stripe A based on the wavecal aperture location
SP_NOM_B		Nominal location of the spectral extraction region for FUV segment B or NUV stripe B based on the wavecal aperture location
SP_NOM_C		Nominal location of the spectral extraction region for NUV stripe C based on the wavecal aperture location
SP_OFF_A		Offset from SP_NOM_A at which the spectrum was found
SP_OFF_B		Offset from SP_NOM_B at which the spectrum was found
SP_OFF_C		Offset from SP_NOM_C at which the spectrum was found
SP_SLP_A		Slope of FUV segment A or NUV stripe A spectrum
SP_SLP_B		Slope of FUV segment B or NUV stripe B spectrum
SP_SLP_C		Slope of NUV stripe C spectrum
SP_HGT_A		Height in pixels of the FUV Segment A spectral extraction region
SP_HGT_B		Height in pixels of the FUV Segment B spectral extraction region
DPIXEL1A		Fractional part of pixel coordinate for FUV segment A or NUV stripe A (pixels). Average binning error
DPIXEL1B		Fractional part of pixel coordinate for FUV segment B or NUV stripe B (pixels). Average binning error

`DPIXEL1C`		Fractional part of pixel coordinate for NUV stripe C (pixels). Average binning error
TIME-TAG Parameters (in Extension header 1 or greater)		
`BUFFTIME`	*BUFFER-TIME*	*Onboard memory half-buffer-fill time*
`OVERFLOW`		Number of science data overflows
`NBADEVNT`		Total number of events deleted in screening (NUV)
`NBADEVTA`		Total number of events deleted in screening (FUV, segment A)
`NBADEVTB`		Total number of events deleted in screening (FUV, segment B)
`NEVENTS`		Total number of events in raw data (NUV)
`NEVENTSA`		Total number of events in raw data (FUV, segment A)
`NEVENTSB`		Total number of events in raw data (FUV, segment B)
FUV TIME-TAG Parameters (in Extension header 1 or greater)		
`TBRST_A` `TBRST_B`		Time lost to bursts on FUV segment A (seconds) Time lost to bursts on FUV segment B (seconds)
`TBADT_A` `TBADT_B`		Time lost to BADTCORR screening on FUV segment A (sec) Time lost to BADTCORR screening on FUV segment B (sec)
`NPHA_A` `NPHA_B`		Number of events lost due to pulse height screening on segment A Number of events lost due to pulse height screening on segment B
`NBRST_A` `NBRST_B`		Number of events lost due to burst screening on segment A Number of events lost due to burst screening on segment B
`NBADT_A` `NBADT_B`		Number of events flagged by BADTCORR for segment A Number of events flagged by BADTCORR for segment B
`NOUT_A` `NOUT_B`		Number of events outside the active area for segment A Number of events outside the active area for segment B
NUV TIME-TAG Parameters (in Extension header 1 or greater)		
`NBADT`		Number of events flagged by BADTCORR
`TBADT`		Time lost to BADTCORR screening (sec)
TAFGFLASH Parameters (in Extension header 1 or greater)		
`NUMFLASH`		Integer number of flashes
`LMPDURi`		Duration of flash i, seconds

`LMP_ONi`		Lamp turn-on time for flash i, seconds since EXPSTART
`LMPOFFi`		Lamp turn-off time for flash i, seconds since EXPSTART
`LMPMEDi`		Median time of flash i, seconds since EXPSTART
Velocity Reference Frame Conversion (in Extension header 1 or greater)		
`V_HELIO`		Geometric to heliocentric velocity
`V_LSRSTD`		Heliocentric to standard solar LSR
Doppler Correction Parameters (in Extension header 1 or greater)		
`ORBITPER`		Orbital period used onboard for Doppler correction
`DOPPER`		Doppler shift period (seconds)
`DOPPMAGV`		Doppler shift magnitude (Km/sec)
`DOPPON`		Doppler correction flag (T or F)
`DOPPZERO`		Commanded time of zero Doppler shift (MJD)
Instrument Doppler Correction Parameters (in Extension header 1 or greater)		
`ORBTPERT`		Orbital period used onboard for Doppler correction
`DOPMAGT`		Doppler shift magnitude (low-res pixels)
`DOPPONT`		Doppler correction flag (T or F)
`DOPPZEROT`		Commanded time of zero Doppler shift (MJD)
Global Count Parameters (in Extension header 1 or greater)		
`GLOBRATE`		Global count rate (NUV)
`GLOBRT_A`		Global count rate (FUV, segment A)
`GLOBRT_B`		Global count rate (FUV, segment B)
Subarray Readout Parameters[1] (in Extension header 1 or greater)		
`NSUBARRY`		Number of subarrays (1–8)
`CORNER[N]X`		Subarray N axis1 corner pt in unbinned detector pixels. Valid values of N are 0 to 7
`CORNER[N]Y`		Subarray N axis2 size in unbinned detector pixels, with N=0 to 7
`SIZE[N]Y`		Subarray N axis1 corner pt in unbinned detector pixels, with N=0 to 7
`SIZE[N]X`		Subarray N axis2 size in unbinned detector pixels, with N=0 to 7

Stim Pulse Parameters (in Extension header 1 or greater; for FUV data only)		
`STIMRATE`		Approximate STIM pulse injection rate

`STIMA_LX`
`STIMA_LY`
`STIMA_RX`
`STIMA_RY`
		Segment A Left STIM pulse X centroid in raw data
		Segment A Left STIM pulse Y centroid in raw data
		Segment A Right STIM pulse X centroid in raw data
		Segment A Right STIM pulse Y centroid in raw data

`STIMB_LX`
`STIMB_LY`
`STIMB_RX`
`STIMB_RY`
		Segment B Left STIM pulse X centroid in raw data
		Segment B Left STIM pulse Y centroid in raw data
		Segment B Right STIM pulse X centroid in raw data
		Segment B Right STIM pulse Y centroid in raw data

`STIMA0LX`
`STIMA0LY`
`STIMA0RX`
`STIMA0RY`
		Reference location of Segment A Left STIM pulse X coordinate
		Reference location of Segment A Left STIM pulse Y coordinate
		Reference location of Segment A Right STIM pulse X coordinate
		Reference location of Segment A Right STIM pulse Y coordinate

`STIMB0LX`
`STIMB0LY`
`STIMB0RX`
`STIMB0RY`
		Reference location of Segment B Left STIM pulse X coordinate
		Reference location of Segment B Left STIM pulse Y coordinate
		Reference location of Segment B Right STIM pulse X coordinate
		Reference location of Segment B Right STIM pulse Y coordinate

`STIMASLX`
`STIMASLY`
`STIMASRX`
`STIMASRY`
		RMS width of Segment A Left STIM pulse X coordinate
		RMS width of Segment A Left STIM pulse Y coordinate
		RMS width of Segment A Right STIM pulse X coordinate
		RMS width of Segment A Right STIM pulse Y coordinate

`STIMBSLX`
`STIMBSLY`
`STIMBSRX`
`STIMBSRY`
		RMS width of Segment B Left STIM pulse X coordinate
		RMS width of Segment B Left STIM pulse Y coordinate
		RMS width of Segment B Right STIM pulse X coordinate
		RMS width of Segment B Right STIM pulse Y coordinate

Pulse Height Parameters (in Extension header 1 or greater for FUV data only)		
`PHALOWRA` `PHALOWRB`		Pulse height screening lower limit for segment A Pulse height screening lower limit for segment B
`PHAUPPRA` `PHAUPPRB`		Pulse height screening upper limit for segment A Pulse height screening upper limit for segment B
Image Statistics and Data Quality Flags (in Extension header 1 or greater)		
`NGOODPIX`		Number of good pixels (NUV)
`NGOOD_A`		Number of good pixels (FUV, segment A)
`NGOOD_B`		Number of good pixels (FUV, segment B)

SDQFLAGS		Serious data quality flags. (Can be modified as a **calcos** parameter; see Section 3.4.22)
GOODMAX		Maximum value of good pixels (NUV)
GOODMAXA		Maximum value of good pixels (FUV, segment A)
GOODMAXB		Maximum value of good pixels (FUV, segment B)
GOODMEAN		Mean value of good pixels (NUV)
GOODMN_A		Mean value of good pixels (FUV, segment A)
GOODMN_B		Mean value of good pixels (FUV, segment B)
SOFTERRS		Number of soft error pixels (DQF=1)
Deadtime Correction Keywords (in Extension header 1 or greater)		
DEADRT DEADRT_A DEADRT_B		Count rate used for the NUV dead time correction (cps) Count rate used in the FUV Segment A dead time correction (cps) Count rate used in the FUV Segment B dead time correction (cps)
DEADMT DEADMT_A DEADMT_B		NUV Deadtime correction method (DATA, DEVENTS, or MEVENTS) FUVA Deadtime correction method (DATA, DEVENTS, or MEVENTS) FUVB Deadtime correction method (DATA, DEVENTS, or MEVENTS)
TIME-TAG Events Orientation Keywords (in Extension header 1 or greater)[2]		
TCTYP2 TCTYP3		Axis type for dimension 1 Axis type for dimension 2
TCRVL2 TCRVL3		Sky coordinates of 1st axis Sky coordinate of 2nd axis
TCDLT2 TCDLT3		Axis 1 degrees per pixels Axis 2 degrees per pixels
TCRPX2 TCRPX3		Axis 1 pixel of tangent plane direction Axis 2 pixel of tangent plane direction
TALEN2 TALEN3		Length of axis 1 Length of axis 2
TC2_2 TC2_3 TC3_2 TC3_3		Partial of first axis coordinate with respect to x Partial of first axis coordinate with respect to y Partial of second axis coordinate with respect to x Partial of second axis coordinate with respect to y

`TCUNI2` `TCUNI3`		Units of first coordinate value Units of second coordinate value
World Coordinate System and Related Parameters (in Extension header 1 or greater)		
`WCSAXES`		Number of World Coordinate System axes
`CRPIX1` `CRPIX2`		x-coordinate of reference pixel y-coordinate of reference pixel
`CRVAL1` `CRVAL2`		First axis value at reference pixel Second axis value at reference pixel
`CTYPE1` `CTYPE2`		The coordinate type for the first axis The coordinate type for the second axis

`CD1_1`
`CD1_2`
`CD2_1`
`CD2_2`
		Partial of first axis coordinate with respect to x
		Partial of first axis coordinate with respect to y
		Partial of second axis coordinate with respect to x
		Partial of second axis coordinate with respect to y

`CUNIT1` `CUNIT2`		Units of first coordinate value Units of second coordinate value
`LTV1` `LTV2`		Offset in X to subsection start Offset in Y to subsection start
`LTM1_1` `LTM2_2`		Reciprocal of sampling rate in X Reciprocal of sampling rate in Y
Detector Voltages (in Extension header 1 or greater)		
`HVLEVELA`		Actual segment A commanded HV level (counts)
`HVLEVELB`		Actual segment B commanded HV level (counts)

---

[1] For FUV data subarrays 0–3 refer to segment A, and subarrays 4–7 refer to segment B.

[2] The values for these keywords are currently deleted from the output files except for NUV Imaging.

Table 2.16: Spectroscopic Calibration Switch Keywords.

EXPTYPE	*EXTERNAL/SCI* *EXTERNAL/CAL*				*WAVECAL*		*DARK*		*FL*
DETECTOR	*FUV*		*NUV*		*FUV*	*NUV*	*FUV*	*NUV*	*FUV*
OBSMODE	*TIME-TAG*	*ACCUM*	*TIME-TAG*	*ACCUM*	*TIME-TAG*	*TIME-TAG*	*TIME-TAG*	*TIME-TAG*	*TIME-TAG*

Module									
BRSTCORR	Omit	N/A	N/A	N/A	Omit[1]	N/A	Omit[1]	N/A	Omit[2]
BADTCORR	Perform	N/A	Perform	N/A	Perform	Perform	Perform	Perform	Perform
PHACORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
RANDCORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
RANDSEED	−1	−1	N/A	N/A	−1	N/A	−1	N/A	−1
XWLKCORR[2]	Omit	N/A	N/A	N/A	Perform	N/A	Perform	N/A	Perform
YWLKCORR	Perform	N/A	N/A	N/A	Perform	Perform	Perform	N/A	Perform
TEMPCORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
GEOCORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
DGEOCORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
IGEOCORR	Perform	Perform	N/A	N/A	Perform	N/A	Perform	N/A	Perform
DOPPCORR	Perform	Perform	Perform	Perform	N/A	N/A	N/A	N/A	N/A
DEADCORR	Perform	Perform	Perform	Perform	Perform	Perform	Perform	Perform	Perform
FLATCORR	Perform	Perform	Perform	Perform	Perform	Perform	Omit	Omit	Omit
XTRACTALG[1]	TWOZONE	TWOZONE	BOXCAR	BOXCAR	BOXCAR	BOXCAR	BOXCAR	BOXCAR	BOXCAR
ALGNCORR[1]	Perform	Perform	Omit	Omit	Omit	Omit	Omit	Omit	Omit
DQICORR	Perform	Perform	Perform	Perform	Perform	Perform	Perform	Perform	Perform
WAVECORR	Perform	Perform	Perform	Perform	Perform	Perform	Omit	Omit	Omit
X1DCORR	Perform	Perform	Perform	Perform	Perform	Perform	Omit	Omit	Omit
BACKCORR	Perform	Perform	Perform	Perform	Perform	Perform	Omit	Omit	Omit
FLUXCORR	Perform	Perform	Perform	Perform	N/A	N/A	Omit	Omit	Omit
TDSCORR	Perform	Perform	Perform	Perform	N/A	N/A	Omit	Omit	Omit
HELCORR	Perform	Perform	Perform	Perform	N/A	N/A	Omit	Omit	Omit
TRCECORR[1]	Perform	Perform	Omit	Omit	Omit	Omit	Omit	Omit	Omit
STATFLAG	T	T	T	T	T	T	T	T	T

[1] FUV data taken at LP1 and LP2 will still use BOXCAR extraction. ALGNCORR and TRCECORR will be set to OMIT.

[2] XWLKCORR is OMIT for LP=4 and PERFORM for all other LPs. DGEOCORR is PERFORM for LP=4 and OMIT for all other LPs.

Table 2.17: Imaging Calibration Switch Keywords.

EXPTYPE	***EXTERNAL/SCI*** ***EXTERNAL/CAL***		***WAVECAL***	***DARK***	***FLAT***	***ACQ/IMAGE***
DETECTOR	***NUV***		***NUV***	***NUV***	***NUV***	***NUV***
OBSMODE	***TIME-TAG***	***ACCUM***	***TIME-TAG***	***TIME-TAG***	***TIME-TAG***	***ACCUM***
Modules						
BADTCORR	Perform	N/A	Perform	Perform	Perform	N/A
FLATCORR	Perform	Perform	Perform	Omit	Omit	Perform
DEADCORR	Perform	Perform	Perform	Perform	Perform	Perform
DQICORR	Perform	Perform	Perform	Perform	Perform	Perform
PHOTCORR	Perform	Perform	Perform	Omit	Omit	Perform
TDSCORR	Perform	Perform	Perform	Omit	Omit	Perform
STATFLAG	T	T	T	T	T	T

Table 2.18: Reference File Keywords.

Note that some reference files depend on LP (lifetime position), cenwave, etc. See https://hst-crds.stsci.edu/.

Note also that reference file names in italic bold (e.g., ***BRSTTAB***) apply only to FUV data.

Reference File	Description
BRSTTAB	Burst parameter table
BRFTAB	Baseline reference frame reference table
BADTTAB	Bad time interval reference table
PHATAB	Pulse height discrimination reference table
GEOFILE	Geometric distortion table
DGEOFILE	Delta Geometric Correction Reference File
YWLKFILE	Y Walk Correction Lookup Reference Image
XWLKFILE	X Walk Correction Lookup Reference Image
DEADTAB	Deadtime reference file
FLATFILE	Pixel to pixel flat-field reference file

LAMPTAB	Template calibration lamp spectra table
WCPTAB	Wavecal parameters table
DISPTAB	Dispersion coefficient reference table
BPIXTAB	Bad pixel table
GSAGTAB	Gain sag table
XTRACTAB	1-D spectral extraction information table
FLUXTAB	Photometric throughput table
TDSTAB	Time-dependent sensitivity correction table
SPWCSTAB	Spectroscopic World Coordinate System table
TRACETAB	1-D spectral trace table
PROFTAB	2-D spectrum profile table
HVTAB	High voltage command level reference table
SPOTTAB	Hot spot reference table
TWOZXTAB	Two-zone spectral extraction information table

2.7 Error and Data Quality Arrays

The **calcos** pipeline propagates data quality flags throughout the calibration process. The **flux error** estimates are computed by propagating the individual uncertainty components related to the flat-field, gross count rate, and smoothed background count rate in each wavelength bin. **Note that starting with calcos 3.3.10 the Poisson error calculations are achieved using the asymmetric "Frequentist-Confidence" method in astropy rather than the Gehrels (1986) upper confidence limit equation. See COS ISR 2021-03 for more information.**

2.7.1 Upper and Lower Flux Error Arrays

The ERROR and ERROR_LOWER arrays contain estimates of the 1-sigma upper and lower flux uncertainties, respectively, in each wavelength bin. For high S/N observations the ERROR and ERROR_LOWER uncertainty values will be symmetric; however, the flux errors become increasingly asymmetric when count levels fall below ~30 counts. Information about the three general terms that contribute to the flux errors are provided in each wavelength bin via the VARIANCE_FLAT, VARIANCE_COUNTS, and VARIANCE_BKG columns. For X1D and X1DSUM<N> files comprised of single exposures, the general equation for calculating net count rate uncertainties is:

$$\sigma_{u,l;N_i} = \frac{1}{t}\sqrt{\left(\frac{N_i t}{E_j SNR_{ff}}\right)^2 + \left[\varepsilon_i f_{u,l}(GC_i t)\right]^2 + \left[-\varepsilon_i f_{u,l}\left(\frac{\Delta n_1}{\Delta w \Delta n_2} BK_i t\right)\right]^2}$$

where the first term under the radical is VARIANCE_FLAT and the f_{u,l} function is the "Frequentist-Confidence" method in CalCOS but can be substituted for a function of the users choice. For example, users wishing to use symmetric root-N errors could calculate the net count rate uncertainties as:

$$\sigma_{u,l;N_i} = \frac{1}{t}\sqrt{VARIANCE\_FLAT_i + VARIANCE\_COUNTS_i + VARIANCE\_BKG_i}$$

where the net count rate uncertainty can be converted into a flux uncertainty using the relation:

$$\sigma_{F_{u,l;i}} = \frac{F_i}{N_i} \cdot \sigma_{u,l;N_i}$$

In contrast, X1DSUM and X1DSUM[N] files comprised of two or more exposures interpolate onto a common wavelength grid, and as a result are affected by covariance between adjacent wavelength bins along with including measurements with potentially different data quality flags. The upper and lower flux errors in X1DSUM and X1DSUM[N] files are thus approximations based on treating all contributions to a given wavelength bin as "effective counts." The summed contributions for the three general error terms are provided in the same VARIANCE_FLAT, VARIANCE_COUNTS, and VARIANCE_BKG columns as in the X1D files. Further information is provided in Section 4.2 of COS ISR 2021-03.

2.7.2 Data Quality Flags

Every photon event in a COS `corrtag` file has a Data Quality (DQ) flag (Table 2.19). Each flagged condition sets a specific bit in the 16-bit DQ word, thus allowing each event during an exposure to be flagged with multiple conditions using the bitwise logical OR operation. DQ flags can be divided into five types:

1. Spatial flags mark events which fall on a detector region which may be questionable. The BPIXTAB reference file marks the corners of each region on the detector which falls into each of these categories. Separate BPIXTAB files are used for the FUV and NUV detectors. These regions were determined by visual inspection of a set of science data files. For FUV data, the GSAGTAB is applied along with the BPIXTAB and SPOTTAB. The GSAGTAB is used to flag regions that are severely gain sagged (with a median PHA of less than 3). The SPOTTAB is the hot spot reference table.

The DQICORR step of **calcos** maps these spatial regions to the individual photon events, and the **x1dcorr** module uses these flags and the value of SDQFLAGS to create the DQ and DQ_WGT arrays, and ultimately to determine which events to include in the final (`x1dsum`) spectrum (Section 3.4.18).

The spatial flags include:

- Detector shadows (4) include the locations of the grid wires for the FUV detector, and the vignetted region on the NUV detector.
- Poorly calibrated regions (8) include areas near the edge of the detector which may be suspect.
- Very low response regions (16) are areas on the detector where the response presents a >80% depression.
- Background features (32) correspond to regions on the detector where the background count rate has been observed to be higher than the surrounding region and/or unstable.
- The pixel out-of-bounds flag (128) marks regions outside of the calibrated region of the detector.
- Low response regions (1024) are areas on the detector where the response presents a >50% depression.
- Low PHA features (4096) are regions in which unusual features have been identified in long background exposures. These features may have an effect on very low count rate observations.
- Gain-Sag holes (8192) are regions on the FUV detector where the gain is low enough that the calibration may be affected (see Section 3.7.15 describing the GSAGTAB reference file). Gain-Sag holes differ from low-response regions in being time-dependent and so are updated in the GSAGTAB instead of BPIXTAB.

2. Temporal flags mark photons that occur during time spans in which the data quality is suspect. Events flagged in this way will be removed from the data products, and the exposure time will be adjusted accordingly. Two types of temporal flags are used:

- FUV event bursts (64), which are flagged by the BRSTCORR module of **calcos**. As of this writing, no bursts have been seen on orbit, so the BRSTCORR step is set to OMIT by default. If bursts are seen at some point, it is likely that the parameters in the BRSTTAB reference table will have to be adjusted before using BRSTCORR.
- Other Bad Time Intervals (2048) can be defined in the BADTTAB reference file, for time ranges that are known to be problematic. At present, STScI has not defined any bad time intervals, but users running **calcos** on their own may wish to define their own intervals in order to exclude times with high background, etc.

3. Spatial and Temporal flags mark events that fall on a specific part of the detector, but also during specific time spans in which the data quality is suspect. Currently only the hotspot flag falls into this category.

- Hotspot flag (2) only applies to FUV data. If a hotspot overlaps any of the good time intervals, the region is added to the set of regions that are applied to create the DQ mask and against which each event is tested to assign a DQ value. The hot-spot regions are flagged in the two-zone extraction module, even if they fall only in the outer zone, and they do not contribute to the summed spectra in the x1dsum file.

4. Event flags are set by **calcos** if a photon event falls outside defined thresholds. Currently, only the FUV Pulse Height flag (512) falls into this category. All FUV events with pulse heights falling outside the range specified in the PHATAB reference file will have this flag set, and the data will be excluded by the DQICORR module. This flag does not apply to NUV data. The default value of SDQFLAGS does not include 512, but pulse height thresholding is still conducted by default.

5. Lost Data flags occur if data are missing for some reason, such as errors in transmitting the data from the instrument to the ground. Data marked with these flags is always excluded from the final products. There are two flags in this category:

- Reed-Solomon errors (1)
- Fill Data (256)

Screening for temporal and event flags is done by turning calibration switches on or off, or by altering reference files. Once a photon has been determined to have a bad temporal or event flag, it will never appear in a final data product (i.e., x1dsum or x1dsumN) unless the modules which screen for it are turned off or the reference files which define them are changed. Events with a spatial DQ flag are included in the calibrated product, and flagged in the final DQ array. The screening for the spatial flags can be easily altered by changing the SDQFLAGS keyword in the header of the raw data file. The default value for SDQFLAGS for the FUV is 8346, and it is 152 for the NUV.

The DQ extension of raw `ACCUM` files will be filled only when there are missing (data lost) or dubious (software error) data. If no such errors exist, initialization will produce an empty data quality extension whose header has `NAXIS=0`. These flags are set and used during the course of calibration, and may likewise be interpreted and used by downstream analysis applications. See Section 3.4.16 for more information on the data quality initialization calibration module.

Table 2.19: COS Data Quality Flags

FLAG Value	Bit Setting	Quality Condition	Type	FUV /NUV
	0000 0000 0000 0000	No anomalies	N/A	Both
1	0000 0000 0000 000**1**	Reed-Solomon error	Lost data	Both
2	0000 0000 0000 00**1**0	Hot Spot	Spatial and Temporal	FUV
4	0000 0000 0000 0**1**00	Detector shadow	Spatial	Both
8	0000 0000 0000 **1**000	Poorly calibrated region (including detector edge)	Spatial	Both
16	0000 0000 000**1** 0000	Very low response region (>80% depression)	Spatial	Both

32	0000 0000 00**1**0 0000	Background feature	Spatial	FUV
64	0000 0000 0**1**00 0000	Burst	Temporal	FUV
128	0000 0000 **1**000 0000	Pixel out-of-bounds	Spatial	Both
256	0000 000**1** 0000 0000	Fill data	Lost data	Both
512	0000 00**1**0 0000 0000	Pulse Height out of bounds	Event	FUV
1024	0000 0**1**00 0000 0000	Low response region (>50% depression)	Spatial	Both
2048	0000 **1**000 0000 0000	Bad time interval	Temporal	Both
4096	000**1** 0000 0000 0000	Low PHA feature	Spatial	Both
8192	00**1**0 0000 0000 0000	Gain-Sag Hole	Spatial	FUV
16384	0**1**00 0000 0000 0000	FUV detector edge high dark rates	Spatial	FUV

Note 1: Flag values in bold italics (e.g., ***128***) are used in SDQFLAGS.

Note 2: Additional information on detector edge dark rates may be found in: COS ISR 2019-11.

2.7.3 Explanation of DQ flags

The DQ flags that are listed in Table 2.19 are the flags that represent a particular data quality feature to be aware of. Each DQ flag has an assigned bit, so they each have unique values that can be combined and then disentangled. Values in the **_x1d** DQ array that aren't listed in the table represent multiple DQ flags at that pixel. To understand which DQ flags are combined into the final DQ, bitwise math needs to be performed, or checking the numbers in binary. For example, a DQ value of 1040 in binary is 0000 0100 0001 0000. Using Table 2.19, we can see that the 1s align with DQ Flag 16 (0000 0000 0001 0000), a very low response region, and DQ Flag 1024 (0000 0100 0000 0000), a low response region. To confirm, 1024 + 16 = 1040, so those are indeed the flags that went into the final DQ flag at that pixel.

Alternatively, to isolate a DQ value such as the gain-sag holes (DQ flag = 8192), one can look for all of the values in the DQ array that contain 8192 using bitwise math. The syntax for this in python is: **dq_array&8192 == 8192**. This syntax will be slightly different depending on the programming language you use, but should be generally similar. If you are a python user, there is also a function in **numpy** called **bitwise_and**, which does the same thing.

The DQ_WGT column takes into account only those DQ flags that are contained within the SDQFLAGS value. This is a header keyword for which the value is set in the first science extension of the file headers. Pixels that contain any of the DQ values within the SDQFLAGS value will have a DQ_WGT of 0. Everywhere else, the DQ_WGT will be 1 in an **_x1d** file. As a note, the DQ_WGT can be greater than 1 in the x1dsum files, as it combines the DQ_WGT arrays from the individual files.

For the FUV, the following “serious” DQs (SDQs) are by default flagged for removal from data using SQDFLAGS=8346: hot spots (2), poorly calibrated regions (8), very low response regions (16), pixel-out-of-bounds (128), and gain-sag-holes (8192). For the NUV, the following SDQs are by default flagged for removal from data using SDQFLAGS=152: poorly calibrated regions (8), very low response regions (16), and pixel out-of-bounds (128). The SDQFLAGs value is different in the NUV and the FUV simply due to differences in the detector.

We recommend that users not remove any of the aforementioned SDQs from the SDQFLAGS value. Doing so will result in these bad regions not being removed from data, which significantly affects data quality. To flag additional DQs for removal beyond the default SDQs, users should add the DQ value in question to the current SDQFLAGS value. For example, to also remove all background features (DQ = 32) from FUV data, one would add 32 + 8346, and update the headers in the **_rawtag** or **_rawaccum** file to SDQFLAGS = 8378. The file should then be calibrated through **calcos** again. The results should be that the DQ_WGT will now equal 0 wherever there is a DQ 32 flag (a bitwise and of 32 mentioned above) in addition to wherever it was 0 before.

Chapter 3: COS Calibration

Chapter Contents

3.1 Raw Data Compilation

The basic inputs to **calcos** are raw science data files. This section provides a brief overview of how these files are generated from raw spacecraft telemetry.

Telemetry containing COS science data is downlinked from the *HST* satellite through a Tracking and Delay Relay Satellite System (TDRSS) satellite to a ground station in White Sands, NM. From there it is sent to Goddard Space Flight Center where the data capture facility packet processor (PACOR) collects the downlinked science data and places them into telemetry "pod files." These pod files are then transmitted to STScI where they are saved to a permanent storage medium. The STScI ingest pipeline, **DP** (previously known as **OPUS**) then unpacks the data, populates keyword values extracted from the telemetry stream, reformats the data, and repackages them into raw and uncalibrated, but scientifically interpretable, data files.

The raw files are then processed by the **calcos** software to produce a variety of calibrated data files. The results of these procedures are used to populate the databases that form the searchable archive catalog at STScI describing the individual instrument exposures.

3.2 Pipeline Processing Overview

The calibration pipeline, **calcos**, has been developed by STScI to support the calibration of *HST*/COS data. Although the COS pipeline benefits from the design heritage of previous *HST* instruments and of the *Far Ultraviolet Spectroscopic Explorer* (*FUSE*), the **calcos** modules are tailored specifically to the COS instrument and based on data reduction algorithms defined by the COS Investigation Definition Team (IDT) and the COS team at STScI. As with other *HST* pipelines, **calcos** uses an association table (the `_asn` files) to specify the data files to be included when calibrating, and employs header keywords to specify the calibration steps to be performed and the reference files to be used. **Calcos** is written in **Python**, an open-source, easy-to-read scripting language, with many libraries for data reduction and analysis. **Calcos** can be found in the **stenv** python distribution, which is available for download from STScI.

Calcos is designed with a common underlying structure for processing FUV and NUV channels which, respectively, use a cross delay line (XDL) and a Multi Anode Microchannel Array (MAMA) detector. The **calcos** calibration pipeline includes pulse-height filtering and geometric correction for the FUV channel, and flat-field, deadtime, and Doppler correction for both channels. It includes methods for obtaining an accurate wavelength calibration by using the onboard spectral line lamps. A background subtracted spectrum is produced and the instrument sensitivity is applied to create the final flux calibrated spectrum.

There are two basic types of raw data files: `TIME-TAG` photon lists and `ACCUM` images of the detector. **Calcos** must convert these into one dimensional calibrated flux and wavelength arrays, and must be able to perform different types of calibration processes to accommodate the different input types.

The level of calibration performed depends upon the data type.

- Acquisition-mode exposures (`ACQ/SEARCH`, `ACQ/PEAKXD`, and `ACQ/PEAKD`) are not calibrated by **calcos**, with the exception of `ACQ/IMAGE`. Only the raw data from the uncalibrated modes are provided.
- All other science data, including NUV imaging data (`ACQ/IMAGE`), are completely calibrated. This includes pulse height filtering, geometric and thermal correction for the FUV data, flat fielding, and linearity corrections. The spectroscopic data are also flux calibrated and corrected for time dependence in the instrumental sensitivity. The data flow and calibration modules for processing the data are described in detail in sections 3.3 and 3.4.

The treatment of `TIME-TAG` and `ACCUM` mode data differs:

- Raw data taken in `TIME-TAG` mode are event lists (`rawtag` binary tables). The basic calibration is done on the tabular data, producing a calibrated (`corrtag`) events table. The events are then accumulated into a calibrated image (`flt`) by **calcos**.
- Raw data taken in `ACCUM` mode (`_rawaccum`) are binned into an image array onboard the spacecraft.

For spectral data, **calcos** extracts a spectrum from the flat-fielded image, computes associated wavelengths, and converts the count rates to flux densities, yielding a one-dimensional, background subtracted spectrum. For FUV data there will normally be two spectra, one from segment A and one from segment B. The two FUV segments are processed independently. For NUV data there will normally be three spectra, one for each spectral "stripe." When multiple exposures with the same setting (grating and central wavelength) but different FP-POS are contained within a single visit, these are combined into a single, summed spectrum.

See Chapter 2 for the naming conventions of the various input, temporary, and output calibrated files.

3.2.1 Overview of TWOZONE extraction

With the move to Lifetime Position 3 in February 2015, it became increasingly difficult to find science and background regions on the FUV detector that are free from overlap with the gain sagged regions from Lifetime Position 1. To allow reliable spectral extraction close to these gain sagged regions, a new method of spectral extraction was developed and implemented in **calcos** starting with version 3.0. Under the older "BOXCAR" algorithm, a rather large extraction region is used to ensure that all of the flux is collected, even for slightly miscentered targets. If any pixel in the BOXCAR extraction region is identified as bad (i.e., has a data quality flag within the Bad Pixel File (BPIXTAB) that matches those included in SDQFLAGS), the entire wavelength bin is rejected as bad and excluded from the summed files.

The newer "TWOZONE" algorithm is based on the assumption that bad pixels and gain-sagged regions that are in the outer wings of the point-source profile do not have a large enough impact on the extracted flux to force rejection of the wavelength bin; instead wavelength bins should only be rejected if a bad pixel occurs in the core of the profile. This allows for spectrum extraction with only a small error even when the far wings of the profile may overlap with gain-sagged regions near LP1. Note that the locations of LPs 3 through 6 were carefully chosen so that previous gain-sagged regions would not significantly impact the spectral quality and flux accuracy of the science spectra.

To implement this concept, the TWOZONE method divides the spectral extraction region into two parts: an INNER zone that defines the core of the profile, and an OUTER zone that includes the entire region used for the spectral extraction (note that the OUTER zone as defined here includes the INNER zone). The upper and lower boundaries for each of these zones are wavelength dependent and are defined in terms of the fraction of enclosed energy expected for the cross-dispersion profile of a point source. These enclosed energy fractions are set for each CENWAVE setting in the new TWOZXTAB reference file. For all settings, the reference files are currently set by default to define the central 80% of the profile's enclosed energy as the INNER zone and 99% as the OUTER zone, but these boundaries can be adjusted to tailor the extraction (see 3.6 Customizing COS Data Calibration). The wavelength dependent point-source spatial profiles for each setting are contained in the PROFTAB reference file.

This approach has a number of additional consequences. In order to tabulate reference profiles that are sufficiently smooth as a function of wavelength, it proved necessary to first straighten the spectral image to correct the small-scale distortions in the cross-dispersion direction. This resulted in the addition of a new TRCECORR step, which uses corrections tabulated in the TRACETAB reference file. In addition, precise alignment of the observed spectrum with the reference profile is needed to ensure accurate flux extraction, and to do this the new ALGNCORR step was added. **We recommend that the TRCECORR steps always be used whenever using the new TWOZONE algorithm, and be omitted when using the older BOXCAR algorithm. While it is possible to turn these steps on and off separately, the reliability of the extracted spectra produced may be adversely affected.**

⚠ ***Note: The TWOZONE algorithm is only used for FUV data taken at the third and subsequent COS FUV Lifetime Positions (LP3 and up).***

The TWOZONE algorithm is only used for FUV data taken at the third and subsequent COS FUV Lifetime Positions (LP3 and up). All NUV data and FUV data taken at LP1 and LP2 continue to be calibrated using the older BOXCAR algorithm. Note that when most FUV settings were moved to LP3 and later subsequent LP, the 1055 and 1096 CENWAVE settings of the G130M grating were left at LP2 because of their large cross-dispersion widths, and they will therefore continue to be calibrated with the BOXCAR algorithm.

3.2.2 Extended sources

Since the new TWOZONE algorithm shrinks the final region used for the spectral extraction to enclose only 99% of the expected point-source profile enclosed energy, the flux accuracy for extended sources will be more easily affected than was the case for the BOXCAR algorithm, which uses a larger fixed extraction height. In addition, the more extended spatial profile for these sources may increase the overlap with the gain-sagged regions near LP1, leading to significant loss of flux. Observations of extended sources therefore likely require customized extractions to produce optimum results, and close examination of the spectral images and extractions to identify artifacts in the reduced products is recommended.

⚠ ***Note: Observations of extended sources likely require customized extractions to produce optimum results.***

3.3 Calcos--Structure and Data Flow

The **calcos** pipeline is comprised of three main components that provide calibration of the COS data. This structure incorporates modules that:

1. correct the data for instrument effects (e.g., noise, thermal drifts, geometric distortions, pixel-to-pixel variations in sensitivity),
2. generate an exposure-specific wavelength-calibrated scale, and
3. extract and produce the final (one-dimensional) flux-calibrated (summed) spectrum for the entire observation.

Both COS FUV and NUV TIME-TAG event lists and ACCUM images are fully calibrated by **calcos**. Target acquisition exposures are not calibrated by **calcos**, except for ACQ/IMAGE, although the raw data from these acquisitions are available through the data archive.

As with *HST* calibration pipelines for other instruments, the choice of which operations are performed during calibration is controlled by calibration switches, which are stored in the primary FITS header. **DP** sets the switches that are appropriate for a given data type to PERFORM for steps to be carried out by **calcos**, and then **calcos** changes them to COMPLETE or SKIPPED in the calibrated files. When **DP** creates the raw data files, it also populates the headers with the names of the appropriate reference files for each calibration operation. Any calibration step may require zero, one, or more calibration reference files. Exactly how the data are processed depends on whether they are FUV TIME-TAG or ACCUM spectra, NUV TIME-TAG or ACCUM spectra, or NUV images. The names of the keywords containing the switches and reference file names were introduced in Table 2.16, Table 2.17, and Table 2.18, and their roles in the data reduction and the calibration steps are described in the following sections.

Figure 3.1–Figure 3.6 show how a single raw file moves through the pipeline for FUV TIME-TAG (TWOZONE), FUV TIME-TAG (BOXCAR), FUV ACCUM, NUV TIME-TAG and NUV ACCUM spectroscopic data, and for NUV images. Each figure shows, from left to right, the input files, the processing steps performed by each module, and the output files. Note that in some instances, output files are created and then subsequently modified. In these cases, the file is shown at the end of a dashed arrow at the point it is created and again by a solid arrow at the point where it is finalized. Steps that apply only when spectra are extracted are marked with an * in Figure 3.1 through Figure 3.5. FUV data taken after February 9, 2015 will use the TWOZONE extraction by default, except for the blue modes G130M/1055 and 1096. Users can also confirm which extraction algorithm was used to generate a given FUV data file using the XTRCTALG keyword, which will be set to either BOXCAR or TWOZONE.

For ACCUM data, Doppler corrections are done onboard. Consequently, for these spectra certain reference files are transformed to the coordinate system of the data, rather than the other way around. We note in Figure 3.3 and Figure 3.5 when this is done.

Figure 3.1: FUV `TIME-TAG` Spectroscopic Pipeline Flow Chart (TWOZONE).

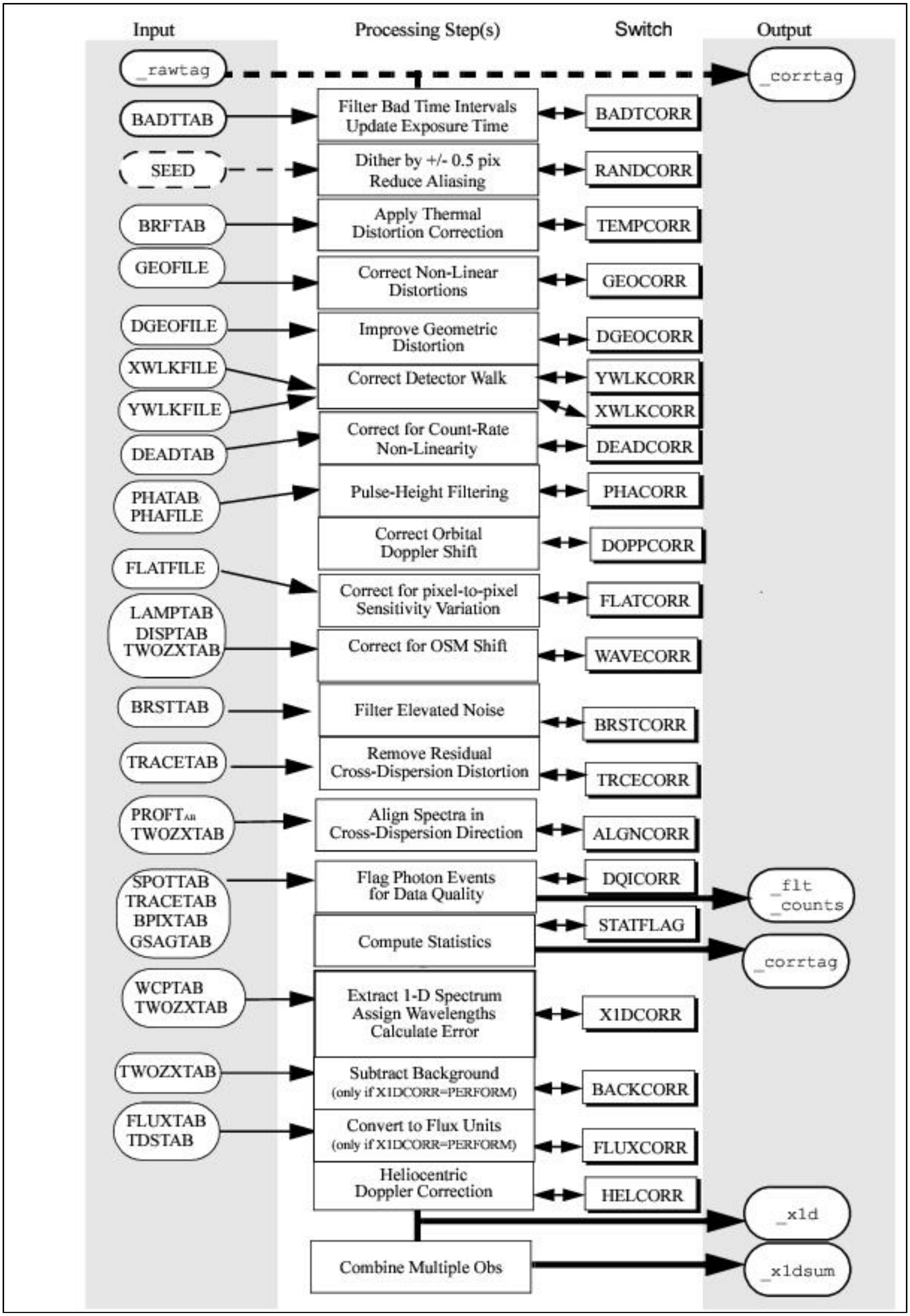


Figure 3.2: FUV `TIME-TAG` Spectroscopic Pipeline Flow Chart (BOXCAR).

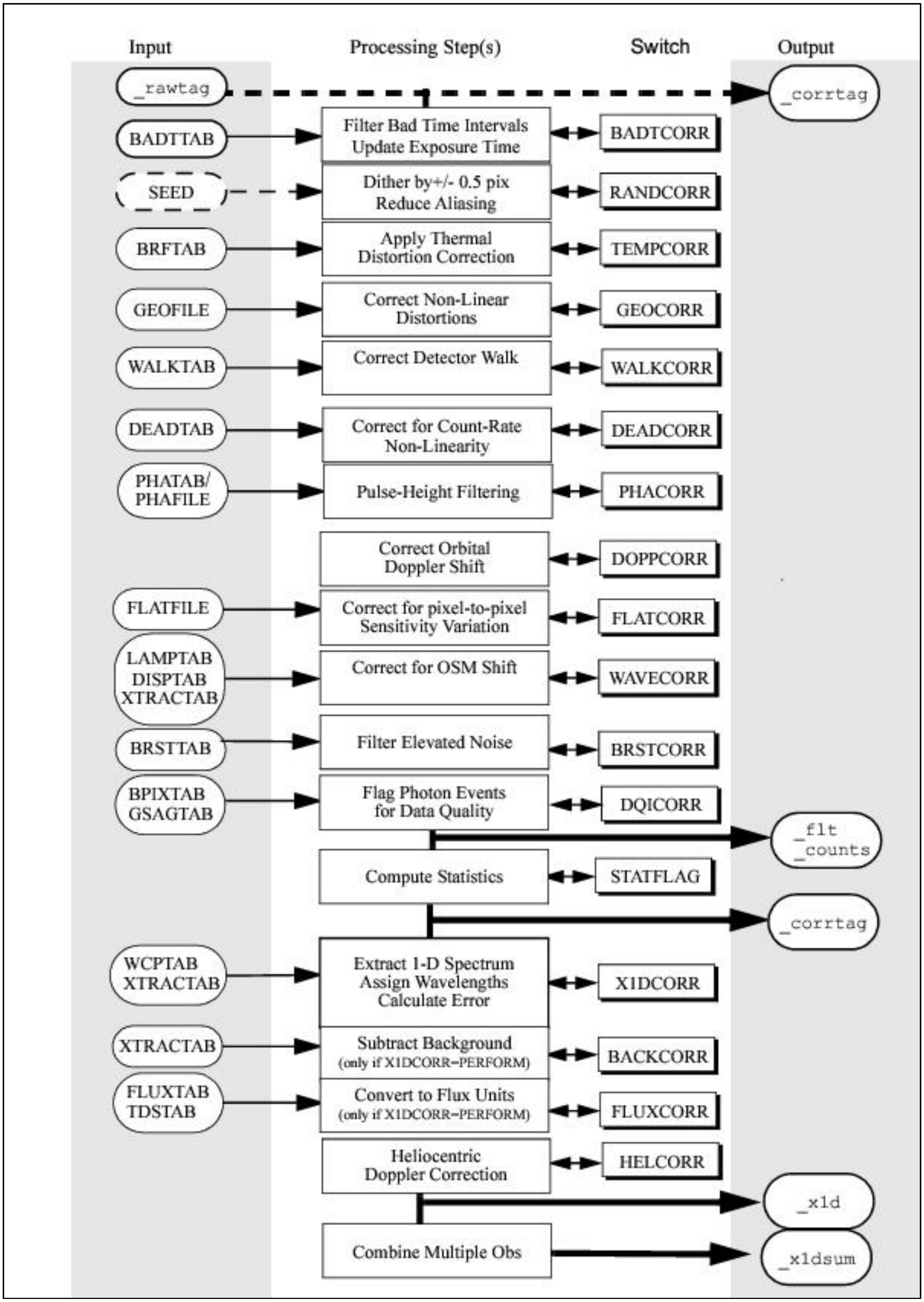
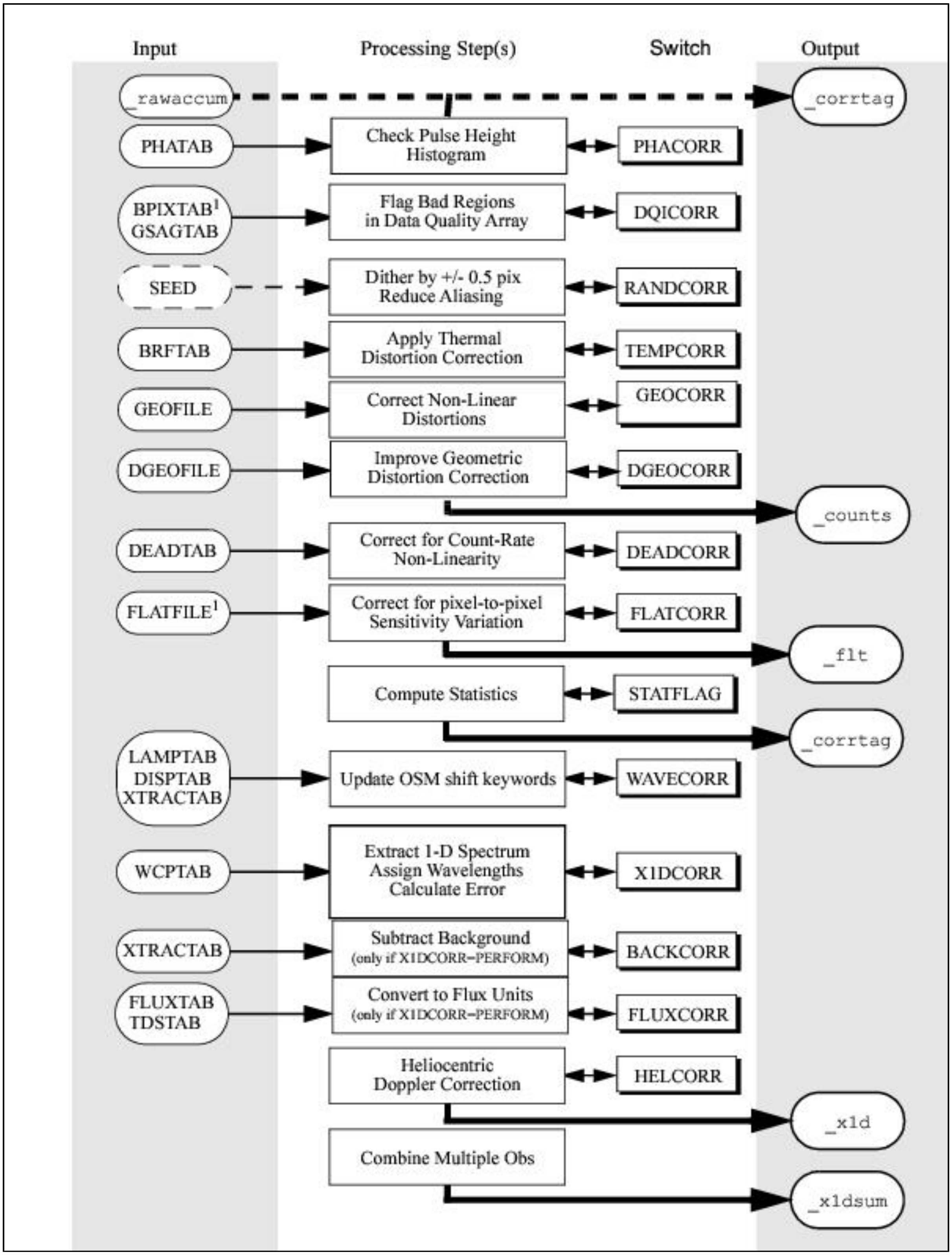


Figure 3.3: FUV ACCUM Spectroscopic Pipeline Flow Chart.



[1] Reference files that are transformed to the Doppler-corrected coordinate system of the data before being applied.

Figure 3.4: NUV `TIME-TAG` Spectroscopic Pipeline Flow Chart.

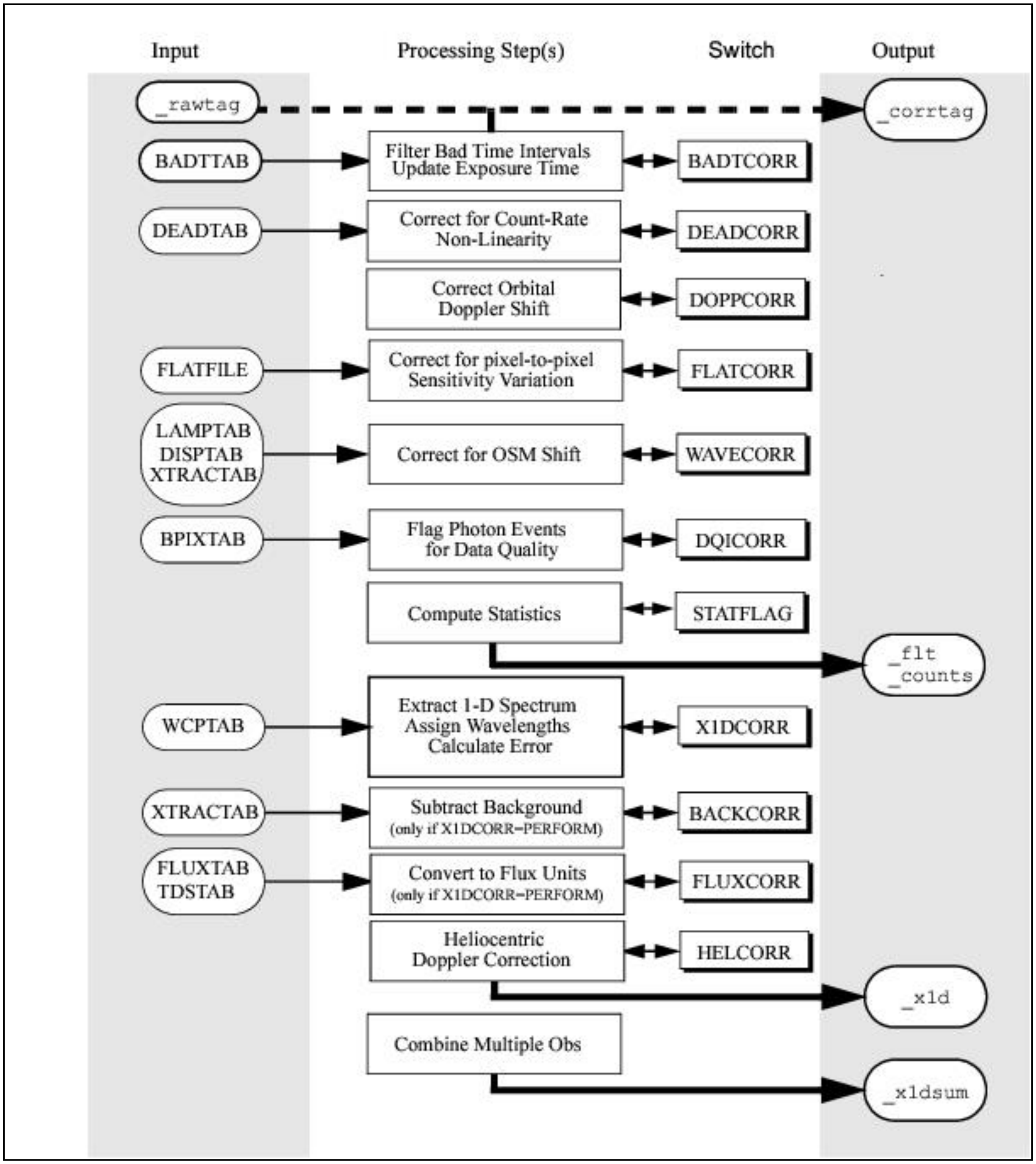
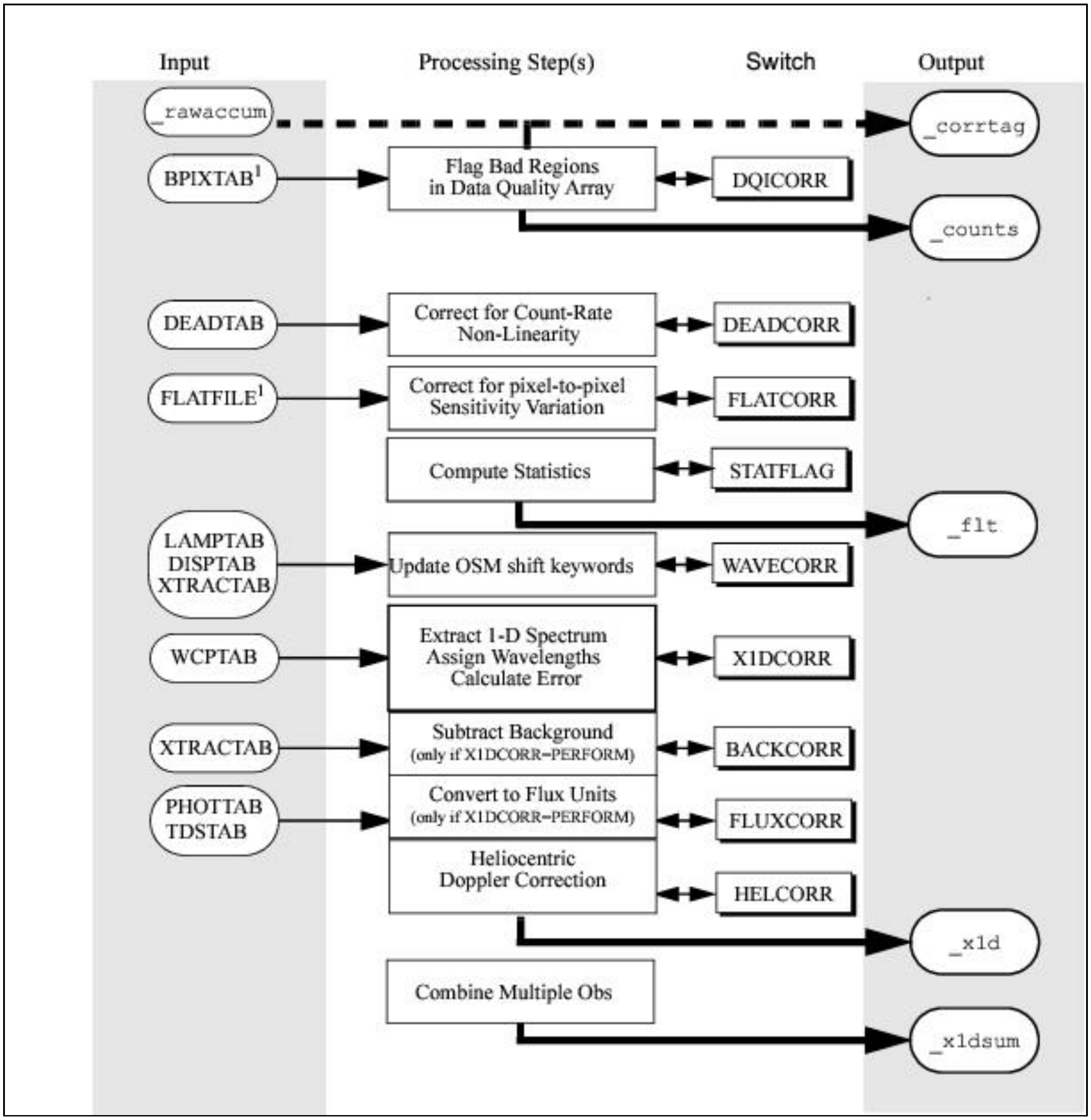
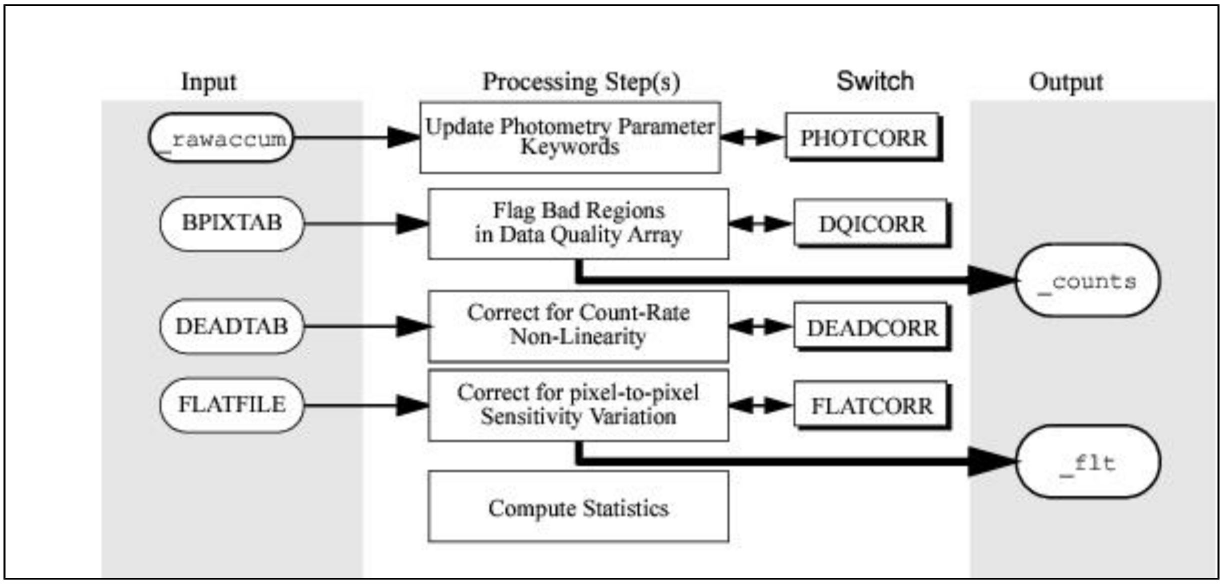


Figure 3.5: NUV ACCUM Spectroscopic Pipeline Flow Chart.



[1] Reference files that are transformed to the doppler corrected coordinate system of the data before being applied.

Figure 3.6: NUV Image Pipeline Flow Chart.



3.4 Descriptions of Spectroscopic Calibration Steps

This section provides a more detailed description of the calibration processing steps and algorithms applied by **calcos** (v3.2.1 and later), including the switches, reference file inputs, science file inputs, and the output products. Setting the calibration switch to PERFORM enables the execution of the corresponding pipeline calibration task while setting to OMIT will cause that task to be skipped.

Future modifications and updates to **calcos** will be announced in STScI Analysis Newsletters (STANs), **calcos** release notes, and documented at the COS Web site:

http://www.stsci.edu/hst/instrumentation/cos/documentation/stsci-analysis-newsletter-stan
http://www.stsci.edu/hst/instrumentation/cos/documentation/calcos-release-notes
http://www.stsci.edu/hst/instrumentation/cos

To provide a concrete example, the calibration steps for FUV `TIME-TAG` data in the pipeline processing flow are described next. When present, each sub-section header begins with the switch that activates the module.

3.4.1 Initialization

When the pipeline is initiated, it first opens an association file to determine which files should be processed together. For `TIME-TAG` data (but not `ACCUM` data), it also creates a `corrtag` file before anything else is done. The initial contents of this file are simply a copy of the `rawtag` file, except that new columns have been added to the `corrtag` file. It is then updated throughout the running of the pipeline.

3.4.2 BADTCORR: Bad Time Intervals

This module flags time intervals in `TIME-TAG` data that have been identified as bad.

- Reference file: `BADTTAB`
- Input files: `rawtag`
- Header keywords updated: `EXPTIME`, `EXPTIMEA` and `EXPTIMEB` (for FUV data), `NBADT`, or `NBADT_A` and `NBADT_B` (number of events flagged for NUV or FUVA and B, respectively) and `TBADT` or `TBADT_A` and `TBADT_B` (time lost to bad events in NUV or FUVA and FUVB, respectively).

The `BADTTAB` table lists zero or more intervals of time which will be excluded from the final spectrum for various reasons. This file is currently empty (as of May 2018), but it could be populated by the COS team if events occur on orbit which render data collected during specific time intervals not scientifically useful. It is also available for the convenience of the user. For example, the user may wish to eliminate observations obtained in the daytime portion of the orbit to minimize airglow contamination, or they may want to isolate a certain portion of an exposure. In these cases, modifying `BADTTAB` may be the most convenient means to accomplish this. Events in the `rawtag` file having times within any bad time interval in `BADTTAB` are flagged in the DQ column of the `corrtag` table with data quality = 2048. The exposure time is updated to reflect the sum of the good time intervals, defined in Section 2.4.1. This step applies only to `TIME-TAG` data.

3.4.3 RANDCORR: Add Pseudo-Random Numbers to Pixel Coordinates

This module adds a random number between –0.5 and +0.5 to each x and y position of a photon detected by the FUV detectors.

- Reference file: none
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: `RANDSEED`

For FUV `TIME-TAG` data RANDCORR adds random numbers to the raw coordinates of each event, i.e.:

$$\mathrm{XCORR} = \mathrm{RAWX} + \Delta\mathrm{x}$$
$$\mathrm{YCORR} = \mathrm{RAWY} + \Delta\mathrm{y}$$

Where Δx and Δy are uniformly distributed, pseudo-random numbers in the interval $-0.5 < \Delta x, \Delta y \leq +0.5$.

The result of this operation is to convert the raw integer pixel values into floating point values so that the counts are smeared over each pixel's area.

For FUV `ACCUM` data, a pseudo TIME-TAG list of x and y values is created with an entry for each recorded count. Next, a unique Δx and Δy are added to each entry.

If the `RANDSEED` keyword in the raw data file is set to its default value of –1, the system clock is used to create a seed for the random number generator. This seed value is then written to the `RANDSEED` keyword in the output files. Alternatively, an integer seed (other than –1) in the range –2147483648 to +2147483647 can be specified by modifying the `RANDSEED` keyword in the raw data file. Doing so will ensure that identical results will be obtained on multiple runs.

RANDCORR is only applied to events in the active area of the detector, as defined in the `BRFTAB`. Stim pulses, for example, do not have this correction applied.

3.4.4 TEMPCORR: Temperature-Dependent Distortion Correction

This module corrects for linear distortions of the FUV detector coordinate system that are caused by changes in the temperature of the detector electronics.

- Reference file: `BRFTAB`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: none

The FUV XDL detector has virtual, not physical, detector elements that are defined by the digitization of an analog signal. The charge packet associated with a photon event is split and transported to opposite sides of the detector where the difference in travel time of the two packets determines the location of the photon event on the detector. Since the properties of both the delay line and the sensing electronics are subject to variations as a function of temperature, apparent shifts and stretches in the detector format can occur.

To measure the magnitude of this effect, electronic pulses (Figure 1.7) are recorded at two reference points in the image ("electronic stim pulses") at specified time intervals throughout each observation. TEMPCORR first determines the locations and separations of the recorded stim pulse positions and then compares them to their expected locations in a standard reference frame (as defined in columns SX1, SY1, SX2, and SY2 of the `BRFTAB` file). The differences between the observed and reference stim pulse positions are used to construct a linear transformation between the observed and reference frame locations for each event (or pseudo-event in the case of `ACCUM` data). TEMPCORR then applies this transformation to the observed events, placing them in the standard reference frame. The stim pulse parameters are written to the file headers using the keyword names described in Table 2.15. In cases where one of the stim pulses falls off the active area of the detector, **calcos** assumes it is in its normal position and outputs a warning before continuing with calibration. This may significantly affect the reliability of the wavelength scale.

3.4.5 GEOCORR and IGEOCORR: Geometric Distortion Correction

This module corrects geometric distortions in the FUV detectors.

- Reference file: `GEOFILE`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: none

The GEOCORR module corrects for geometric distortions due to differences between the inferred and actual physical sizes of pixels in the FUV array (ground measurements indicated that geometric distortions in the NUV MAMA are negligible). It produces a geometrically-corrected detector image with equally sized pixels. This is done by applying the displacements listed in the reference file, GEOFILE, which lists the corrections in x and y for each observed pixel location. The geometric distortion varies across the detector, and the GEOFILE gives the distortion only at the center of each pixel.

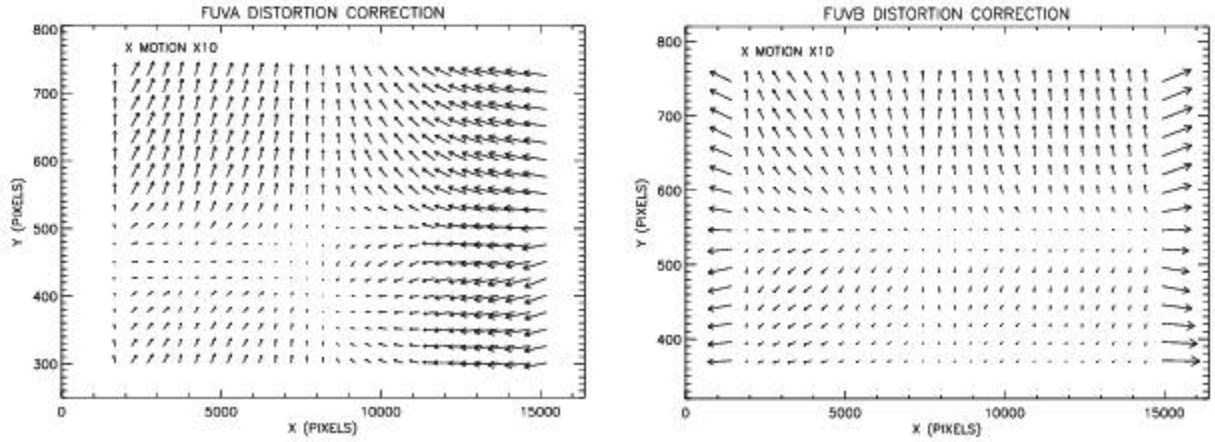
If IGEOCORR is set to 'PERFORM' (the default), the displacements to correct the distortion at (XCORR, YCORR) will be interpolated to that location, which includes a fractional part (even before geometric correction) due to TEMPCORR and RANDCORR. If IGEOCORR is 'OMIT,' the correction will be taken at the nearest pixel to (XCORR, YCORR).

`GEOFILE` was created by using a ray-trace analysis of the COS FUV optical system. A set of wavelength calibration exposures was taken while stepping the aperture mechanism in the cross-dispersion direction to create an image of dispersed line profiles. The ray trace and measured line positions were compared to determine the shift between the measured (uncorrected) and predicted (corrected) line positions (see Figure 3.7).

The distortion corrections are given as images in the GEOFILE in the following order:

- Extension 1 contains an image of the X distortions for the FUVA
- Extension 2 contains an image of the Y distortions for the FUVA
- Extension 3 contains an image of the X distortions for the FUVB
- Extension 4 contains an image of the Y distortions for the FUVB

Figure 3.7: A Map of the FUV Geometric Correction.



A map of the FUV geometric correction, scaled by a factor of 10 in the x-direction for illustration purposes, for detector segment A (left) and segment B (right) in user coordinates. The arrow points from the observed to the corrected position.

3.4.6 DGEOCORR: Delta Geometric Distortion Correction

This module further corrects geometric distortions in the FUV detectors in addition to normal geometric correction (GEOCORR). This module is designed to run after GEOCORR and to operate in the same manner, but cannot be done without first performing GEOCORR.

- Reference file: `DGEOFILE`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: none

The DGEOCORR module is designed to improve upon the GEOCORR corrections by removing residual geometric distortions that remain after the geometric correction has been applied. The correction is done by applying the displacements listed in the reference file, `DGEOFILE`, which lists the corrections in x and y for each observed pixel location.

If IGEOCORR is 'PERFORM' (the default), the displacements to correct the distortion at (XCORR, YCORR) will be interpolated to that location. If IGEOCORR is 'OMIT,' the correction will be taken at the nearest pixel to (XCORR, YCORR). Note that the use or omission of IGEOCORR applies to both GEOCORR and DGEOCORR as it is not appropriate to use IGEOCORR for only GEOCORR or DGEOCORR. DGEOCORR cannot be set to 'PERFORM' unless GEOCORR is also set to 'PERFORM', as DGEOCORR assumes that the GEOCORR has already been performed.

The `DGEOFILE` distortion corrections are in the same format to those of the `GEOFILE` and are given as images in the following order:

- Extension 1 contains an image of the X distortions for the FUVA
- Extension 2 contains an image of the Y distortions for the FUVA
- Extension 3 contains an image of the X distortions for the FUVB
- Extension 4 contains an image of the Y distortions for the FUVB

3.4.7 XWLKCORR, YWLKCORR: Walk Correction

These modules correct for the fact that the reported position of events on the FUV XDL detector is a function of pulse height (an effect known as walk). XWLKCORR corrects for walk effects in the X (dispersion) axis; YWLKCORR corrects for walk in the Y (cross-dispersion) direction.

- Reference files: `XWLKFILE`, `YWLKFILE`
- Input file: `rawtag`
- Header keywords updated: none

The file formats for the `XWLKWFILE` and `YWLKFILE` are identical. Each contains two binary extensions with 16384 × 32 IMAGE extensions. One extension (EXTNAME = 'FUVA') is used to correct the data on Segment A, and the other (EXTNAME = 'FUVB') is used for Segment B. These images are used as lookup tables, with the columns (0–16383) of each table corresponding to the thermally and geometrically corrected X-detector pixel, and the rows of each table corresponding to the pulse height value (0–31). Note that both X- and Y-walk corrections are only a function of the X pixel and independent of Y.

Since the X-coordinate is a floating point value after thermal and geometric corrections, the correction is found by interpolating the adjacent values found in the table. The calculated value is then subtracted from X or Y to give the final, corrected location.

In reality, the walk and geometric distortions occur in parallel in the detector, i.e., the raw (x,y) position reported by the electronics is a function of both position and pulse height. In fact, an alternative way of thinking about these effects is that the detector has a different geometric correction at each PHA value. However, because of the way the data used for the geometric correction was obtained, and for simplicity in implementation, the two are done serially in **calcos**.

The format of the walk correction was changed significantly in **calcos** version 3.2.1. Prior to that version, a polynomial in X and PHA was used to apply the correction using the WALKCORR routine. However, that method did not provide a fine enough correction with polynomials of reasonable order.

3.4.8 DEADCORR: Non-linearity Correction

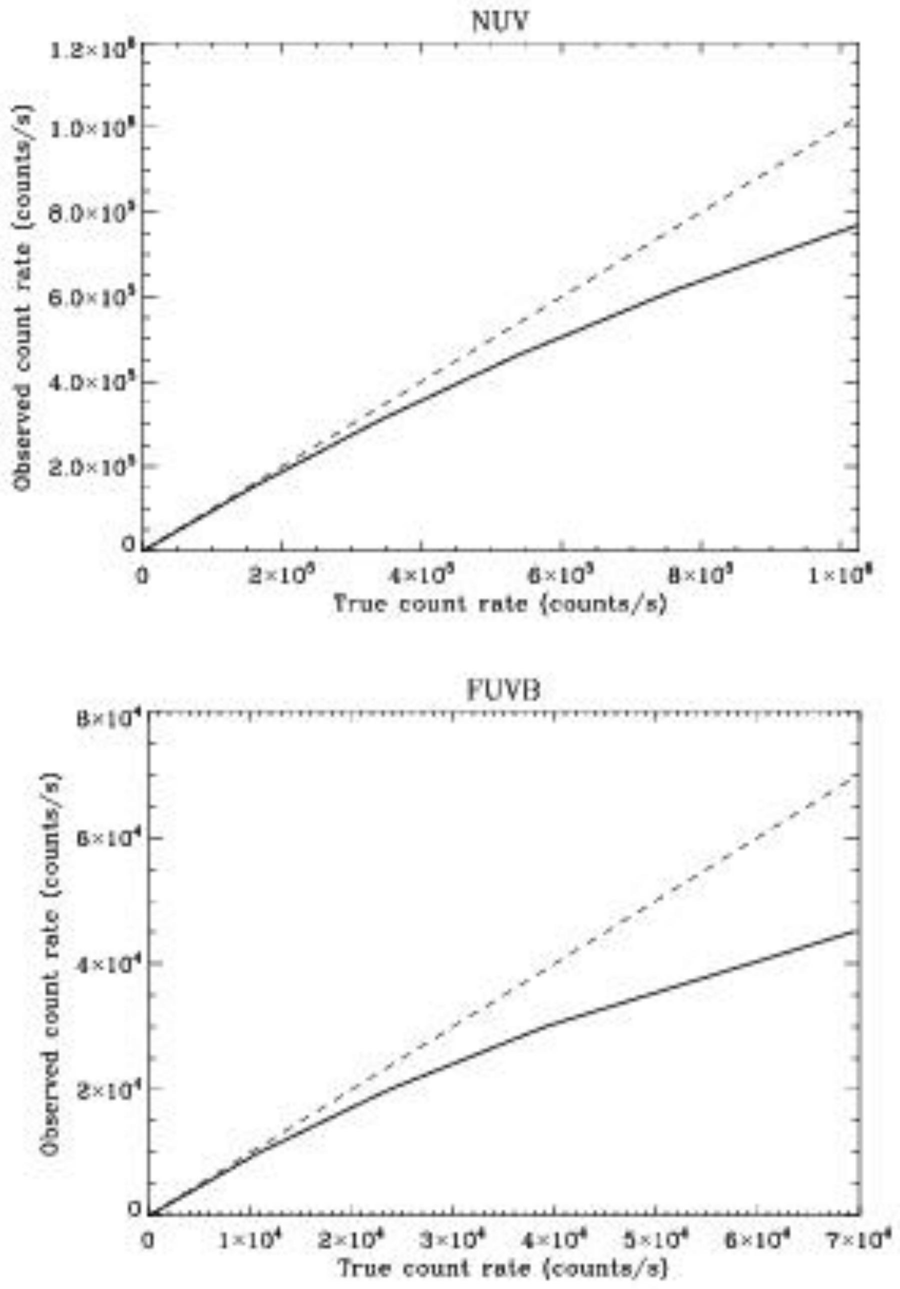
This module corrects for count rate-dependent non-linearities in the COS detectors, also known as the "dead time" correction.

- Reference file: `DEADTAB`
- Input files: `rawtag`, `rawaccum`, `images`
- Header keywords updated: none

DEADCORR corrects for non-linear photon detection in the COS detector electronics. Both the FUV and NUV detector electronics have a small temporal overhead when counting events. This overhead becomes noticeable when the count rates become large.

The efficiency of the detector's photon counting is calculated as the ratio of the true count rate and the observed count rate. This value is referred to as the deadtime. The deadtime for each detector is modeled and the reference file `DEADTAB` contains a lookup table of the correction for various count rates. Figure 3.8 shows how the measured count rates deviate from the actual count rates as a function of the actual count rate for the NUV detector, and segment B of the FUV detector (the FUV segment A curve is nearly identical).

Figure 3.8: FUV and NUV Deadtime.



The solid curves are the observed count rates versus true count rates for the COS detectors and the dashed lines are for perfect detectors. *Top*: The NUV MAMA. *Bottom*: Segment B of the FUV XDL detector (the curve for Segment A is nearly identical). Significant deviations from the true count rates occur at about 15,000 counts per second for the XDL detectors, and at roughly 10 times this rate for the MAMA.

For `TIME-TAG` data the deadtime correction is computed every 10 seconds. The observed count rate is the number of events within that time interval, and the deadtime factor is determined by interpolation within the values in `DEADTAB`. The values in the EPSILON column in the `corrtag` file for events within that time interval will then be divided by the deadtime factor. For `ACCUM` data the observed average count rate taken from a header keyword for the digital event counter is used. The deadtime factor is then found by interpolation within the `DEADTAB`, the same as for `TIME-TAG` data, and the science and error arrays divided by the deadtime factor. The deadtime correction parameters are written to the file headers using the keyword names described in Table 2.15.

3.4.9 PHACORR: Pulse Height Filter

This module operates on FUV data and flags events whose pulse heights are outside of nominal ranges.

- Reference file: `PHATAB`, `PHAFILE`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: `NPHA_A`, `NPHA_B`, `PHAUPPRA`, `PHAUPPRB`, `PHALOWRA`, `PHALOWRB`

This module works differently for FUV `TIME-TAG` and `ACCUM` data. It is not used for NUV data.

For FUV TIME-TAG data, each event includes a 5 bit (0–31) Pulse Height Amplitude (PHA). The value of the pulse height is a measure of the charge produced by the microchannel plate stack, and can be used to identify events which are likely due to cosmic rays or detector background. The `PHATAB` reference file lists lower and upper pulse height thresholds expected for valid photon events for each detector segment. The PHACORR module compares each event's pulse height to these thresholds, and if the pulse height is below the Lower Level Threshold (LLT) or above the Upper Level Threshold (ULT), the event is flagged in the DQ column of the `corrtag` table with a data quality bit of 512. The upper and lower thresholds are also written to the `PHALOWRA` (`PHALOWRB`) and `PHAUPPRA` (`PHAUPPRB`) keywords in the output data files for segment A (B), while the number of events flagged is written to the `NPHA_A` and `NPHA_B` keywords.

Default values of the lower (LLT) and upper (ULT) thresholds have been chosen based on the properties of the detector and are implicit in data used when generating other reference files (e.g., `FLUXTAB`).

With continuing exposure to photons, pulses from the micro channel plates (MCPs) have smaller amplitudes, a phenomenon known as "gain sag." As this occurs, the thresholds in the `PHATAB` may be updated to maximize the number of real events counted. Which `PHATAB` is used for data collected at a particular time will be handled by the USEAFTER date keyword in the calibration file header.

The `PHAFILE` reference file is an alternative to the `PHATAB`, and allows pulse-height limits to be specified on a per-pixel basis rather than a per-segment basis. The `PHAFILE` has a primary header and four data extensions, consisting of the FUVA PHA lower limits, FUVA PHA upper limits, FUVB PHA lower limits, and FUVB PHA upper limits respectively. The use of a `PHAFILE` instead of a `PHATAB` (if both are specified and `PHACORR=PERFORM`, the `PHAFILE` will take precedence) allows a number of adjustments, including (for example) the use of a lower PHA threshold in gain-sagged regions, thus allowing more background events to be filtered out while still continuing to detect photon events in gain-sagged regions. As of May 2018, no `PHAFILE` has been produced by the COS team, but in the future one or more such files may be produced for use with FUV `TIME-TAG` data. Note that the use of a `PHAFILE` requires **calcos** 2.14 or later.

⚠ ***Modifying the pulse height threshold values could lead to incorrect results in the calibrated products, and should therefore be done with extreme caution.***

For FUV `ACCUM` data, pulse height information is not available for individual events. However, a 7 bit (0–127) Pulse Height Distribution (PHD) array, containing a histogram of the number of occurrences of each pulse height value over the entire detector segment, is created onboard for each exposure. PHACORR compares the data in this `pha` file to the values in the `PHATAB` file. Warnings are issued if the peak of the distribution (modal gain) does not fall between the scaled values of LLT and ULT; or if the average of the distribution (mean gain) does not fall between the MIN_PEAK and MAX_PEAK values in `PHATAB`. The `PHALOWRA` and `PHAUPPRA`, or `PHALOWRB` and `PHAUPPRB` keywords are also populated in the output files with the LLT and ULT values from the `PHATAB`.

3.4.10 DOPPCORR: Correct for Doppler Shift

This module corrects for the effect that the orbital motion of *HST* has on the arrival location of a photon in the dispersion direction.

- Reference files: `DISPTAB`, `XTRACTAB` or `TWOZXTAB`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: none

During a given exposure the photons arriving on the FUV and NUV detectors are Doppler shifted due to the orbital motion of *HST*. The orbital velocity of *HST* is 7.5 km/s, so spectral lines in objects located close to the orbital plane of *HST* can be broadened up to 15 km/s, which can be more than a resolution element.

DOPPCORR corrects for the orbital motion of *HST*. It operates differently on TIME-TAG and ACCUM files:

For TIME-TAG files the raw events table contains the actual detector coordinates of each photon detected, i. e., the photon positions will include the smearing from the orbital motion. In this case DOPPCORR will add an offset to the pixel coordinates (the XCORR column) in the events table to correct for this motion. The corrected coordinates are written to the column XDOPP in the `corrtag` file for both FUV and NUV data.

For ACCUM files the Doppler correction is applied onboard and is not performed by **calcos**. This means, however, that the pixel coordinates of a spectral feature can differ from where the photon actually hit the detector—a factor which affects the data quality initialization and flat-field correction. Therefore for ACCUM images DOPPCORR shifts the positions of pixels in the bad pixel table BPIXTAB to determine the maximum bounds that could be affected. It is also used to convolve the flat-field image by an amount corresponding to the Doppler shift which was computed on orbit. The information for these calculations are contained in the following header keywords:

- DOPPONT: True if Doppler correction was done onboard.
- ORBTPERT: Orbital period of *HST* in seconds.
- DOPMAGT: Magnitude of the Doppler shift in pixels.
- DOPZEROT: Time (in MJD) when the Doppler shift was zero and increasing.

The "T" suffix at the end of each of these keywords indicates that they were derived from the onboard telemetry, whereas the other keywords described below were computed on the ground from the orbital elements of *HST*. The two sets of keywords can differ by a small amount, but they should be nearly identical.

DOPPCORR assumes that the Doppler shifts vary sinusoidally with time according to the orbital movement of *HST*. The following keywords are used to perform the correction and are obtained from the first extension (EVENTS) in the `rawtag`:

- `EXPSTART` – start time of the exposure (MJD)
- `DOPPZERO` – the time (MJD) when the Doppler shift was zero and increasing (i.e., when *HST* was closest to the target)
- `DOPPMAG` – The number of pixels corresponding to the Doppler shift (used only for shifting the data quality flag arrays and flat fields)
- `ORBITPER` – the orbital period of *HST* in seconds

The data columns used in the correction are TIME (elapsed seconds since `EXPSTART`) and RAWX (position of photon along dispersion direction). The Doppler correction to be applied is then

$$\mathrm{SHIFT} = -(\mathrm{DOPPMAG\ V}/(\mathrm{c} * \mathrm{d})) * \lambda(\mathrm{XCORR}) * \sin(2 * \pi * \mathrm{t}/\mathrm{ORBITPER})\ ,$$

where DOPPMAGV is the Doppler shift magnitude including the pointing direction of *HST*, c is the speed of light (km/s), d is the dispersion of the grating used in the observation (Å/pixel), λ(XCORR) is the wavelength at the XCORR position being corrected (obtained from the dispersion solution for that grating and aperture in the `DISPTAB` reference file) and t is defined as

$$t = (\mathrm{EXPSTART} - \mathrm{DOPPZERO}) * 86400 + \mathrm{TIME}\ ,$$

where the factor of 86400 converts from days to seconds.

3.4.11 FLATCORR: Flat-field Correction

This module corrects for pixel-to-pixel non-uniformities in the COS detectors.

- Reference file: `FLATFILE`
- Input files: `rawtag`, `rawaccum`, `images`
- Header keywords updated: none

The FLATCORR step corrects for pixel-to-pixel sensitivity differences across the detector. It uses a flat-field image located in the file specified by the `FLATFILE` header keyword. Figure 3.9 shows an NUV flat. For spectroscopic data, any wavelength dependence of the detector response or remaining low frequency spatial variations are removed by the flux calibration step (FLUXCORR, Section 3.4.20). Flat fielding is performed in geometrically corrected space, and because the pixel-to-pixel variations should be largely wavelength independent, only one reference image is used per detector or detector segment (NUV, FUVA, and FUVB). The flat-field correction is applied differently for `TIME-TAG` and `ACCUM` mode data for both spectroscopic and imaging modes.

For spectroscopic `TIME-TAG` exposures, each photon in the events list is individually corrected. In the `corrtag` file, the photon weight in the EPSILON column is divided by the flat-field value at the event's detector location rounded to the nearest pixel (XCORR, YCORR for FUV; RAWX, RAWY for NUV).

For spectroscopic `ACCUM` mode data, photons are summed into an image onboard by the COS electronics. To compensate for the motion of *HST* during the observation, spectroscopic exposures are taken with Doppler compensation performed during the accumulation (science header keyword `DOPPONT=TRUE`). During Doppler compensation, photon locations are shifted as the data are received, and the underlying flat field at each imaged pixel is an average of the original pixel position sensitivities. FLATCORR replicates this averaging for the flat-field correction using the same control parameters as those onboard (`DOPPMAGT`, `DOPZEROT`, `ORBTPERT`) if `DOPPCORR=PERFORM` (Section 3.4.10). The convolved flat-field image is applied to the EPSILON column in the pseudo-corrtag file.

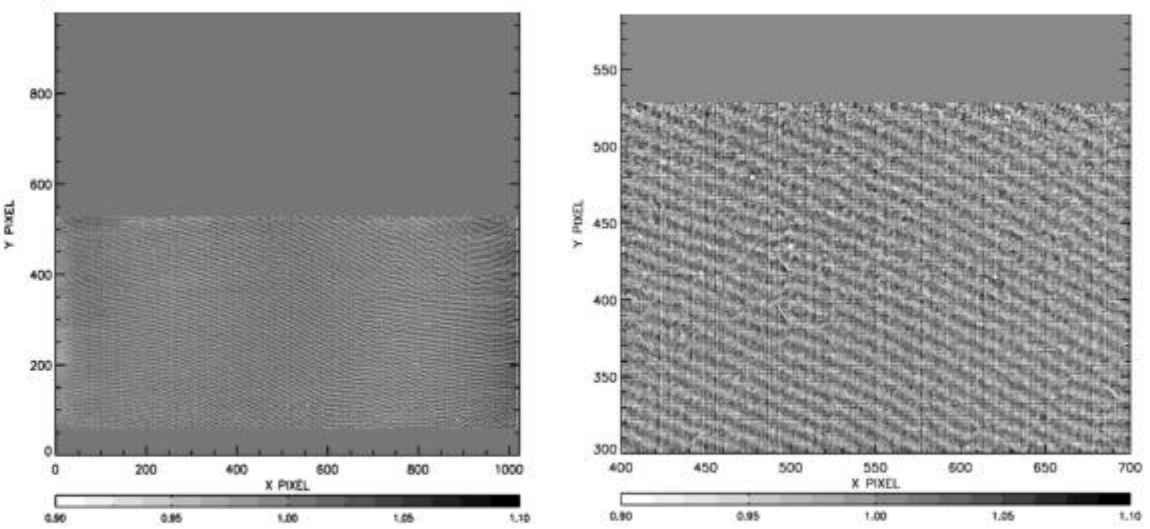
NUV images using the mirrors are not Doppler corrected. In this case, `DOPPCORR=OMIT`, and the input data are divided by the flat field without convolution.

For both the `flt` and `counts` files, error arrays are created based on counting statistics (Section 2.7), but they are not used in further processing.

It was discovered on-orbit that the NUV suffers some vignetting. This causes a structure in "pixel space," affecting roughly the first 200 pixels of all three spectral stripes by as much as 20%. The NUV flat field was originally modified to correct for this effect, but variation in the vignetting caused sufficient errors that the vignetting is no longer included in the NUV flat field. Work continues on grating-specific vignetting corrections.

For the FUV channels, the ground flats proved inadequate. Consequently, the current FUV flats correct primarily for the effects of grid wires and low order flat field variations (see COS ISR 2013-09 and COS ISR 2016-15).

Figure 3.9: Flat-field Images of the NUV MAMA Detector.



The image at left shows the full detector, and the one on the right has been enlarged to illustrate structure in the flat-field images. The hex structure associated with the microchannel plate is visible in both FUV and NUV flat fields.

3.4.12 WAVECORR: Wavecal Correction

For spectroscopic data, this module determines the location of the wavelength calibration spectrum on the detector relative to a template, and then applies zero point shifts to align the wavecal and the template.

- Reference files: `LAMPTAB`, `WCPTAB`, `DISPTAB`, `XTRACTAB` or `TWOZXTAB`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: `SHIFT1[A-C], SHIFT2[A-C], LMP_ONi, LMPOFFi, LMPDURi, LMPMEDi.`
- Creates `lampflash` file for TAGFLASH data.

The wavecal step of **calcos** determines the shift of the 2-D image on the detector along each axis resulting from thermal motions and drifts within an OSM (Optics Select Mechanism) encoder position. This step applies only to spectroscopic data, `TIME-TAG` and `ACCUM`, for both the FUV and NUV detectors. The shifts are determined from one or more contemporaneous wavelength calibration observations of a spectral line lamp (wavecal) which must be obtained without moving the OSM between the science and wavecal exposures.

There are four types of wavecals, as described in the COS IHB sec. 5.7. For `ACCUM` data the spectrum of the calibration lamp is contained in an exposure that is separate from that of the science (AUTO or GO wavecals). For `TIME-TAG` data the wavecals can also be separate exposures, but the default when observing with the PSA aperture is `TAGFLASH` mode. In the `TAGFLASH` mode the line lamp is turned on and off (flashed) one or more times during each science exposure, producing a wavecal spectrum that is offset in the cross-dispersion direction from the science spectrum (See Figure 1.9, and Figure 1.10). The algorithm used to determine the shifts is the same in either case, but the way that the shift is determined at the time of the observation differs. Thus, we begin by describing how the offsets are found.

Determining the offsets of the wavecal spectra:

For each wavecal, the location of the spectrum in the cross-dispersion direction is determined by collapsing the spectrum along the dispersion direction using the extraction slope defined in the `XTRACTAB` table (SLOPE). The location of the brightest pixel, after boxcar smoothing, is taken as the spectrum location and that location is compared to the nominal position defined in the `XTRACTAB` table (B_SPEC). The offsets from nominal positions for segments A and B (FUV) or stripes A, B, and C (NUV) are recorded in the `lampflash` file (which is created at this stage) in the `SHIFT_XDISP` field. The two FUV segments are processed independently. Cross-dispersion shifts are determined for each NUV stripe and then the average is computed and applied to all three stripes. The sign of the `SHIFT_XDISP` entry is positive if the spectrum was found at a larger pixel number than the nominal location.

To determine the offsets in the dispersion direction, the wavecal spectrum is collapsed along the cross-dispersion direction and compared to the template wavecal (`LAMPTAB`) taken with the same grating, central wavelength, and FPOFFSET. For the NUV, wavecal spectra offsets for each stripe are determined independently. The line positions are determined from a least squares fit to a shifted and scaled version of the template spectrum. The maximum range for shifting the wavecal and template wavecal spectra is defined by the value of `XC_RANGE` in the `WCPTAB` table. **Calcos** takes into account the `FP-POS` of the wavecal spectrum by shifting it by `FP_PIXEL_SHIFT` (from the column in the `LAMPTAB`) where `FP_PIXEL_SHIFT=0` for FP-POS=3 and all other FP-POS settings are shifted to the FP-POS=3 position before fitting them to the template wavecal. The final shift is stored as `SHIFT_DISP` in the lampflash file and the minimum value of chi squared is stored in the `CHI_SQUARE` array.

Applying the offsets to the science spectra:

The way the offsets are applied to the spectral data depends on whether the data were obtained with AUTO or GO wavecals or with `TAGFLASH` wavecals. For AUTO or GO wavecals, the wavecals are obtained at different times than the spectral data and temporal interpolation is done to determine the appropriate shifts. For `TAGFLASH` data, the wavecals are interspersed with the spectral data, allowing more precise and, consequently, more intricate corrections to be made. In either case, the result is saved in the `X[Y]FULL` entries in the `corrtag` file. Because the corrections can be time dependent, the differences between `X[Y]CORR` and `X[Y]FULL` can also be time dependent. This step of the calibration amounts to a time dependent translation of the detector coordinate system to a coordinate system relative to the wavecal spectrum, which is more appropriate for wavelength calibration.

AUTO or GO wavecals

For `ACCUM` science exposures which are bracketed by AUTO or GO wavecal observations, the shifts determined from the bracketing wavecal exposures are linearly interpolated to the middle time of the science observation, and the interpolated values are assigned to the `SHIFT1[A-C]` (dispersion direction) and `SHIFT2[A-C]` (cross-dispersion direction) keywords in the science data header. If there is just one wavecal observation in a dataset, or if there are more than one but they don't bracket the science observation, the `SHIFT1[A-C]` and `SHIFT2[A-C]` keywords are just copied from the nearest wavecal in the association to the science data header.

For non-`TAGFLASH TIME-TAG` science exposures bracketed by AUTO or GO wavecal observations, the shifts determined from the wavecals are interpolated (linearly) so that each event in the `corrtag` file is shifted according to its arrival time. The `SHIFT1[A-C]` and `SHIFT2[A-C]` keywords recorded in the science data header are in this case the averages of the values applied. As in the `ACCUM` case, if there is only one wavecal observation in a dataset, or if there are more than one but they do not bracket the science observation, the `SHIFT1[A-C]` and `SHIFT2[A-C]` keywords are just copied from the nearest wavecal to the science data header.

TAGFLASH DATA

A `TAGFLASH` wavecal is a lamp exposure that is taken concurrently with a `TIME-TAG` science exposure, and the photon events for both the wavecal lamp and the science target are mixed together in the same events table. In many respects, `TAGFLASH` wavecals are handled differently from conventional wavecals.

The nominal start and stop times for each lamp flash are read from keywords in the `corrtag` table. The actual start and stop times can differ from the nominal times, so **calcos** determines the actual times (restricted to being within the nominal start-to-stop intervals) by examining the number of photon events within each 0.2-second interval in the wavecal region defined in the `XTRACTAB` table. A histogram of the count rate is constructed. The histogram is expected to have one peak near zero, corresponding to dark counts, and another at high count rate, due to the lamp illumination. The average count rate when the lamp is on is taken to be the count rate for the second peak of the histogram. The lamp turn-on and turn-off times are taken to be the times when the count rate rises above or sinks below half the lamp-on count rate.

Calcos uses the time of the median photon event within a lamp turn-on and turn-off interval as the time of the flash. The keywords `LMP_ONi` and `LMP_OFFi` (i is the one-indexed flash number) are updated with the actual turn-on and turn-off times, in seconds, since the beginning of the science exposure. The keywords `LMPDURi` and `LMPMEDi` are updated with the actual duration and median time of the flash.

As before, the cross dispersion location of each wavecal spectrum is determined by collapsing it along the dispersion direction and comparing it with the template in the `XTRACTAB` table to produce the `SHIFT_XDISP` entries in the `lampflash` file. The wavecal spectrum is then collapsed along the cross-dispersion direction to produce a 1-D spectrum that is fit to the template spectrum to obtain the `SHIFT_DISP` entries.There is one row in the lampflash table for each flash. Typically there will be more than one wavecal flash during a science exposure; so the shifts will be piece-wise linearly interpolated between flashes. The `SHIFT1[A-C]` and `SHIFT2[A-C]` values that are recorded in the science data header are the average of the shift values found from the different flashes.

SPLIT Wavecals at LP6

Due to a light leak through the flat-field calibration aperture, wavecal lamps cannot be flashed on during a science exposure at LP6. Therefore, science exposures at LP6 are essentially bracketed by AUTO or GO wavecals, and the process for applying the offset to the science spectra is similar to that for AUTO or GO wavecals described above. However, for exposures longer than 960s, **calcos** will apply an additional wavelength shift, the magnitude of which has been determined by empirically modeling COS data. This is done to simulate a lamp flash that is typically taken for such longer length exposures. There is an estimated increase in the wavelength uncertainty of < 0.5 pixels for LP6 science exposures with this additional wavecal shift applied.

Additional Functions

WAVECORR also corrects the `flt` and `counts` files which result from both `ACCUM` and `TIME-TAG` science data for the offsets in the dispersion and cross-dispersion directions. However, since these images are in pixel space they can only be corrected by an integer number of pixels. The flt and counts images are corrected by the nearest integer to `SHIFT1[A-C]` and `SHIFT2[A-C]`. `DPIXEL1[A-C]` is the average of the difference between `XFULL` and the nearest integer to `XFULL`, where `XFULL` is the column by that name in the `corrtag` table. This is the average binning error in the dispersion direction when the `flt` and `counts` images are created from the corrtag table. `DPIXEL1[A-C]` is zero for ACCUM data. This shift is used when computing wavelengths during the X1DCORR step.

3.4.13 BRSTCORR: Search for and Flag Bursts

This module flags "event bursts" in the FUV `TIME-TAG` data for removal.

- Reference file: `BRSTTAB`, `BRFTAB`, `XTRACTAB` or `TWOZXTAB`
- Input files: `rawtag`
- Header keywords updated: `TBRST_A`, `TBRST_B` (time affected by bursts in segments A and B), `NBRST_A` , `NBRST_B` (number of events flagged as bursts in segments A and B), `EXPTIME`, `EXPTIMEA`, `EXPTIMEB`.

The COS FUV detectors are of the same design as the detectors used on the FUSE mission. The FUSE detectors were seen to experience sudden, short-duration increases in counts while collecting data. These events, called bursts, led to very large count rates and occurred over the entire detector. Thus far, no bursts have been recorded on-orbit for the COS FUV detectors, and the default setting for BRSTCORR is OMIT. Nevertheless, it is possible that COS will exhibit similar bursts at some point, and so the BRSTCORR module remains in the pipeline and is available to identify bursts and flag their time intervals should they occur. This module can only be applied to FUV `TIME-TAG` data.

The first step in the screening process is to determine the count rate over the whole detector, including stim pulses, source, background, and bursts. This rate determines which time interval from the `BRSTTAB` table to use for screening.

Screening for bursts is then done in two steps. The first step identifies large count rate bursts by calculating the median of the counts in the background regions, defined in the `XTRACTAB` reference file, over certain time intervals (DELTA_T or DELTA_T_HIGH for high overall count rate data). Events with count rates larger than MEDIAN_N times the median are flagged as large bursts.

The search for small count rate bursts is done iteratively, up to MAX_ITER. This step uses a boxcar smoothing of the background counts (taking the median within the box) and calculates the difference between the background counts and the running median. The boxcar smoothing is done over a time interval MEDIAN_DT or MEDIAN_DT_HIGH. Elements that have already been flagged as bursts are not included when computing the median. For an event to be flagged as affected by a small burst the difference between the background counts and the running median has to be larger than the following quantities:

1. A minimum burst count value: BURST_MIN * DELTA_T (or DELTA_T_HIGH for large overall count rates),
2. A predetermined number of standard deviations above the background: STDREJ * square_root (background counts),
3. A predetermined fraction of the source counts: SOURCE_FRAC * source counts.

The source counts value in 3) is the number of events in the source region defined in the `XTRACTAB` table minus the expected number of background counts within that region.

All events that have been identified as bursts are flagged in the data quality column (DQ in the `corrtag` table) with data quality bit = 64. In addition **calcos** updates the following header keywords to take into account time and events lost to burst screening: `TBRST_A` and `TBRST_B` (time lost to bursts in segments A and B); `NBRST_A`, `NBRST_B` (number of events lost to bursts in segments A and B), `EXPTIME`, `EXPTIMEA` and `EXPTIMEB`.

When running **calcos** a user can specify that the information about bursts be saved into a file. This output text file contains four columns, each with one row per time interval (DELTA_T or DELTA_T_HIGH). Column 1 contains the time (seconds) at the middle of the time interval, column 2 contains the background counts for that time interval, column 3 contains a 1 for time intervals with large bursts and is 0 elsewhere, and column 4 contains a 1 for time intervals with small bursts and is 0 elsewhere.

⚠ ***Note: Although a systematic study has not been performed, as of May 2018, no bursts have been detected.***

3.4.14 TRCECORR: Apply Trace Correction

This module corrects for the fact that the spectral trace from a target in the FUV channel is not completely straight, but wanders up and down by several pixels over the full wavelength range due to uncorrected detector distortions.

- Reference file: `TRACETAB`
- Input files: `rawtag`, `rawaccum`
- Header keywords updated: none

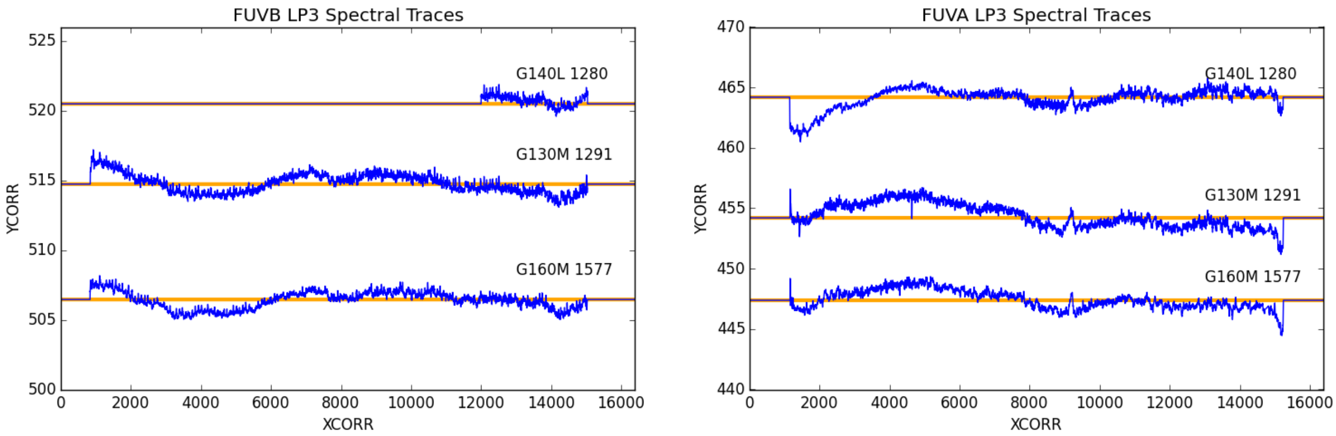
Even after the geometric distortion correction is applied in the FUV channel, there remains some residual distortion that shows up as an unevenness of the spectral trace as a function of wavelength.

TRCECORR uses a TRACETAB reference file, which is a FITS table containing a row for each valid combination of {SEGMENT, OPT_ELEM, APERTURE, CENWAVE}. SEGMENT can be FUVA or FUVB (the trace correction is not performed for NUV data), OPT_ELEM can be any of the FUV gratings (G140L, G130M, G160M), APERTURE can be PSA or BOA (no trace correction is performed on the WCA aperture), and CENWAVE can be any valid value for its corresponding grating. The reference file is selected based on the value of the INSTRUME and LIFE_ADJ keywords. Currently, the trace correction is only performed on data with LIFE_ADJ=3 or greater (i.e., LP3 or greater).

Each row in the table contains a table of values of the trace correction for each integer value of XCORR (1–6384). The correction is applied by looping over all events in the `corrtag` file, linearly interpolating the trace correction at the (non-integer) value of XCORR, and subtracting this value from the YFULL value of the event. Only events that are inside the active area and outside the WCA aperture are corrected.

The spectral traces at LP3 for one CENWAVE of each of the COS FUV gratings are illustrated in Figure 3.10. We note that the traces at subsequent LP are similar. The effect of the TRCECORR reduction step for the 1280 setting of the G140L grating is shown in Figure 3.11.

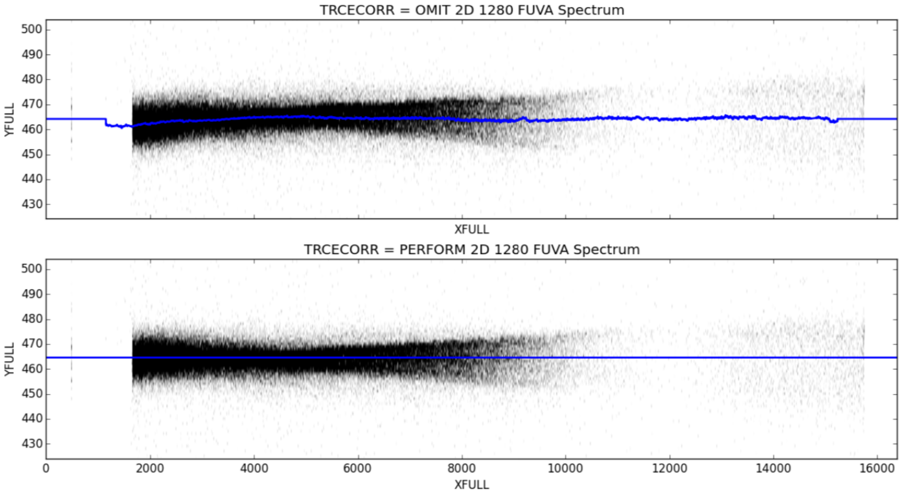
Figure 3.10: Spectral Trace Locations.



Spectral trace locations for one CENWAVE setting of each of the COS FUV gratings are shown. Trace locations for the other CENWAVE settings are very similar. The difference between the measured YCORR center as a function of XCORR (blue lines) and the median center (orange line) is tabulated in the TRACETAB reference file as a function of XCORR and it is this offset that is subtracted from the event locations during the TRCECORR step to straighten the science images.

Note: It is intended that the TRCECORR step be used together with the ALGNCORR step and the TWOZONE algorithm in X1DCORR. While it is possible to use TRCECORR with other combinations of settings, the resulting calibrated products may not be optimal. For more information on customized TWOZONE reductions see Section 3.6.3.

Figure 3.11: TRCECORR reduction step.



A spectral image for G140L FUVA is shown without (*above*) and with (*below*) the TRCECORR correction applied. The blue line shows the trace.

3.4.15 ALGNCORR: Alignment Correction

The ALGNCORR correction calculates and applies a constant shift to events to ensure that their centroid in the cross-dispersion direction is always the same as that of the reference profile in the PROFTAB reference file. The ALGNCORR step is intended to be used together with the TRCECORR step and with the TWOZONE algorithm in the X1DCORR step. Using ALGNCORR without TRCECORR or with the alternate BOXCAR algorithm of the X1DCORR step may produce unpredictable and poorly calibrated results.

- Reference file: `PROFTAB`, `TWOZXTAB`
- Input files: rawtag, `rawaccum`
- Header keywords updated: SP_LOC_A, SP_LOC_B (vertical location of spectrum in segments A, B), SP_ERR_A, SP_ERR_B (Poisson uncertainty in location of spectrum in segments A, B), and SP_OFF_A, SP_OFF_B (vertical shift applied to events in segments A, B)

When using the TWOZONE extraction algorithm, it is important to make sure that the 2D spectral image is accurately centered on the reference profile, since the inner region of the reference profile is significantly narrower than the full BOXCAR extraction region. This is accomplished in the ALGNCORR step by accurately measuring the flux-weighted centroid of the spectral profile in the cross-dispersion direction, and comparing that centroid to that of a reference 2D point-source spectral profile in the PROFTAB reference file. The offset between these two centroids is applied to the YFULL values of each event, in the sense of moving the centroid of the science spectrum so that it matches that of the reference profile.

The PROFTAB reference file is selected based on the values of the INSTRUME (must be COS) and LIFE_ADJ keywords (currently there are only reference files for Lifetime Position 3 and up). It is a FITS table containing a row for each valid combination of {SEGMENT, OPT_ELEM, APERTURE, CENWAVE}. SEGMENT can be FUVA or FUVB (the alignment correction is not performed for NUV data), OPT_ELEM can be any of the FUV gratings (G140L, G130M, G160M), APERTURE can be PSA or BOA (no alignment correction is performed on the WCA aperture), and CENWAVE can be any valid value for its corresponding grating.

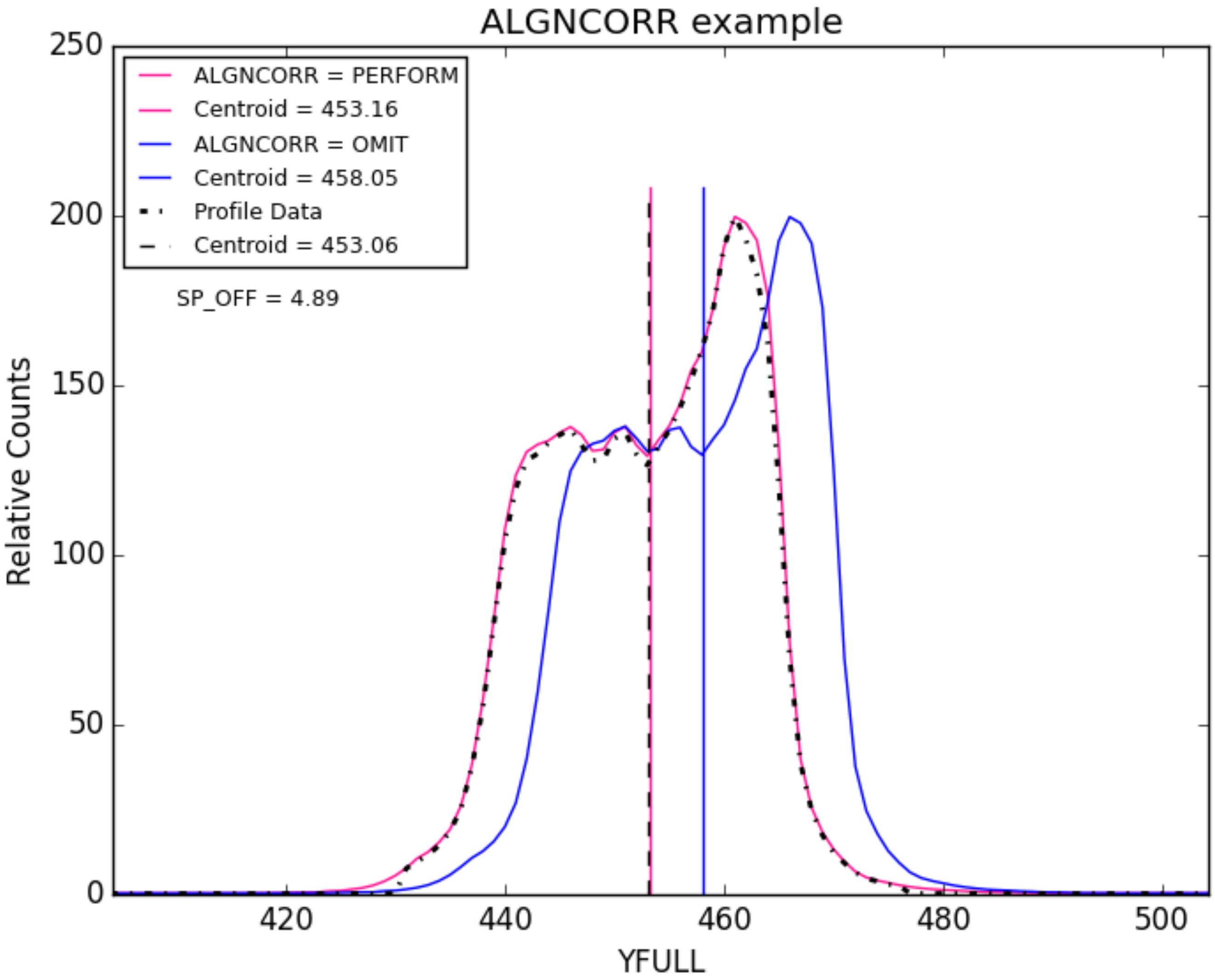
The calibration step works by first binning events in the `corrtag` file to create images of the counts in (XFULL, YFULL) space. A mask is then calculated and applied to the data to filter out data that would bias the centroid measurement. Such data include regions around strong airglow lines, plus any data from a column that contains a pixel whose DQ value contains any of the bits corresponding to the SDQFLAGS that are read from the data header. In **calcos** version 3.1, this was changed so that DQ values flagging gain sagged regions (DQ=8192) are not used to mask out columns, even when that DQ value is included in SDQFLAGS. This allows the alignment step to work properly even when very wide gain sagged regions exist in the wing of the profile.

The mean number of counts per pixel in the background regions for the valid columns is calculated, and this mean value is subtracted from each pixel in the 2D science spectrum. The location of the background regions is defined in the TWOZXTAB table. The 2D background-subtracted spectrum is then calculated along the dispersion direction (excluding columns with SDQFLAGS or containing airglow), and the flux-weighted centroid of the collapsed background-subtracted spectrum is calculated, along with its Poisson uncertainty. If this value differs from the current best value of the centroid by more than 0.005 pixels, the process is repeated with the background and science apertures shifted by the offset between the current and new centroid until convergence occurs.

Having calculated the flux-weighted centroid for the science data, the flux-weighted centroid of the reference profile is calculated using exactly the same mask for SDQFLAGS and the same background regions. The offset between the centroids of the science profile and the reference profile is applied to the YFULL values of every event that is inside the active area and outside the WCA aperture. The centroid, offset, and Poisson uncertainty are then written to the header of the science extension in keywords SP_LOC_A, SP_LOC_B, SP_OFF_A, SP_OFF_B, SP_ERR_A and SP_ERR_B.

The ALGNCORR correction is illustrated in Figure 3.12.

Figure 3.12: ALGNCORR reduction step.



This figure shows the collapsed cross dispersion profile of a G130M 1222 FUVA spectral image with (red line) and without (blue line) the correction from the ALGNCORR step applied. The broken line shows the collapsed cross dispersion reference profile to which the observed profile was aligned. The flux weight centroid values for each of the profiles are also marked. Notice how the reference profile and observed profile locations closely match after the alignment correction is applied.

3.4.16 DQICORR: Initialize Data Quality File

This module identifies pixels which are suspect in some respect and creates the DQ extension for the `flt` and `counts` images.

- Reference file: `BPIXTAB`, `GSAGTAB`, `SPOTTAB`, `TRACETAB`
- Input files: `rawtag`, `rawaccum`, `images`
- Header keywords updated: none

The DQICORR step assigns DQ values to affected events, and creates a DQ image extension that is used in the extraction step. It uses the Bad Pixel Table (BPIXTAB), Gain Sag Table (GSAGTAB) and Hotspot Table (SPOTTAB) to identify the regions that are relevant for the exposure using the following conditions:

- BPIXTAB: all regions included
- GSAGTAB: regions included if the DATE of the gain sag region is before the start of the exposure and the HVLEVEL of the gain sag extension matches that of the exposure
- SPOTTAB: regions included if the temporal extent of the spot (given by the START and STOP times of the spot region) overlaps the good time intervals of the exposure.

The COS data quality flags are discussed in Section 2.7.2 and are listed in Table 2.19. Figure 3.13 shows examples of the types of regions isolated by the DQ flags and the effect they can have on an extracted spectrum. DQICORR proceeds differently for `TIME-TAG` and `ACCUM` mode exposures, but the flags in the `flt` and `counts` images are created similarly in preparation for spectral extraction. Consequently, we describe each mode separately.

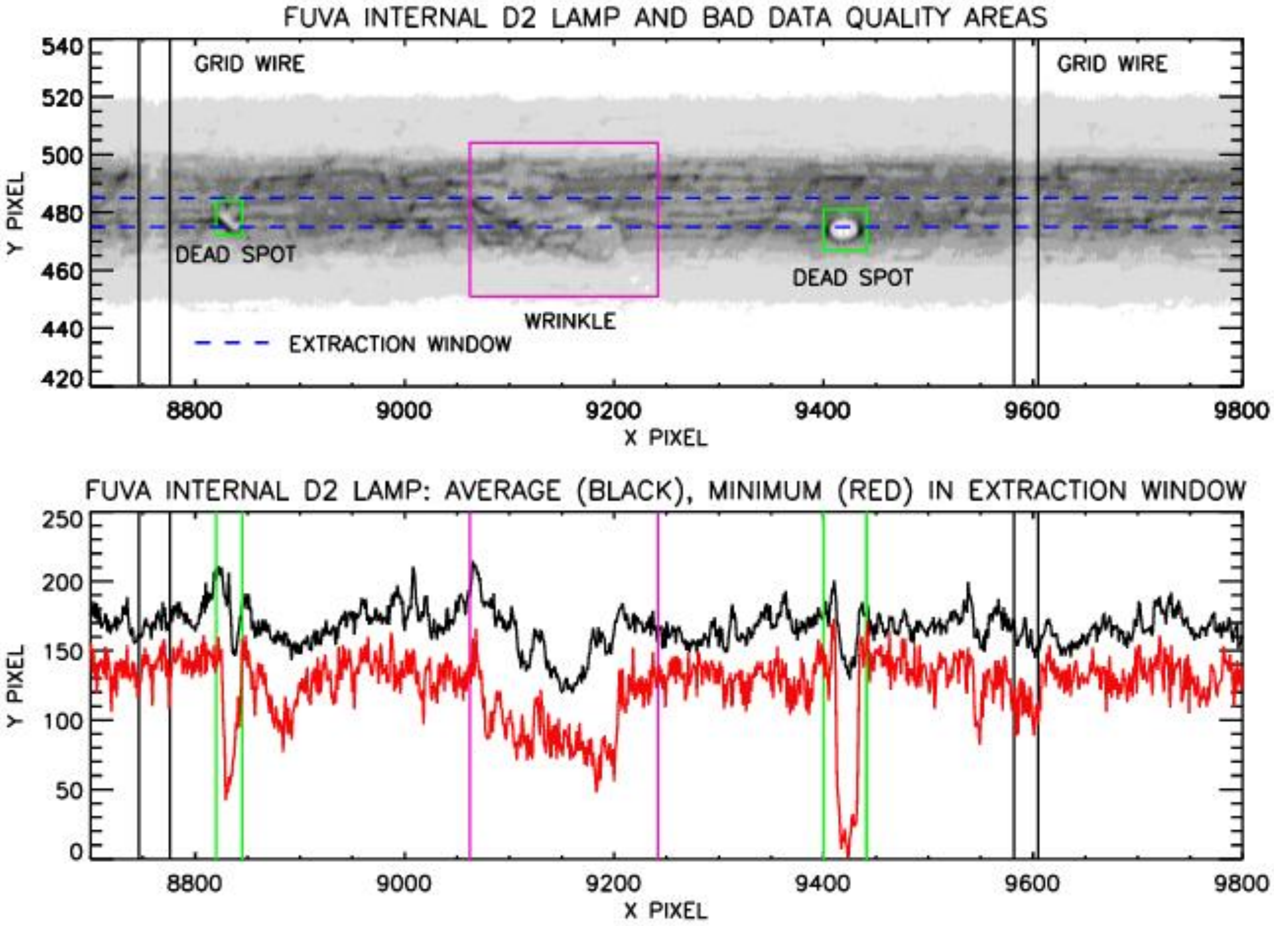
TIME-TAG:

DQICORR compares the XCORR, YCORR pixel location of each event in the `corrtag` file to the relevant rectangular regions as described above. The value in the DQ column for that event is then updated with the flags of all the regions (if any) that contain that pixel location. When the `flt` and `counts` images are generated from the `corrtag` file, photons which arrived during bad times or bursts are omitted from the image and ERR array. For FUV data, events whose PHAs were flagged as out of bounds are omitted as well. However, data with spatial DQ flags are retained at this stage. The third FITS extension of the `flt` and `counts` files is an array of data quality values generated directly from the `BPIXTAB`, `GSAGTAB`, and `SPOTTAB` reference files. If `DOPPCORR=PERFORM`, the included locations are Doppler-smeared and the flags from all neighboring pixels that contribute to the `flt` and `counts` image pixels are combined.

ACCUM:

For `ACCUM` exposures, the `rawaccum` image file will already have a third FITS extension of data quality values if any pixel had been flagged when constructing the raw image (the third extension does not exist for TIME-TAG data). The extension will be a null image if all initial data quality flags are zero. This is the case for NUV data, but not for FUV. For FUV `ACCUM` exposures, photons are collected for only part of the detector segment and an initial data quality array is created to mark the pixels outside those subimage boundaries (flag=128, outside active area), so there will always be data flagged as missing When **calcos** creates the `flt` and `counts` images, it first converts the `rawaccum` image to a pseudo-time-tag table. In this table, the DQ column is updated with the DQ flags from `BPIXTAB` just as for the TIME-TAG data. In addition, the third extension of the `flt` and `counts` files contains a Doppler-smeared version of the `BPIXTAB`, `GSAGTAB`, and `SPOTTAB` reference files, but it also includes the initial flag assignments in the `rawaccum` DQ extension.

Figure 3.13: The FUV Flat Field.



An FUV flat field obtained during ground testing illustrates the different kinds of blemishes and regions of lower sensitivity that occur. These regions are flagged in `BPIXTAB` according to the feature type, e.g., a "wrinkle" is a kind of detector flaw and grid wire is an example of a detector shadow. Dead Spots are also known as Low Response Regions.

The SDQFLAGS (Serious Data Quality FLAGS) header keyword value indicates which DQ values (see Table 2.19 for definitions) should be excluded from the statistical calculations. This value is the sum of the flag values to be excluded.

> ✓ *To select an alternative definition of SDQFLAGS, the user should modify the rawtag or rawaccum header and reprocess the file with calcos.*

3.4.17 STATFLAG: Report Simple Statistics

This module computes some statistical measures that provide general information about COS science observations.

- Reference file: `TWOZXTAB` or `XTRACTAB`, `BRFTAB`
- Input files: `flt`, `counts`, `x1d`, `lamptab`
- Header keywords updated: `NGOODPIX`, `GOODMEAN`, `GOODMAX`

STATFLAG enables the reporting of statistics for COS observations. STATFLAG is enabled by default for all science observations and operates on `x1d`, `counts`, and `flt` data products. STATFLAG is intended to provide a very basic statistical characterization of the events and locations on the detectors that are known to be good.

STATFLAG reports the following statistics:

- NGOODPIX: The number of good pixels or collapsed spectral columns. For the `counts` and `flt` images, this is the number of pixels in the spectral extraction or imaging region. For the `x1d` file, each 'Y' column in the spectral extraction region of the `flt` file is combined to produce the one-dimensional spectrum. The DQ of each column is the logical OR of the DQ flags of the individual pixels. Only collapsed spectral columns that pass the DQ conditions indicated by SDQFLAGS are considered good for purposes of calculating statistics.
- GOODMEAN: The mean of the good bins in counts per bin. For the `counts` and `flt` files, a bin is an individual pixel, while for `x1d` files, a bin is a collapsed spectral column.
- GOODMAX: The maximum of the good bins in the same units as the mean.

3.4.18 X1DCORR: Locate and Extract 1-D Spectrum

This module extracts a one-dimensional spectrum from the image of the spectrum on the detector.

- Reference files: `TWOZXTAB` or `XTRACTAB`, `WCPTAB`
- Input files `flt`, `counts`
- Header keywords updated: `SP_LOC_[ABC]`, `SP_OFF_[ABC]`, `SP_NOM_[ABC]`, `SP_SLP_[ABC]`, `SP_HGT_[ABC]`
- Creates `x1d` files

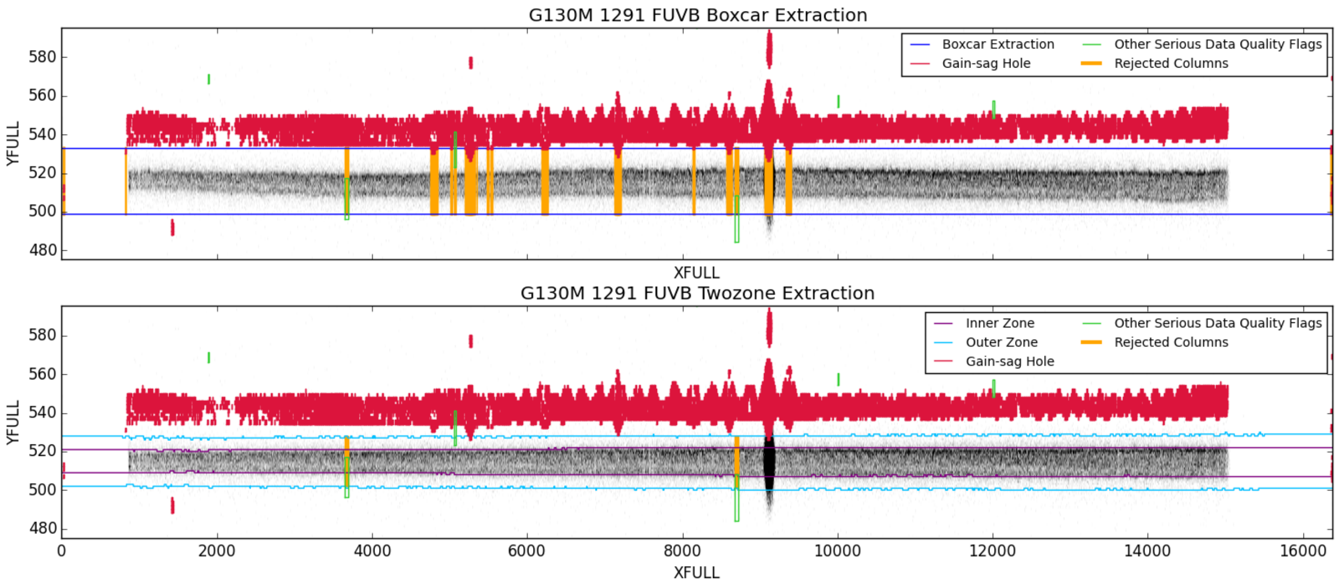
A 1-D spectrum and its error array are extracted from the `flt` and `counts` images by summing the counts in the cross-dispersion direction within a band centered on the spectrum. The data are not resampled in the dispersion direction. Wavelengths are assigned by evaluating a polynomial function (dispersion relation) in pixel coordinates. The background is subtracted (see BACKCORR; Section 3.4.19) to derive the net count rate, and the absolute flux is computed from the net count rate (see FLUXCORR; Section 3.4.20). With **calcos** version 3.0 or later, X1DCORR added support for the new TWOZONE extraction algorithm in addition to the older BOXCAR algorithm. The fundamental differences between these algorithms are the way the regions are chosen for the extraction of the data and the method of combining data quality flags in these regions. These differences are described below. Note, however, that the TWOZONE algorithm is designed to be used with the new TRCECORR and ALGNCORR steps, while for the BOXCAR, these steps should always be set to OMIT.

CASE WITH XTRCTALG=BOXCAR

When using the BOXCAR algorithm, the parameters controlling the extraction are taken from the XTRACTAB reference file described in Section 3.7.12 and Table 3.9.

The spectral extraction of a source is performed by collapsing the data within a parallelogram of height HEIGHT that is centered on a line whose slope and intercept are given by SLOPE and B_SPEC respectively. Similarly, two background spectra are determined by collapsing the data within parallelograms of height B_HGT1 and B_HGT2 centered on the lines defined by SLOPE and B_BKG1, and SLOPE and BKG2, respectively. The background spectra are then smoothed in the dispersion direction by a boxcar of width BWIDTH. These are then scaled and subtracted from the source spectrum.

Figure 3.14: FUV Extraction Region.



A comparison of the BOXCAR (upper panel) and TWOZONE (lower panel) extraction algorithms is shown for the FUVB segment of a G130M 1291 observation done at LP3. The observed events are shown using a grey-scale image. Regions suffering more than 5% local sensitivity loss due to gain sag are marked in red, while other serious data quality flags in the extraction region are outlined with green boxes. The yellow bars show the wavelength bins that are rejected as bad by each of the algorithms. Note that since the TWOZONE algorithm only rejects wavelength bins when a bad region lies within the inner zone, it rejects far less of the spectrum than does the BOXCAR algorithm.

CASE WITH XTRCTALG=TWOZONE

For the TWOZONE algorithm, the parameters controlling the extraction are taken from the TWOZXTAB reference file. The appropriate row from this table is selected based on the detector SEGMENT, OPT_ELEM, CENWAVE, and APERTURE.

The TWOZONE algorithm assumes that the spectrum has been straightened using the TRCECORR step of **calcos** and aligned with the appropriate reference profile by the ALGNCORR step.

When performing the TWOZONE extraction, **calcos** divides the spectral extraction region into two parts. One region, referred to as the "inner zone," defines the vertical extent of the region over which data quality flags are considered for inclusion in output data quality array, while the other, referred to as the "outer zone," defines the region over which events are summed to produce the GROSS flux and other vectors calculated from the `flt` and counts `images` (see Figure 3.14). Note that, despite its name, the outer zone will include the entire region included in the inner zone. The upper and lower boundaries of these zones vary as a function of position in the dispersion direction.

After the TWOZONE algorithm had been developed and implemented for the FUV, a region of transient elevated counts appeared (known as a "hotspot"). These events could seriously affect the extracted FUV flux if they are included whether they are in the inner or outer zone, so in **calcos** version 3.1 code was added to ensure that columns with DQ flags that match those in the SDQOUTER keyword are not included in any summed spectra.

The locations of these zone boundaries are calculated using the quantities HEIGHT, B_SPEC, LOWER_OUTER, UPPER_OUTER, LOWER_INNER and LOWER_OUTER taken from the TWOZXTAB reference file, together with the reference profile selected from the appropriate row of the PROFTAB reference file, which is selected based on the exposure SEGMENT, OPT_ELEM, CENWAVE and APERTURE. The reference profile, selected from the PROFTAB is truncated to HEIGHT number of rows centered at location B_SPEC. In each column of the profile, a cumulative sum of the profile is calculated and normalized to unity, so that it runs from 0 to 1 over the height of the profile. This cumulative sum is interpolated to find the locations at which the cumulative sum crosses the LOWER_OUTER, UPPER_OUTER, LOWER_INNER, and LOWER_OUTER boundaries. These interpolated boundaries are rounded outwards to integer pixel values, so the UPPER values are rounded up and the LOWER values rounded down. The final integer boundary values are then placed in the vectors Y_LOWER_OUTER, Y_UPPER_OUTER, Y_LOWER_INNER, and Y_LOWER_INNER, and these are be used to define the region of the `flt` and `counts` images over which the GROSS and NET are summed and over which the data quality values are combined.

The background region selection for the TWOZONE algorithm is done in a similar way as for the BOXCAR algorithm. The background per pixel is calculated for each column by dividing the total background counts by the number of background pixels. This value is subtracted from the intensity in every pixel in the science aperture.

Output arrays

This section provides the details of the spectral extraction process and the construction of the arrays that populate the `x1d` files. Table 3.1 lists these arrays along with others that are used to calculate them. Names listed in capital letters in this table correspond to columns in the `x1d.fits` files. Names given in lower case refer to temporary quantities used in the calculations that are not included in the output files, but which are used in the definition of some of the included quantities. The summed `x1dsum[n]` files are described in Section 3.4.22.

Table 3.1: Variables used in 1-D Spectral Extraction.

Variable	Description
SEGMENT	A string array listing the segments/stripes contained in the file
NELEM	An integer listing the number of elements in the extracted arrays
EXPTIME	The exposure times used for each segment, in double-precision format
e[i]	Effective count rate, extracted from `flt` file
GROSS[i]	Gross count rate, extracted from `counts` file
GCOUNTS[i]	Gross counts
BACKGROUND[i]	Smoothed background count rate, extracted from `counts` file
eps[i]	e[i] /GROSS[i]
NET[i]	Net count rate = eps[i] (GROSS[i] – BACKGROUND[i])
ERROR[i]	Upper bound of internal error estimate

ERROR_LOWER[i]	Lower bound of internal error estimate
VARIANCE_FLAT[i]	Term used for calculating the internal error due to the flat-field error.
VARIANCE_COUNTS[i]	Term used for calculating the internal error due to the gross counts.
VARIANCE_BKG[i]	Term used for calculating the internal error due to the background counts.
FLUX[i]	Calibrated flux
WAVELENGTH[i]	Wavelength scale in Angstroms.
DQ_WGT[i]	Weights array
DQ	Bitwise OR of the DQ in the extraction region
snr_ff	The value of keyword SNR_FF from the flat-field reference image
extr_height	The number of pixels in the cross-dispersion direction that are added together for each pixel of the spectrum
bkg_extr_heigh	The number of pixels in the cross-dispersion direction in each of the two background regions
bkg_smooth	The number of pixels in the dispersion direction for boxcar-smoothing the background data
bkg_norm	Float (extr_height) / (2.0*float (bkg_extr_height))
calcos 3.0 added the following new variables	
DQ_ALL[i]	Data quality flags over the full extraction region
NUM_EXTRACT_ROWS[i]	Number of extracted rows
ACTUAL_EE[i]	Actual energy enclosed between outer zone boundaries
Y_LOWER_OUTER[i]	Index of lower outer extraction zone boundary
Y_LOWER_INNER[i]	Index of lower inner extraction aperture boundary
Y_UPPER_INNER[i]	Index of upper inner extraction zone boundary
Y_UPPER_OUTER[i]	Index of upper outer extraction zone boundary
BACKGROUND_PER_PIXEL [i]	Average background per pixel
lower_outer_value[i]	Fraction of flux enclosed at and below row Y_LOWER_OUTER
lower_inner_value[i]	Fraction of flux enclosed at and below row Y_LOWER_INNER
upper_inner_value[i]	Fraction of flux enclosed at and below row Y_UPPER_INNER
upper_outer_value[i]	Fraction of flux enclosed at and below row Y_UPPER_OUTER

Note: Variables beginning with a capital letter are saved in the output `x1d` file. An "[i]" represents array element i in the dispersion direction.

The columns in the `x1d` files are now described in more detail.

SEGMENT: A string array listing the segments/stripes contained in the file.

NELEM: An integer listing the number of elements in the extracted arrays.

EXPTIME: The exposure times used for each segment, (which can differ for FUV data), in double-precision format.

GROSS: The GROSS count rate spectrum is obtained from the `counts` file by summing over the extraction region. While, as described earlier in this section, the definition of the extraction region differs between the BOXCAR and TWOZONE algorithm, in each case the sum over each cross dispersion column runs from the Y_LOWER_OUTER to Y_UPPER_OUTER location listed in the `x1d` output table. These sums always include the endpoints.

GCOUNTS: This is simply the number of gross counts, or `GROSS` times `EXPTIME`.

BACKGROUND: Two background regions are sampled on the `counts` array to obtain a mean background count rate spectrum. For FUV data, these are above and below the spectrum (see Figure 3.15). For NUV data they are above stripe C and below stripe A (Figure 3.16). The background regions are extracted in the same way as the spectrum. The values in the two background regions are added, boxcar-smoothed in the dispersion direction, and scaled by the sizes of their extraction regions before being subtracted from the science spectrum. Details of the background extractions are given in Section 3.4.19.

NET: The NET spectrum is the difference between the GROSS spectrum and a properly scaled BACKGROUND spectrum multiplied by an array which accounts for flat-field and dead-time effects. This array is eps[i] = e[i] /GROSS[i], where e[i] is an element in an array extracted from the `flt` file in exactly the same way as the GROSS spectrum is extracted from the `counts` file. Consequently, this factor corrects the NET spectrum for flat-field and dead-time effects. When the TWOZONE algorithm is used, an additional correction factor of 1 /ACTUAL_EE is also applied to account for the actual enclosed energy fraction in each column.

ERROR and ERROR_LOWER: The ERROR and ERROR_LOWER arrays, which represent the upper and lower 1-sigma flux errors, are calculated via error propagation of the net count rate equation:

$$N_i = \varepsilon_i(GC_i - BK_i)$$

where N_i is the net count rate, e_i is the inefficiency factor, GC_i is the gross count rate, and BK_i is the background count rate per wavelength bin (i). Propagating the various error terms leads to the final error equations:

$$\sigma_{u,l;N_i} = \frac{1}{t}\sqrt{\left(\frac{N_i t}{E_j SNR_{ff}}\right)^2 + [\varepsilon_i f_{u,l}(GC_i t)]^2 + \left[-\varepsilon_i f_{u,l}\left(\frac{\Delta n_1}{\Delta w \Delta n_2} BK_i t\right)\right]^2}$$

where $f_{u,l}$ is the upper/lower "Frequentist-Confidence" Poisson error estimates from the astropy.stats. poisson_conf_interval package. The raw ERROR and ERROR_LOWER arrays involve elements from both the flt and counts files. The ERROR and ERROR_LOWER arrays contained in the _x1d files differ from **the flt and counts files** in the sense that **the _x1d files** have the absolute flux calibration applied (i.e., the net count rate errors are converted into flux errors; see Section 3.4.20).

VARIANCE_FLAT: The first term from the ERROR and ERROR_LOWER equations that gives the contribution to the total flux error provided by the flat-field uncertainty. The value in the VARIANCE_FLAT field is calculated as:

$$VARIANCE\_FLAT_i \equiv \left(\frac{N_i t}{E_j SNR_{ff}}\right)^2$$

where N_i is the net count rate and t is the exposure time. The product of the number of columns summed (E_j) and the S/N of the flat-field (SNR_{ff}) can be determined from simple arithmetic.

VARIANCE_COUNTS: The second term from the ERROR and ERROR_LOWER equations that gives the contribution to the total flux error provided by the gross count uncertainty. The value in the VARIANCE_COUNTS field is calculated as:

$$VARIANCE\_COUNTS_i \equiv \varepsilon_i^2 GC_i t$$

where e_i is the inefficiency factor, GC_i is the gross count rate, and t is the exposure time.

VARIANCE_BKG: The third term from the ERROR and ERROR_LOWER equations that gives the contribution to the total flux error provided by the smoothed background uncertainty. The value in the VARIANCE_BKG field is calculated as:

$$VARIANCE\_BKG_i \equiv \varepsilon_i^2 \left(\frac{\Delta n_1}{\Delta w \Delta n_2} BK_i t\right)$$

where e_i is the inefficiency factor, BK_i is the background count rate, and t is the exposure time. The additional term that includes the height and width of the background smoothing box can be solved for with simple arithmetic.

FLUX: The FLUX array in the `x1d` file is the NET spectrum corrected by the appropriate time dependent sensitivity curve. The details of this process are discussed in Section 3.4.20.

WAVELENGTH: As part of the spectral extraction, **calcos** assigns wavelengths to pixels in the extracted spectra using dispersion coefficients from the reference table `DISPTAB`. Wavelengths correspond to the center of each pixel. For each segment or stripe, grating, central wavelength, and aperture, the `DISPTAB` table contains the dispersion solution with respect to the template spectral lamp table that was used in the WAVECORR step. The dispersion solution has the following form:

$$\mathrm{WAVELENGTH[i]} = \mathrm{A}_0 + \mathrm{A}_1\mathrm{x[i]} + \mathrm{A}_2\mathrm{x[i]}^2 + \mathrm{A}_3\mathrm{x[i]}^3$$

where WAVELENGTH[i] is the wavelength in Angstroms, x[i] is the pixel coordinate in the dispersion direction, and A_i are the dispersion coefficients.

DQ_WGT: The DQ_WGT array has one point for each extracted point in the spectrum. It is 0 or 1 depending on whether the DQ for a given point is allowed according to the header keyword, SDQFLAGS. The SDQFLAGS value depends on the configuration of the instrument. These SDQFLAGS values set the DQ_WGT to 0 for events that are near the edge of the detector, dead spots, hot spots or outside the subarray (see Table 2.19). Otherwise, DQ_WGT = 1. The DQ_WGT array is used to construct the `x1dsum` file discussed in Section 3.4.22.

DQ: The DQ array in the `x1d` file is the bitwise OR of the members of the DQ array, contained in the third FITS extension of the `counts` file. For the BOXCAR extraction, this includes all of the points in the `counts` image that contribute to an element of the GROSS spectrum. Consequently, if anything is flagged within the extraction region, it is reflected in the `x1d` DQ array. For the TWOZONE extraction, the DQ flags in each column are only combined from Y_LOWER_INNER to Y_LOWER_OUTER. This causes DQ flags included only in the outer zone to be ignored, unless they are in the DQ value SDQOUTER from the primary header, in which case they are included in the value of DQ.

DQ_ALL: The DQ_ALL array gives the DQ value for the full outer zone extraction region in the case of TWOZONE extraction, otherwise it gives the same value as in the DQ array.

NUM_EXTRACT_ROWS: This gives the height of the extraction aperture as a function of column number. For BOXCAR extraction, this will be a constant equal to the height of the extraction region. For TWOZONE extraction, it will vary with column number/wavelength.

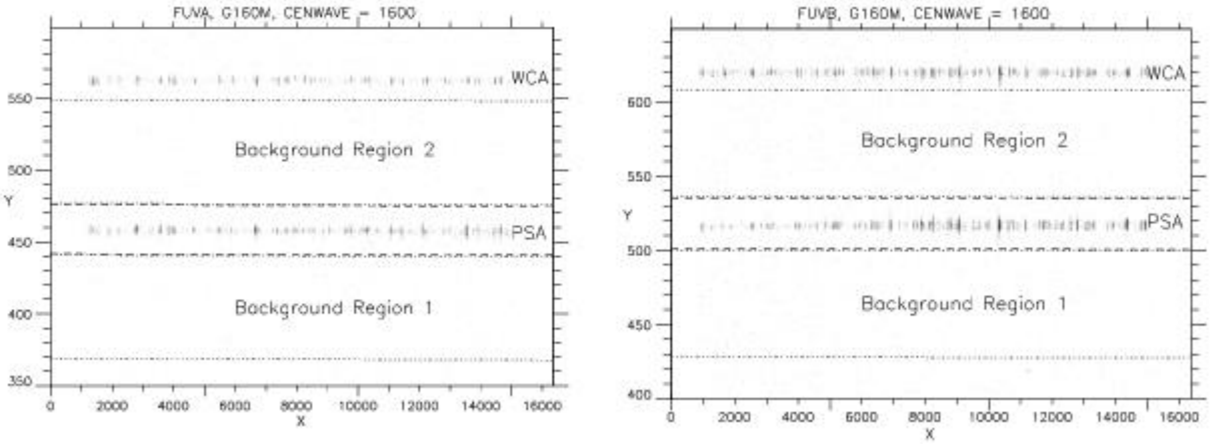
ACTUAL_EE: With BOXCAR extraction, this will be 1.0. With TWOZONE extraction, this will vary from column to column. Extraction is done on whole pixels, so while the outer boundaries are supposed to enclose the fraction of the flux specified by the difference between the LOWER_OUTER and UPPER_OUTER values taken from the TWOZXTAB, in practice the height equals Y_UPPER_OUTER-Y_LOWER_OUTER+1. The actual fraction of the encircled energy that is enclosed is reported in the ACTUAL_EE variable.

Y_LOWER_OUTER, Y_LOWER_INNER, Y_UPPER_INNER, Y_UPPER_OUTER: These variables give the row number of the boundaries defining the inner and outer regions. In the case of spectra extracted using the BOXCAR algorithm, the inner and outer indices are the same, and the indices follow the slant of the extraction aperture.

BACKGROUND_PER_PIXEL: This gives the smoothed background per pixel, i.e., the total background in the extraction aperture divided by the number of rows included in the background aperture for each column.

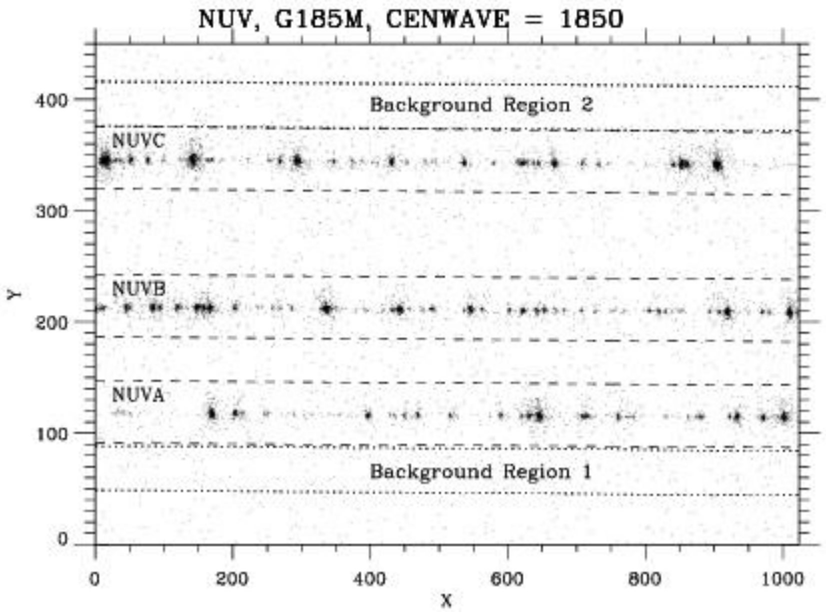
Finally, the **SP_* KEYWORDS** (listed in Table 2.15) provide useful information on the location of the spectrum in the cross-dispersion direction and the location where the spectrum is extracted. When XTRCALG=TWOZONE, ALGNCORR is set to PERFORM and the AGLNCORR task is used to set these keywords. See the description of the ALGNCORR section for a discussion of how these keywords are populated in that case. When ALGNCORR=OMIT, as is normally the case for XTRCTALG=BOXCAR, these values are set by the X1DCORR task as follows. The actual location of the spectrum is found from the flt file through a two step process. First, the image of the active area is collapsed along the dispersion direction to produce a mean cross-dispersion profile. Second, a quadratic is fit to a full-width-half-maximum-pixel region (with a minimum of 5 pixels) centered on the maximum of the profile. The difference between this value and the expected location, SP_NOM_A[B], is given as SP_OFF_A[B]. The actual location where the spectrum is extracted is given by `SP_LOC_A[B]`. For BOXCAR pipeline extractions, `SP_LOC_A[B]` = `SP_NOM_A[B]`, and `SP_OFF_A[B]` is listed for informational purposes only. However, it is possible to override these values and extract a spectrum at the `SP_OFF` location or any other by using the stand alone version of **x1dcorr** discussed in Section 5.1.1.

Figure 3.15: FUV Background Extraction Regions.



Portions of the undistorted images of the FUV detector segments (compare to Figure 2.2) illustrating the regions used to extract the spectrum and define the background. The dashed lines indicate the spectral extraction window, and the dotted lines define the background extraction region (BOXCAR vs. TWOZONE).

Figure 3.16: NUV Background Extraction Regions.



Portion of the NUV detector showing spectral extraction regions used for the three non-contiguous PSA spectra (dashed lines) and for the background (dotted lines).

3.4.19 BACKCORR: 1D Spectral Background Subtraction

This module determines the background contribution to the extracted spectrum and subtracts it.

- Reference file: `XTRACTAB` or `TWOZXTAB`
- Input files: `flt`, `counts`
- Header keywords updated: none

The BACKCORR module computes the number of counts in the background regions, scales them by the ratio of sizes of the spectral extraction region to background regions, and then subtracts that value from the extracted spectrum at each wavelength. There are two background regions defined. For FUV data at LP3 and LP4, there is one above and one below the object spectrum (see Figure 3.15). For LP5 and later, please look at COS ISRs. For the NUV spectra, the two regions are above and below the three stripes (see Figure 3.16). Each background region is a parallelogram with the same slope used to define the object extraction region, but with different y-intercepts. The parameters of the background extraction region in the FUV are:

- HEIGHT: the full height (along the cross-dispersion) of the object extraction region in pixels
- BHEIGHT (for TWOZONE extraction) *or* B_HGT1 and B_HGT2 (for BOXCAR extraction): the full height (along the cross-dispersion) of the background regions in pixels. The upper and lower edges of each background region are defined as ±(BHEIGHT−1)/2 pixels from the line tracing the center of each region
- BWIDTH: the full width (along the dispersion) of the box-car average performed on the background
- B_BKG1: y-intercept of first background region
- B_BKG2: y-intercept of second background region
- SLOPE: the slope of the line tracing the centers of both the spectrum and background regions

The centers of background regions 1 and 2 in the cross-dispersion (Y) direction follow a linear function in the dispersion (X) direction according to the function:

$$Y = mX + b$$

where m is the slope of the background (keyword SLOPE), and b is the Y-intercept of the background region (B_BKG1 or B_BKG2). At the i-th pixel along the dispersion direction (X) the background is computed by first summing all of the counts in pixels in the cross-dispersion within ±(BHEIGHT/2) of the central Y pixel of the background box. Data in flagged regions, as defined by the DQ flags, are ignored, and counts that occur during bad time intervals or that have out-of-bounds PHAs never make it to the `counts` file. If a flagged region does not cover the full height of a background region, the count rate in the non-flagged region will be scaled up to account for the omitted region.

Once the counts are summed for all X pixels, the result is averaged over ±BWIDTH/2 pixels along the dispersion direction. This gives a local average background (with known anomalous pixels such as dead spots or strong hot spots excluded). Both background regions are computed in this way, and then they are summed and divided by two to yield an average background rate. This average is then scaled to the number of pixels in the object extraction box by multiplying it by the factor "HEIGHT/(2*BHEIGHT)." The result is the background count rate BK[i] in Table 3.1, which is written to the BACKGROUND column in the `x1d` file. The background-subtracted count rate (corrected for flat field and dead time) is written to the NET column in the `x1d` table.

3.4.20 FLUXCORR/TDSCORR: Conversion to Flux

This module converts extracted spectrum into physical units, and allows for time dependencies in the conversion.

- Reference files: `FLUXTAB`, `TDSTAB`
- Input file: `x1d`
- Header keywords updated: none

If `FLUXCORR=PERFORM`, FLUXCORR divides the NET and ERROR columns by the appropriate sensitivity curve read from the `FLUXTAB` reference table, which converts them to flux units (erg cm^{-2} s^{-1} $Å^{-1}$). The NET divided by the sensitivity is written to the FLUX column of the `x1d` file, while the ERROR column is modified in-place.

The sensitivity curves read from the reference files are interpolated to the observed wavelengths before division. The flux calibration is only appropriate for point sources and has an option to accommodate time-dependent sensitivity corrections.

If `TDSCORR=PERFORM`, then the module TDSCORR will correct for temporal changes in the instrumental sensitivity relative to the reference time given by `REF_TIME` keyword in the FITS header of `TDSTAB`. `TDSTAB` provides the slopes and intercepts needed to construct a *relative* sensitivity curve. The curve for the epoch of the observation is determined by piecewise linear interpolation in time using the slopes and intercepts closest to the time of the observation. The sensitivity may be discontinuous at an endpoint of a time interval. Different piecewise linear functions may be specified for each of the wavelengths listed in the table. This process results in a relative sensitivity at the epoch of the observation, at the wavelengths given in the reference table. Interpolation between these wavelengths to the observed wavelength array is also accomplished by piecewise linear interpolation.

3.4.21 HELCORR: Correction to Heliocentric Reference Frame

This module converts the observed wavelengths to Heliocentric wavelengths.

- Reference file: none
- Input files: `rawtag`, `x1d`

- Header keywords updated: `V_HELIO`

In addition to the Doppler smearing from *HST* orbital motion, the photons acquired during an observation are also Doppler shifted due to the orbital motion of the Earth around the Sun (V ~ 29.8 km/s). The sign and magnitude of the Doppler shift depend on the time of the observation as well as the coordinates of the target (i.e., the position of the target relative to the orbital plane of the Earth).

The HELCORR module in **calcos** transforms wavelengths of a spectrum to the heliocentric reference frame. It is applied to the extracted 1D spectrum during the operation of X1DCORR, by utilizing the keyword `V_HELIO`, which is the contribution of the Earth's velocity around the Sun to the radial velocity of the target (increasing distance is positive), in km/s. It is computed by **calcos** and written to the science data header of the output `corrtag` file before spectral extraction is performed.

The shift at each wavelength is:

$$\lambda_{\text{Helio}} = \lambda_{\text{Obs}}[1-(V_{\text{Helio}}/c)]$$

where λ_{Helio} is the corrected wavelength (Å), c is the speed of light in km/s and λ_{Obs} is the wavelength before the Heliocentric correction.

The velocity vector of the Earth is computed in the J2000 equatorial coordinate system, using derivatives of low precision formulae for the Sun's coordinates in the Astronomical Almanac. The algorithm does not include Earth-Moon motion, Sun-barycenter motion, or the Earth-Sun light-time correction.

3.4.22 Finalization (making the `x1dsum` files)

Once the processing is complete, an `x1d` file is written for each spectroscopic exposure. This file includes spectra from both segments A and B for the FUV detector, and from all three stripes for the NUV detector. In addition, one or more `x1dsum` files are created. This is done even if only one spectrum was obtained.

The `x1dsum` files differ from the `x1d` files in one important respect. When an `x1dsum` file is created the DQ_WGT array (Section 2.7.2) is used to determine whether a point is good or bad. When only a single file contributes to the `x1dsum` file, if DQ_WGT = 0 for a pixel, then the counts, net and flux arrays for that point are set to zero. If the `x1dsum` or `x1dsum<n>` (for FP-POS observations) includes several `x1d` files (see Section 2.4.3), then, for each point in the spectrum, only those files with a DQ_WGT = 1 at that point are included (weighted by the individual exposure times), and the DQ_WGT array in the `x1dsum` file is updated to reflect the number of individual spectra which contributed to the point. If the updated value of DQ_WGT for a particular point is 0, then the value of the spectrum at that point is set to 0 in the `x1dsum` file.

3.5 Descriptions of Imaging Calibration Steps

The processing of NUV imaging data is depicted in Figure 3.6. It is an abbreviated version of the pipeline that only involves those steps which identify bad data and linearize the initial counts. No absolute flux calibration is performed and no background is identified or subtracted.

The final data products for NUV imaging data are the `flt` and `fltsum` files described in Section 2.4.2. Like the `x1dsum` files, an `fltsum` file is created even if only one exposure is processed. However, since no shifting is performed for imaging observations (see Figure 3.6), the `fltsum` file is a simple exposure-time-weighted mean of the individual `flt` files (and it is identical to the `flt` file if only one exposure contributed to it). The DQ flags image of the `fltsum` file and, for that matter, all of the individual `flt` images, are identical. This is because the only data which make it into an `flt` or `counts` image are free of temporal or event flags (see Section 2.7). Consequently, in the absence of shifting, all of their spatial flags should be identical.

Although imaging data are not flux calibrated, a crude calibration can be performed using the total count rate from the `flt` file and one of the two keywords provided in the count rate extension header. The header keyword `PHOTFLAM` is appropriate for a source spectrum that is flat and featureless across the MAMA detector band when measured in units of power/area/wavelength and `PHOTFNU` is appropriate for a source that is flat in power/area/Hz units. The values provided for the `PHOTFLAM` and `PHOTFNU` keywords depend on the specific combination of mirrors and apertures used in the observation.

3.6 Customizing COS Data Calibration

Sometimes the pipeline calibration performed shortly after the data were obtained from the telescope is not the best possible calibration for your science program. There are a number of reasons why it may be desirable to recalibrate your data. The most likely reasons include:

- More appropriate reference files have become available since the data were originally obtained.
- Some steps need to be repeated with different input parameters. For example, you may wish to re-perform the 1-D spectral extraction with a smaller BOXCAR height in order to minimize the background, or you may wish to cut a `TIME-TAG` exposure into sub-exposures, in order to study time variability.

In the first case, we recommend simply re-requesting the data from the archive, providing a reduction produced with the latest reference files. In the second case, to tailor the calibration to your individual preferences, it may be beneficial to run **calcos** yourself on your local machine, or to use tasks that improve the reference files or allow customized treatment of the data. **Calcos** is imported and executed within **Python**.

⚠ ***Be sure you are using the latest versions of the calcos, COS calibration files, and raw data files (which list the latest reference files in their headers). Calcos release information can be found at http://www.stsci.edu/hst/instrumentation/cos/documentation/calcos-release-notes/.***

Calcos contains provisions for recalibrating raw data. Users can specify the pipeline processing steps to be performed and select the associated reference files. However, **calcos** was not designed to run its various modules independently, i.e., it is not re-entrant. The pipeline flow is modified by setting calibration switches or reference file names and then rerunning the entire pipeline. The calibration switches in the headers of the calibrated data files will reflect the operations performed on the calibrated data and the reference files used.

✓ Users are encouraged to use the Jupyter notebooks for COS. A list of those available is given in 5.1 Data Reduction and Analysis Applications.

3.6.1 Mechanics of Tailored Recalibration

If you chose to recalibrate your COS data on your local machine, there is a certain amount of set up required for **calcos** to run properly. The operations mentioned in the checklist below will be described in detail in the following subsections:

- Set up a directory structure for the required reference files.
- Determine which reference files are needed and retrieve them from the Archive.
- Set the environment variable `lref` to point to your reference file directory.
- Update the input data file headers (including reference file names).
- Set the calibration switches in the headers of the raw data files to perform the needed steps. The default calibration switches are listed in Table 2.16 and Table 2.17.
- Update the input association files if changing files to be included.
- Run **calcos**.

Set up the Directory Structure for Running calcos

Before running **calcos**, you will need to define an environment variable to indicate the location of the directory containing the needed calibration reference files. The names of the calibration files are preceded with the logical path name "`lref$`" in the COS science headers. You will need to define an environment variable from the host command line (see below) that is appropriate to your host machine. For Unix/Linux /Mac systems, the appropriate command for a `.bashrc` file and the directory `"/data/vega3/cos/cal_ref/," for example,` would be:

```
% export lref=/data/vega3/cos/cal_ref/
```

Note that an alternative to using the `lref$` variable is specifying the full pathnames to the reference files in the science headers, however there is a limit in the length of these pathnames.

When running calcos *or any of its modules, you must define environment variables* (*such as* `lref$`).

Retrieve Reference Files

To recalibrate your data, you will need to retrieve the reference files used by the different calibration steps to be performed. The names of the reference files to be used during calibration must be specified in the primary header of the input files, under the section "CALIBRATION REFERENCE FILES." Note that the data headers will be populated already with the names of the reference files used during pipeline calibration at STScI.

The COS reference files are all in FITS format, and can be in either IMAGE or BINTABLE extensions. The names of these files along with descriptions of their contents are given in Section 3.7. The rootname of a reference file is based on the time that the file was delivered to the Calibration Reference Data System (CRDS).

Chapter 1 of the *Introduction to HST Data Handbooks* describes how to retrieve data and reference files via the World Wide Web. To retrieve the best reference files via MAST (generally meaning the most recent reference files), check "Best Reference Files" in the "Reference Files" section of the Retrieval Options form. The reference files can also be downloaded from CRDS at https://hst-crds.stsci.edu/.

Edit the Calibration Header Keywords

To edit file headers in preparation for recalibration, use the **astropy.io.fits** convenience function **setval()**. The **setval()** function takes several input parameters: the name(s) of the raw data files to be edited, the header field to be edited, and the new value of the header field. It can be used to change the values of any calibration switches, reference files or tables to the values you wish to use for recalibrating your data. To edit the calibration keyword values:

1. Run the **setval()** function in the **astropy.io.fits** module. This can be done on a single file, or in a for-loop to update several files. For example, you could change the flat reference file in the following way:

   ```
   > from astropy.io import fits
   > fits.setval('filename_rawtag_a.fits', 'flatfile', \
         value='lref$n9n201821_flat.fits')
   ```

2. To update several files at once, use the **glob** module in the **glob** package to select all the raw files using wildcards, and then use a for-loop as shown below:

```
> from astropy.io import fits
> import glob
> rawfiles = glob.glob('*raw*.fits')
> for myfile in rawfiles:
>     fits.setval(myfile, 'flatfile', \
      value='lref$n9n201821_flat.fits', ext=0)
```

Similarly, to turn off the FUV burst calibration switch use the command:

```
> from astropy.io import fits
> fits.setval('filename_rawtag_a.fits', 'brstcorr', \
      value='OMIT', ext=0)
```

Edit the Input Association File

Users may find it necessary to edit the input association file for **calcos**. Reasons for editing an association file might include the use of a different wavecal or to remove a compromised exposure from an association. For this option, the full file name (but not the directory) must be given, and the case must be correct. One way to update an association file is to use the **astropy.io.fits** and **astropy.table.Table modules**. For example, use the **Table.read()** function to open and look at the contents of the association table ldel05050_asn.fits.

```
> from astropy.table import Table
> t = Table.read('ldel05050_asn.fits')
> print(t)
<Table length=5>
 MEMNAME  MEMTYPE MEMPRSNT
  str14    str14    bool
--------- ------- --------
LDEL05JYQ  EXP-FP     True
LDEL05K0Q  EXP-FP     True
LDEL05K2Q  EXP-FP     True
LDEL05K4Q  EXP-FP     True
LDEL05050 PROD-FP     True
```

To quickly see basic exposure information for a list of exposures, such as those in the association, use the **glob** package, a for-loop, and the **getheader()** convenience function in the **astropy.io.fits** module:

```
> from astropy.io import fits
> import glob
> raw_files = glob.glob('ldel05*raw*.fits')
> for myfile in raw_files:
>     hdr0 = fits.getheader(myfile)
>     print(hdr0['filename'], \
>     hdr0['detector'], hdr0['aperture'], \
>     hdr0['opt_elem'], hdr0['cenwave'], \
>     hdr0['obsmode'], hdr0['fppos'])

ldel05jyq_rawtag_a.fits FUV PSA G160M 1611 TIME-TAG 1
ldel05k0q_rawtag_a.fits FUV PSA G160M 1611 TIME-TAG 2
ldel05k2q_rawtag_a.fits FUV PSA G160M 1611 TIME-TAG 3
ldel05k4q_rawtag_a.fits FUV PSA G160M 1611 TIME-TAG 4
```

To remove a member from association, `ldel05050_asn.fits`, use the modules **astropy.io.fits** and **astropy.table.Table** to read in the association and edit the table as follows:

```
>from astropy.io import fits
>from astropy.table import Table
>hdulist = fits.open('ldel05050_asn.fits',mode='update')
>tbdata = hdulist[1].data
>print(Table(tbdata))

 MEMNAME   MEMTYPE MEMPRSNT
--------- ------- --------
LDEL05JYQ  EXP-FP     True
LDEL05K0Q  EXP-FP     True
LDEL05K2Q  EXP-FP     True
LDEL05K4Q  EXP-FP     True
LDEL05050 PROD-FP     True

>tbdata['MEMPRSNT'][0] = False
>tbdata['MEMPRSNT'][1] = False
>print(Table(tbdata))

 MEMNAME   MEMTYPE MEMPRSNT
--------- ------- --------
LDEL05JYQ  EXP-FP    False
LDEL05K0Q  EXP-FP    False
LDEL05K2Q  EXP-FP     True
LDEL05K4Q  EXP-FP     True
LDEL05050 PROD-FP     True

>hdulist.close()
```

Finally, reprocess the data by running **calcos** on the updated association file.

Run calcos

In stenv, users may choose between two methods to run **calcos** using Python or the Unix/Linux/Mac command line. The input arguments and examples for each case are as follows:

1. To run calcos in Python:

```
> import calcos
> calcos.calcos(’filename_asn.fits’, verbosity=2, \ outdir="new")
```

2. To run **calcos** from the Unix/Linux/Mac command line:

```
% calcos -o new --stim stim.txt filename_asn.fits
```

Table 3.2: Arguments for Running calcos in Python.

Argument	Values	Default	Description

asntable	"filename"	" "	Association table (asn) or individual raw file (`rawtag`, `rawaccum`) to be processed
outdir	directory name	None	The name of the ouptut directory
verbosity	0, 1, 2	1	0=quiet, 1=verbose, 2=very verbose
find_target	True or False	False	Have **calcos** find the spectrum location and center the extraction box on that location
create_csum_image	True or False	False	If True, write an image that reflects the counts detected at each pixel (includes deadcorr but not flatcorr), for OPUS to add to the cumulative image.
shift_file	"filename"		File containing wavecal shifts (will override shifts calculated by **calcos**)
save_temp_files	True or False	False	Save temporary files: `x1d_a`, `x1d_b`, `lampflash_a`, and `lampflash_b`
stimfile	"filename"		If specified, the stim pulse positions will be written to (or appended to) this text file
livetimefile	"filename"		If specified, the livetime factors wil be written to (or appended to) this text file
burstfile	"filename"		If specified, burst information will be written to (or appended to) this text file
raw_csum_coords	True or False	False	If True, use raw pixel coordinates (rather than thermally and geometrically corrected) to create the csum image.
only_csum	True or False	False	If True, create a csum image, but most other files will not be written.
binx, biny	`int` or None	None	Binning factor for the X and Y axes, or None, which means that the default binning (currently 1) should be used.
compress_csum	True or False	False	If True, compress the "calcos sum" image.
compression_parameters	string	"gzip, –0.01"	Two values separated by a comma; the first is the compression type (rice, gzip or hcompress), and the second is the quantization level.

Table 3.3: Command-line Options for Running calcos in Unix/Linux/Mac.

Option	Description
--version	Print the version number and exit
-r	Print the full version string and exit
-q	Quiet

-v	Very verbose
-s	Save temporary file
-o outdir	Output directory
--find yes	Have **calcos** find Y location of spectrum
--find no	Extract spectrum at default location
--find cutoff	Find Y location if sigma <= cutoff
--shift filename	File to specify shift values
--stim filename	Append stim pulse locations to filename
--live filename	Append livetime factors to filename
--burst filename	Append burst information to filename
--csum	Create 'calcos sum' image
--only_csum	Do little else but create csum
--raw	Use raw coordinates for csum image
--compress parameters	Compress csum image
--binx X_bin_factor	csum bin factor in X
--biny Y_bin_factor	csum bin factor in Y

To redirect the **calcos** `STDOUT` to a file use the following command:

```
% calcos -v -o new filename_asn.fits > log.txt
```

While we recommend that users run **calcos** on association files, it is possible to run calcos with a single `raw` or `corrtag` file as the input. In this mode, **calcos** will always automatically process both segment files for FUV data if they both exist. For example if `rootname_rawtag_a.fits` is the input for **calcos**, then `rootname_rawtag_b.fits` will automatically be processed. The data from both segments will be calibrated and combined to create the final product, `rootname_x1d.fits`.

Running **calcos** on `rawtag` or `corrtag` files instead of the `asn` file will cause the FUVB-only blue modes (G130M cenwaves 1055 and 1096) to be calibrated without the associated segment A EXP-IWAVE file contained in the `asn`.

3.6.2 Using GO Wavecals

Through the use of associations, **calcos** also contains a provision to select wavecals other than the default for calibration of the science exposures. To use an exposure other than or in addition to the default wavecal, the user can add a row to the association table. The rootname (case insensitive) should be given in the MEMNAME column, the string EXP-GWAVE in the MEMTYPE column, and the value in the boolean MEMPRSNT column set to true. Make sure that the `WAVECORR` keyword in the primary header of the raw science file is set to PERFORM, and then run **calcos** as normal. **Note that GO wavecals can only be used with non `TAGFLASH` data.**

3.6.3 Customizing the TWOZONE extraction

The TWOZONE extraction algorithm and the associated pipeline reference files are optimized for the spectral extraction of bright point source spectra. There are, however, a number of circumstances under which a customized extraction might yield better results.

- For extended sources the use of the TWOZONE algorithm may lead to an underestimate of the measured flux and/or poor alignment of the extraction region with the source.
- For very faint point or extended sources, the ALGNCORR step may not always be able to reliably measure the position of the source, and may default to assuming that the target is already aligned with the reference profile.
- For some very faint point sources it may be possible to significantly improve the signal-to-noise by reducing the extraction height to minimize the included detector background.

If the user simply wishes to use the BOXCAR extraction in place of the TWOZONE algorithm, XTRCTALG should be set to "BOXCAR," and TRCECORR and ALGNCORR should be set to "OMIT." This will use the larger extraction regions defined for that algorithm. However, for observations at LP3 there may be significant overlap with the gain-sagged regions near LP1, and this may affect the accuracy of the calibration or even create artificial spectral features. This is similarly true of LP4 overlapping LP3, although to a lesser extent as LP3 is not as severely gain-sagged as LP1.

Both the XTRACTAB, which is used with the BOXCAR algorithm, and the TWOZXTAB, which is used with the TWOZONE algorithm, contain columns named HEIGHT and B_SPEC. For the BOXCAR algorithm, these parameters together with the SLOPE column directly control the size and location of the extraction region. For the TWOZONE algorithm, the HEIGHT and BSPEC numbers instead control the size and initial location of the region used for the ALGNCORR step. The HEIGHT column in the TWOZONE algorithm is also used to define the cross-dispersion width of the reference profile that is assumed to include 100% of the enclosed energy. The actual extraction region at each wavelength is adjusted so that the enclosed energy fraction of the reference profile matches the values given in the LOWER_OUTER and UPPER_OUTER columns of the TWOZXTAB. For example, if LOWER_OUTER=0.005 and UPPER_OUTER=0.995, at each wavelength the extraction region will be adjusted so that the central 99% of the encircled energy as measured from the reference profile is included. Fractional pixel locations are rounded outwards, and the final extracted flux will be scaled for the exact encircled energy fraction in each column.

To force a spectral extraction using the TWOZONE algorithm to sum over a region that contains only the central 80% of the reference profile's encircled energy, the user would just need to change LOWER_OUTER to 0.1 and UPPER_OUTER to 0.9 in the appropriate row of the TWOZXTAB prior to recalibration of the data.

Values of 0 or 1 for the enclosed energy boundaries have a special meaning. Setting the lower boundary to a value of 0 forces the extraction to start at the bottom of the region defined by a rectangular box of size HEIGHT, while setting the upper boundary to 1, forces it to end at the upper boundary. This can be used to give a rectangular extraction box rather than the wavelength dependent extraction region normally used for the TWOZONE algorithm. For a very extended target, it might be useful to force the use of the full height box, and also increase the HEIGHT allowing a further expansion of the extraction region.

The LOWER_INNER and UPPER_INNER columns in the TWOZXTAB behave very similarly to the "OUTER" boundaries, except that they are used to control the region over which data quality flags are combined rather than the region over which counts are summed. The user can also adjust these values.

The background regions in the TWOZONE algorithm are handled in a simpler fashion. To change where the background regions are located or the height of the background regions, edit the background centers (B_BKG1 and B_BKG2) and the background height (BHEIGHT). The background regions should not be placed directly above the spectrum at LP3, as that is where LP1 is located, and the detector is therefore very gain-sagged in that location. Similarly, the background regions should not be placed directly above the spectrum at LP4, as that is where LP3 is located. Also ensure that the background regions do not overlap the WCA (location found in XTRACTAB). See Figure A.1 in Appendix A for regions with low levels of gain sag.

The user can also override the shifts calculated by ALGNCORR. This can be useful if the automatic algorithm failed to properly center the target. To do this, the user should set the keyword SP_SET_A, (for detector segment FUVA), or SP_SET_B, (for FUVB), to the desired offset value which will be used in place of the SP_OFF_A or SP_OFF_B value calculated by the ALGNCORR algorithm. These keywords should be set in the extension header of the `rawtag` or `corrtag` file used as input for **calcos**.

3.7 Reference Files

This section contains a description of the COS reference files. See Figure 3.1–Figure 3.6 for which modules use these files and Section 3.4 for explanations of how their contents are applied by those modules. The reference files are now described in the order they are called by the pipeline for the case of FUV TIME-TAG data (Figure 3.1).

3.7.1 BADTTAB: Bad Time Interval Table

- File Suffix: `_badt`

The `BADTTAB` reference file lists the start and end times of known bad time intervals. It is used by the BADTCORR calibration module to flag events in `TIME-TAG` events lists which occur during a bad time interval. In later processing the flagged events will be removed from the final calibrated data, and the exposure time header keyword, `EXPTIME`, updated. The bad time interval table consists of segment, start, and end columns (see Table 3.4). The segments columns can be populated with either FUVA, FUVB or ANY. The start and end columns are in Modified Julian Date.

Table 3.4: `BADTTAB` Table Content.

Column Name	Data Type	Description
SEGMENT	String	Detector segment, FUVA, FUVB or ANY
START	Double	Bad time interval start time in MJD

END	Double	Bad time interval end time in MJD

3.7.2 BRFTAB: Baseline Reference Frame Table

- File Suffix: `_brf`

The `BRFTAB` reference file is only applicable to FUV data and is used during pipeline processing in the TEMPCORR module to apply the thermal distortion correction. The FUV detector does not have physical pixels like a CCD. Instead, the x and y positions of detected photon events are obtained from analog electronics, which are susceptible to thermal changes. Electronic stim pulses are normally commanded during integration and are used as physical position reference points. To return the FUV data to a known physical space, the BRFTAB defines the stim pulse positions.

The `BRFTAB` file consists of a primary header extension and a binary table extension. The table lists the stim pulse locations and search regions, and the active detector areas (Table 3.5).

Table 3.5: `BRFTAB` Table Contents.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA or FUVB
SX1	Double	X pixel coordinate (zero indexed) of stim pulse 1[1]
SY1	Double	Y pixel coordinate (zero indexed) of stim pulse 1
SX2	Double	X pixel coordinate (zero indexed) of stim pulse 2[2]
SY2	Double	Y pixel coordinate (zero indexed) of stim pulse 2
XWidth	Long	Half width of search region for stim pulses
YWidth	Long	Half height of search region for stim pulses
A_Left	Long	X pixel of left side of active region
A_Right	Long	X pixel of right side of active region
A_Low	Long	Y pixel of lower side of active region
A_High	Long	Y pixel of upper side of active region

[1] Stim pulse 1 is located in the upper left corner.

[2] Stim pulse 2 is located in the lower right corner.

3.7.3 GEOFILE: Geometric Correction File

- File Suffix: `_geo`

This file is only used for FUV data. The `GEOFILE` is used by the GEOCORR calibration module to perform the geometric correction. The analog nature of the XDL detector means that the physical sizes of the pixels vary across the detector. The geometric distortion maps are used to correct for this variation and to transform the data into a constant physical pixel size early in the data reduction calibration process. After the thermal correction has been applied, the geometric correction can be applied. This implies that all the files used to determine the geometric correction were initially thermally corrected.

Each geometric correction reference file contains four IMAGE extensions. There are two for each segment, and for each segment, there is one for each axis. At a given (X,Y) location in the thermally corrected COS data, the value at that location (corrected for binning and offset) in the geometric correction image gives the distortion to be subtracted from the X or Y coordinates. The order of the extensions are: 1=X coordinate for FUVA, 2=Y coordinate for FUVA, 3=X coordinate for FUVB and 4=Y coordinate for FUVB. This information is also available in the header file, with keywords `EXTVER` (1 for x and 2 for y) and `EXTNAME` (FUVA or FUVB).

3.7.4 DGEOFILE: Delta Geometric Correction File

- File Suffix: `_dgeo`

The delta geometric distortion reference file is used to improve the geometric correction for the FUV detector. It is defined and applied in the same way as the geometric correction, and is only applied to data that have been geometrically corrected. At a given (X,Y) location in the geometrically corrected COS data, the value at that location (corrected for binning and offset) in the delta geometric correction image gives the distortion to be subtracted from the X or Y coordinate.

Each delta geometric correction reference file contains four IMAGE extensions. There are two for each segment, and for each segment, there is one for each axis. At a given (X,Y) location in the corrected COS data, the value at that location (corrected for binning and offset) in the delta geometric correction image gives the distortion to be subtracted from the X or Y coordinates. The order of the extensions are: 1=X coordinate for FUVA, 2=Y coordinate for FUVA, 3=X coordinate for FUVB and 4=Y coordinate for FUVB.

3.7.5 XWLKFILE, YWLKFILE: X Walk Correction, Y Walk Correction

- File Suffix: `_xwalk`, `_ywalk`

The `XWLKFILE` and `YWLKFILE` reference files are only applicable to FUV data and are used during pipeline processing in the XWLKCORR and YWLKCORR modules to correct the effects of walk. The COS FUV XDL detector is subject to gain sag; as physical locations on the detector accumulate photon events, the number of electrons in the charge cloud generated by an event becomes smaller, and as a result the coordinates of the event may be mis-registered by the electronics. These effects depend on event pulse height and the position on the detector.

The design of the `XWLKFILE` and `YWLKFILE` allow both the X and Y position to be corrected based on the geometrically corrected X location and the pulse height. Each file consists of a primary header and a two binary table extension (one for each segment). One extension (EXTNAME='FUVA') is used to correct the data on Segment A, and the other (EXTNAME='FUVB') is used for Segment B. Details of how the correction is applied are given in the XWLKCORR/YWALKCORR section (Section 3.4.6). The data in these files will be updated in the future as the walk correction is further refined. Note that these files replace the original `WALKTAB`, which is now deprecated.

3.7.6 DEADTAB: Deadtime Table

- File Suffix: `_dead`

The `DEADTAB` reference file is used in the DQICORR: Initialize Data Quality File module, to obtain the true number of events received compared to the number of events appearing in the raw data files.

There is one `DEADTAB` reference file for each of the NUV and FUV detectors. Each consists of a primary header and a binary table extension which contains the `LIVETIME` values for a given observed count rate (`OBS_RATE`) and segment. The livetime is defined as:

$$\text{livetime} = \text{observed rate/true rate}$$

and can be used to calculate the true count rate.

3.7.7 PHATAB: Pulse Height Discrimination Table

- File Suffix: `_pha`

The `PHATAB` reference file is only valid for FUV data, and is applied during the PHACORR step of **calcos** to filter non-photon events. The file consists of two header/data units, the first being the primary header, and the second a binary table (see Table 3.6). The table lists the lower and upper thresholds for valid individual pulse heights in `TIME-TAG` mode. In `TIME-TAG` mode, each detector event has an associated pulse-height of 5 bits with values ranging from 0 to 31, The table also gives the minimum and maximum values for the location of the mean value of the pulse height distribution used in `ACCUM` mode. In `ACCUM` mode, a pulse height distribution histogram is generated for the whole exposure over the entire detector and downloaded as part of the science data file. The histogram includes all the digitized events for each segment independently of the currently defined subarrays. Note in `ACCUM` mode the pulse height is a 7 bit number with values ranging from 0 to 127.

Table 3.6: `PHATAB` Table Contents.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA or FUVB
LLT	Long	Lower limit threshold (TIME-TAG)
ULT	Long	Upper limit threshold (TIME-TAG)
MIN_PEAK	Float	Lower limit for location of mean (ACCUM)
MAX_PEAK	Float	Upper limit for location of mean (ACCUM)

3.7.8 PHAFILE: Pulse Height Discrimination File

- File Suffix: `_phf`

This file is only used for FUV data, and is a 2D equivalent to the `PHATAB`. The `PHAFILE` is used by the PHACORR calibration module to filter non-photon events. If both a `PHATAB` and `PHAFILE` are available, the `PHAFILE` will be used.

Each pulse height discrimination reference file contains four IMAGE extensions. There are two for each segment, containing the lower and upper PHA limits for each pixel. At a given (X,Y) location in the uncorrected COS data, the value at that location gives the lowest and highest (respectively) pulse height that will be treated as a valid photon event at that detector location.

3.7.9 FLATFILE: Flat-field File

- File Suffix: `_flat`

`FLATFILE` provides a flat-field image which is used by the pipeline to remove the pixel-to-pixel variations in the detector. The FUV `FLATFILE` consists of a primary header and two 14000 × 400 IMAGE extensions, one for each segment. This file is lifetime dependent. The NUV `FLATFILE` consists of a primary header and a 1024 × 1024 IMAGE extension.

The FUV flat-field reference files correct for grid wire shadows and for an effect of small-scale geometric distortion. There are multiple files for different combinations of lifetime position and grating (G130M, G160M, and G140L).

The NUV flat-field is a combination of internal and external deuterium flat field lamp exposures from thermal-vacuum testing which illuminate the portion of the detector where spectra fall. The data cover the following pixel region of the detector: x (dispersion): 0 to 1023, and y (cross-dispersion): 495 to 964. The rest of the detector, where flat field data are not available, has a value of 1.0. The bottom four and top three rows of the detector do not fit well with the rest of the detector and they are flagged in the data quality table.

3.7.10 LAMPTAB: Template Calibration Lamp Spectra Table

- File Suffix: `_lamp`

The `LAMPTAB` files consist of a primary header and a binary table extension which contains an extracted 1-D spectrum from the internal PtNe calibration lamp through the WCA aperture, for each grating, central wavelength, and FP-POS setting. It is used in the **calcos** pipeline to determine the pixel offset of the observed data. The structure of the template calibration lamp spectra table is shown in Table 3.7. The stepper motor offsets range from –2 to +1 and correspond to FP-POS settings of 1 to 4. Note that `LAMPTAB` files are lifetime position specific in the FUV. LP2 currently uses the `LAMPTAB` file from LP1. The NUV remains at the same LP1.

Table 3.7: `LAMPTAB` Table Contents.

Column Name	Data Type	Description
SEGMENT	String	Segment: FUVA, FUVB, NUVA, NUVB, NUVC
OPT_ELEM	String	Grating name
CENWAVE	Long	Central wavelength (Angstrom)
FPOFFSET	Integer	Array of stepper motor offsets
HAS_LINES	Boolean	Normally True; False if there is no lamp signal
FP_PIXEL_SHIFT	Double	Offset in pixels from FPOFFSET=0

INTENSITY	Float	Wavecal spectrum array

3.7.11 DISPTAB: Dispersion Coefficient Table

- File Suffix: `_disp`

There are multiple `DISPTAB` files in CRDS with similar formats, one for the NUV, and multiple for the FUV at different lifetime positions. They consist of a main header and a binary table in the second HDU. These tables provide the dispersion relations for each segment, aperture, optical element, and central wavelength. Each file has the format given in Table 3.8. The dispersion relation table gives a set of polynomial coefficients for computing wavelength from pixel number (see Oliveira et al., COS ISR 2010-05 and 06 for details).

Each row of the table gives a set of dispersion coefficients. The row to be used is selected on `SEGMENT`, `OPT_ELEM`, `CENWAVE`, and `APERTURE`. Note that DISPTAB files are lifetime position specific.

Table 3.8: DISPTAB Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA, FUVB, NUVA, NUVB, NUVC
OPT_ELEM	String	Grating name
APERTURE	String	Aperture name
CENWAVE	Long	Central wavelength of setting
NELEM	Long	Number of non-zero coefficients in the polynomial
COEFF	Double[4]	Coefficients, up to 4.
D_TV03	Double	Offset from WCA to PSA in Thermal Vac. 2003 data
D	Double	Current offset from WCA to PSA

For P_x = the zero-indexed Doppler corrected pixel value in the dispersion direction, let

$$P'_X = P_X + (D\_TV03 - D) \,,$$

then the corresponding wavelength in Angstroms is given by:

$$\lambda(P'_X) = COEFF[0] + COEFF[1] * P'_X + COEFF[2] * P'_X 2 + COEFF[3] * P'_X 3 \,.$$

3.7.12 XTRACTAB: 1-D Spectral Extraction Table

- File Suffix: `_1dx`

There are multiple `XTRACTAB` files with similar formats, one for the NUV and multiple for the FUV. The FUV `XTRACTAB` files are lifetime dependent. They consist of a main header and a binary table in the second HDU. These tables provide the information needed to extract the spectrum from a geometrically corrected image of the detector for each optical element and central wavelength. Each file has the format given in Table 3.9.

Table 3.9: `XTRACTAB` Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA, FUVB, NUVA, NUVB, NUVC
OPT_ELEM	String	Grating name
CENWAVE	Long	Central wavelength setting
APERTURE	String	Aperture name
SLOPE	Double	Slope of the spectral extraction box
B_SPEC	Double	Intercept of the spectrum extraction box
B_BKG1	Double	Intercept of the lower background spectral extraction box
B_BKG2	Double	Intercept of the upper background spectral extraction box
HEIGHT	Long	Height of the spectral extraction window
B_HGT1	Long	Height of lower background spectral extraction box
B_HGT2	Long	Height of upper background spectral extraction box
BWIDTH	Long	Width of the boxcar filter used to smooth the backgrounds

The spectral extraction of a source is performed by collapsing the data within a parallelogram of height HEIGHT that is centered on a line whose slope and intercept are given by SLOPE and B_SPEC. Similarly, two background spectra are determined by collapsing the data within parallelograms of height B_HGT1 or B_HGT2 centered on the lines defined by SLOPE and B_BKG1, and SLOPE and B_BKG2, respectively. The background spectra are then smoothed by a boxcar of width BWIDTH. These are then scaled and subtracted from the source spectrum.

3.7.13 BRSTTAB: Burst Parameters Table

- File Suffix: `_burst`

The `BRSTTAB` file is used for FUV data only. It provides the parameters needed to identify bursts. It consists of a primary header extension and a binary table extension with the columns listed in Table 3.10. Details of the burst rejection routine are given in Section 3.4.13.

Table 3.10: `BRSTTAB` Table Contents.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA or FUVB

MEDIAN_N	Double	Factor above the median count rate for a time interval to be identified as a burst
DELTA_T	Double	Normal sampling time for large burst detection(s)
DELTA_T_HIGH	Double	High count rate sampling time for large burst detection(s)
MEDIAN_DT	Double	Time interval used to search for localized bursts(s)
BURST_MIN	Double	Minimum threshold rate for small bursts (counts/s)
STDREJ	Double	Number of standard deviations above background noise for small bursts
SOURCE_FRAC	Double	Minimum factor small bursts must be above source counts.
MAX_ITER	Long	The maximum number of iterations used to re-evaluate the median to detect a localized burst
HIGH_RATE	Double	Total count rate threshold to use DELTA_T_HIGH instead of DELTA_T (counts/s)

3.7.14 BPIXTAB: Bad Pixel Table

- File Suffix: `_bpix`

The data quality initialization table identifies rectangular regions on the detectors that are known to be less then optimal. The feature type describes the type of detector blemish enclosed within the bounding box and DQ is the quality value assigned to all events detected within the box. The regions were identified by visual inspection of the combined flat field data for each detector (and segment). The `BPIXTAB` files consist of a primary header and a binary table extension which consists of the columns listed in Table 3.11.

Table 3.11: BPIXTAB Table Content.

Column Name	Data Type	Description
SEGMENT	String	Segment name, FUVA, FUVB, or ANY for NUV
LX	Long	X coordinate of lower left corner of region
LY	Long	Y coordinate of lower left corner of region
DX	Long	Width of region in X
DY	Long	Width of region in Y
DQ	Long	Data quality value to assign to current region
TYPE	String	Comment regarding current region

In the `BPIXTAB` table, the DQ field may be a logical OR due to several different values, each associated with a unique issue (see Table 2.19).

3.7.15 GSAGTAB: Gain Sag Table

- File Suffix: `_gsag`

The gain sag reference table is only applicable for FUV data and it is used along with the bad pixel reference table (`_bpix`) in the DQICORR module. The table provides the locations of rectangular regions for portions of the FUV detector that have very low pulse height amplitude.

After the primary header, each extension of the GSAGTAB is a binary fits table of the gain sagged pixels on the detector at a given voltage. During the pipeline processing, these extensions are selected depending on the SEGMENT and HVLEVEL. Each row in the table gives the location and data quality value for one rectangular region. The DATE column is used to select rows. A row will be used to flag a gain sagged region if the value in the DATE column is less than or equal to the exposure start time. For a description on the columns contained in the binary tables see Table 3.12.

Table 3.12: GSAGTAB Table Format.

Column Name	Data Type	Description
DATE	Double	Modified Julian Date at which the PHA in a region dropped so low that the region should be flagged as gain-sagged
LX	Long	X coordinate of lower left corner of region
LY	Long	Y coordinate of lower left corner of region
DX	Long	Width of region in X
DY	Long	Width of region in Y
DQ	Long	Data quality value assigned to current region

3.7.16 SPOTTAB: Hotspot Table

- File Suffix: `_spot`

The hotspot table is only applicable for FUV data, and is used along with the bad pixel reference table (`_bpix`) and gain sag table (`_gsag`) in the DQICORR module. The table provides the start and stop times, locations and extents of hotspots, which are transient regions of high detector background.

The hotspot table is a FITS table with a primary header and 1 extension with optional EXTNAME = HOTSPOT. Each row has 9 columns: SEGMENT is the segment name the hotspot appears in (FUVA or FUVB). START and STOP are the MJD times of the start and stop of the hotspot. LX and LY are the (XCORR, YCORR) coordinates of the lower left corner of the rectangular hotspot region. DX and DY are the extent, in pixels, of the rectangular hotspot region. DQ is the value of the DQ flag to be applied to the region (see Table 2.19), and COMMENT is a comment string.

The following table describes the column definitions.

Table 3.13: `SPOTTAB` Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment name (FUVA, FUVB)
START	Float	MJD time of start of hotspot
STOP	Float	MJD time of end of hotspot
LX	Integer	X coordinate of lower left corner of region
LY	Integer	Y coordinate of lower left corner of region
DX	Integer	Width of region in X
DY	Integer	Width of region in Y
DQ	Integer	Data quality value assigned to current region
COMMENT	String	Comment

The hotspot is selected based on the value of SEGMENT, and then the START and STOP times are compared to the start and stop times of the good time intervals of the exposure being calibrated. If a hotspot overlaps any of the good time intervals, the region is added to the set of regions that are applied to create the DQ mask and against which each event is tested to assign a DQ value. The hotspot regions are flagged in the two-zone extraction module even if they are only in the outer zone, and they do not contribute to the summed spectra in the `x1dsum` file.

3.7.17 WCPTAB: Wavecal Parameter Table

- File Suffix: `_wcp`

The `WCPTAB` file contains information relevant for the wavecal pipeline processing. This file is lifetime dependent. It consists of primary header and a binary table extension which is described in Table 3.14. `XC_RANGE` is the maximum pixel offset to use when doing a cross correlation between the observed data and the template wavecal. That is, the observed spectrum should be shifted relative to the template by a number of pixels, ranging from –`XC_RANGE` to +`XC_RANGE` inclusive. `XD_RANGE` is half the search range for finding the spectrum in the cross dispersion direction. The search range is from `B_SPEC–XD_RANGE` to `B_SPEC+XD_RANGE` inclusive, where `B_SPEC` is the nominal location of the spectrum from the XTRACTAB table discussed below. `BOX` is the width of the boxcar filter for smoothing the cross-dispersion profile. `RESWIDTH` is the number of pixels per resolution element, and is assigned a value of 6.0 for the FUV detectors and 3.0 for the NUV detector.

When applying the offsets found from the wavecals to the science data, it may happen that there was no wavecal at the same OSM position. In this case, the wavecal that was closest in time to the science observation may be used, with a correction for the difference in OSM positions. That correction is based on `STEPSIZE`, the number of pixels corresponding to one OSM step. There may be a check, however, to guard against using a wavecal that was taken too far away in time from the science observation. If the science observation and wavecal were taken more than `MAX_TIME_DIFF` apart, then the wavecal should not be used for that science observation.

Table 3.14: `WCPTAB` Table Contents.

Column Name	Data Type	Description
OPT_ELEM	String	Grating name
XC_RANGE	Long	Maximum Lag (amplitude) for cross correlation
SEARCH_OFFSET	Double	Zero-point offset for the search range
N_SIGMA[1]	Double	Minimized chi square threshold
RESWIDTH	Double	Number of pixels per resolution element in the dispersion direction
MAX_TIME_DIFF	Double	Defines 'close in time' for wavecals
STEPSIZE	Long	One step of OSM is this many pixels
XD_RANGE	Long	Amplitude of search range for finding spectrum
BOX	Integer	Width of boxcar smoothing filter

[1] Only at LP4.

3.7.18 FLUXTAB: Photometric Throughput Table

- File Suffix: `_phot`

There are multiple `FLUXTAB` files with similar formats, one for the NUV, and multiple for the FUV. The FUV `FLUXTAB` files are lifetime dependent. They consist of a main header and a binary table in the second HDU. These tables provide the information needed to convert from corrected detector counts to flux units of erg s $^{-1}$ cm^{-2} Å^{-1} for each segment, optical element, aperture, and central wavelength. Each file has the format given in Table 3.15.

Table 3.15: `FLUXTAB` Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment Name
OPT_ELEM	String	Name of optical element
CENWAVE	Long	Central wavelength of the setting
APERTURE	String	Name of the aperture
WAVELENGTH	Double	Wavelength array in Angstroms
SENSITIVITY	Float	Sensitivity array

The units of the Sensitivity array are (count s^{-1} $pixel^{-1}$)/(erg s^{-1} cm^{-2} $Angstrom^{-1}$). For each segment, optical element, central wavelength setting, and aperture, these files contain arrays of wavelengths and sensitivities which can be interpolated onto the observed wavelength grid. The net counts can then be divided by the sensitivity curves to produce flux calibrated spectra.

3.7.19 TDSTAB: Time Dependent Sensitivity Table

- File Suffix: `_tds`

There are two such files, one for the FUV and one for the NUV. They are only used for spectroscopic data. The files contain the information necessary to determine the relative sensitivity curve at any given time by interpolating between relative sensitivity curves given at fiducial times which bracket the observation, or else extrapolate the results from the last curve if the observation date is more recent than the last fiducial date. Interpolation data are provided for each segment, optical element, and aperture (see Table 3.16). Updated TDS plots are located at:
http://www.stsci.edu/hst/instrumentation/cos/performance/sensitivity.

Table 3.16: TDSTAB Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment Name
OPT_ELEM	String	Name of optical element
APERTURE	String	Name of the aperture
NWL	Long	Number of wavelength points
NT	Long	Number of time points
WAVELENGTH	Double[NWL]	Wavelength array in Angstroms
TIME	Double[NT]	Fiducial times in MJD
SLOPE	Double[NWL, NT]	Percent per year
INTERCEPT	Double[NWL, NT]	Ratios of current curve to original curves

For an observation obtained at time T, which lies between TIME[j] and TIME[j+1], the sensitivity curve used to calibrate the spectrum will be corrected by the following factor:

$$(\mathrm{T}-\mathrm{REF\_TIME})\mathrm{SLOPE}[\mathrm{i,j}]/(365.25*100)+\mathrm{INTERCEPT}[\mathrm{i,j}],$$

where REF_TIME is a general reference time given in the header of the FITS extension.

3.7.20 TRACETAB: Trace Correction Table

- File Suffix: `_trace`

The trace table gives the variation of the centroid of the spectrum as a function of column number (XCORR) in COS FUV data. This file is lifetime dependent, but currently only provided for LP3 and greater.

The file is a FITS table with a primary header and one extension. The row to be used is selected on SEGMENT, OPT_ELEM, CENWAVE and APERTURE. Each row has 8 columns. DESCRIP supplies a short description, while TRACE_YLOC is the location of the center of the trace. TRACE is an array of 16,384 floats where the index is the value of XCORR and the value is the offset to be subtracted from each event's YFULL value. The value of XCORR for each event is interpolated onto the TRACE array to give the value of the shift to be applied to the corresponding YFULL value of the event. ERROR is an array of 16,384 floats that gives the statistical error of the TRACE measurement. Table 3.17 describes the column definitions.

Table 3.17: TRACETAB Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment name (FUVA or FUVB)
OPT_ELEM	String	Grating name
CENWAVE	Integer	Central wavelength (Angstrom)
APERTURE	String	Aperture name (PSA or BOA)
DESCRIP	String	Description
TRACE_YLOC	Float	YCORR location of center of trace (median)
TRACE	Float	Trace profile y-location array
ERROR	Float	Trace profile error array

3.7.21 PROFTAB: Profile Table

- File Suffix: `_profile`

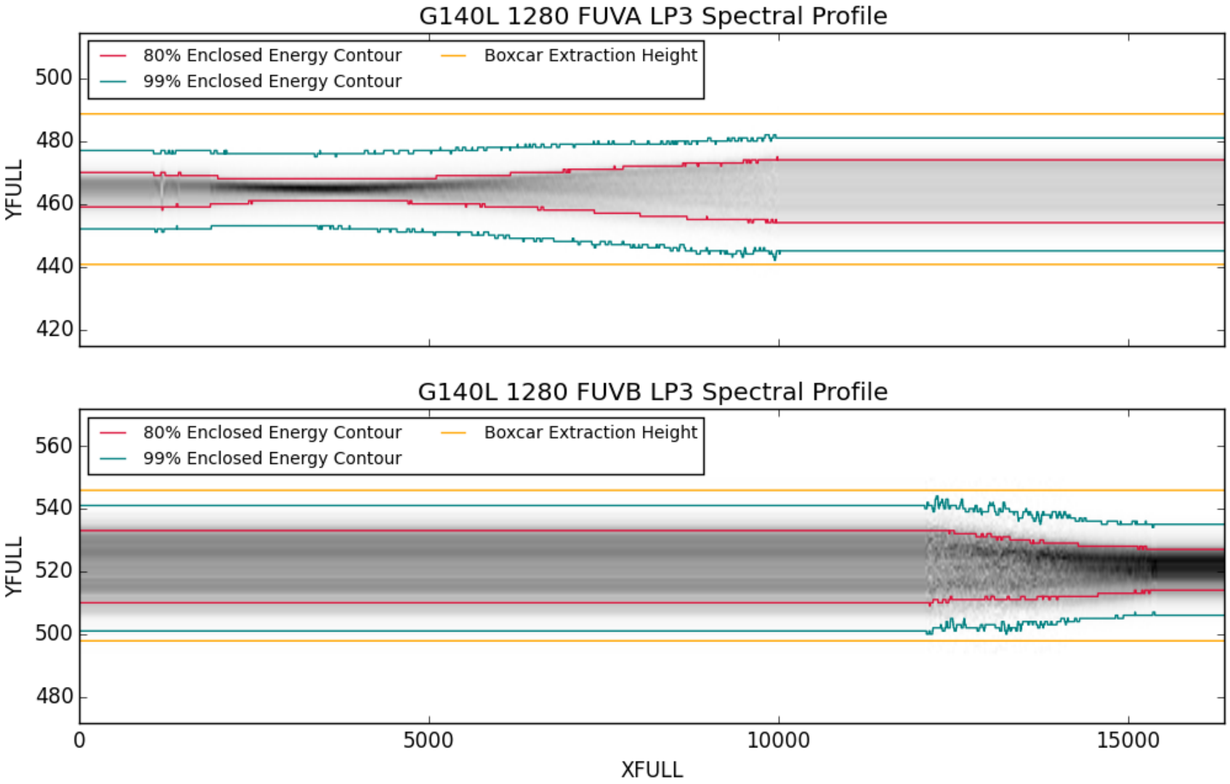
The profile table gives the profile of a point source perpendicular to the dispersion direction as a function of column number (XFULL) in COS FUV data. This file is lifetime dependent, but currently only provided for LP3 and greater.

PROFTAB is a FITS table with a primary header and one extension with optional EXTNAME = PROFILE. The row to be used is selected on SEGMENT, OPT_ELEM, CENWAVE and APERTURE. Each row has 8 columns. DESCRIP gives a short description of the row. CENTER is the measured centroid of the profile in the full-sized array in (XFULL, YFULL) coordinates. ROW_0 is the index of the first row of the profile in the full-sized array. In other words, if the profile has NROWS rows, it corresponds to rows with 0-based indices running from ROW_0 to (ROW_0 + NROWS − 1). PROFILE is the 2-d array of floats that gives the profile in the cross-dispersion direction for each column of data in (XFULL, YFULL) space (offset by ROW_0). During the ALGNCORR step, the flux-weighted centroid of the science data over 'good' rows and columns is calculated, and compared with the flux-weighted centroid of the profile contained in this reference file over the same rows and columns. The difference between these centroids is applied to the YFULL values of the events to align each set of science data to the same center. Table 3.18 describes the column definitions. The 2D spectral profiles contained in the PROFTAB for three settings (G140L/1280, G130M/1291, and G160M/1577) are given in Figure 3.17, Figure 3.18, and Figure 3.19.

Table 3.18: `PROFTAB` Table Format.

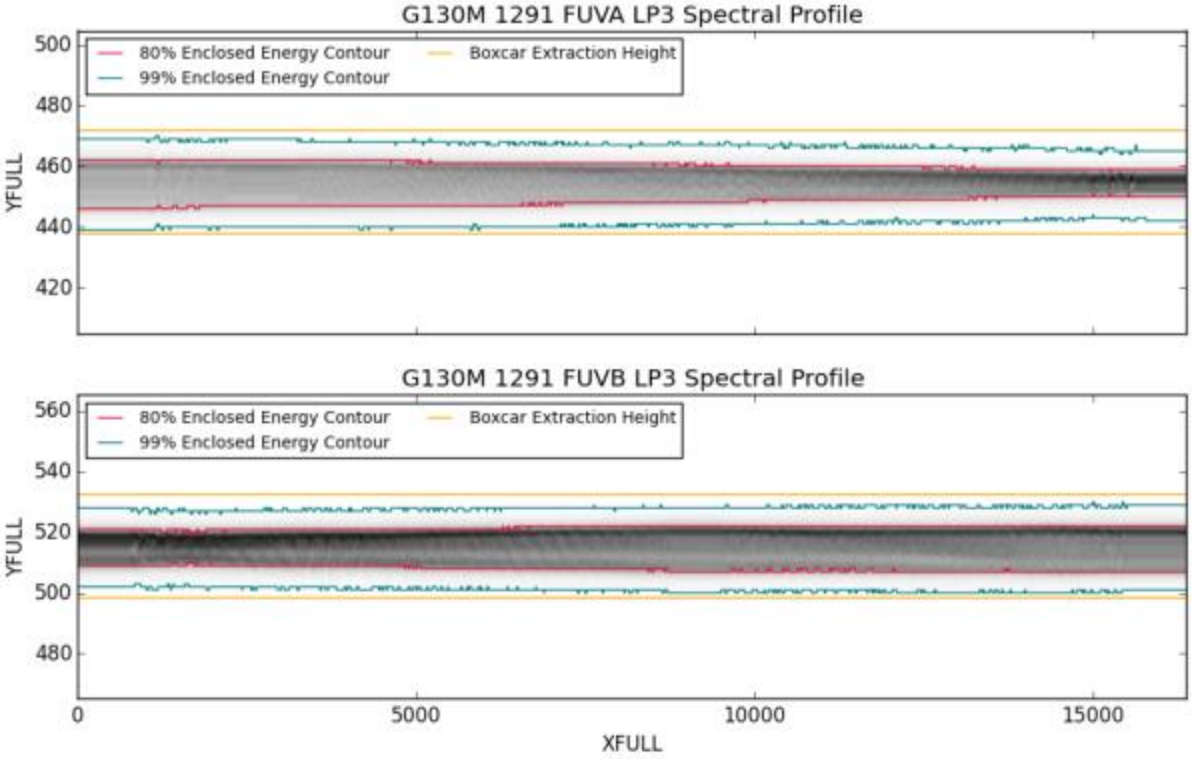
Column Name	Data Type	Description
SEGMENT	String	Segment name (FUVA or FUVB)
OPT_ELEM	String	Grating name
CENWAVE	Integer	Central wavelength (Angstrom)
APERTURE	String	Aperture name (PSA, BOA, or ANY)
DESCRIP	String	Description
CENTER	Float	Profile centroid
ROW_0	Integer	Row offset of profile array
PROFILE	Float	Profile in (XFULL, YFULL), offset by ROW_0

Figure 3.17: 2D Reference Profile for G140L/1280.



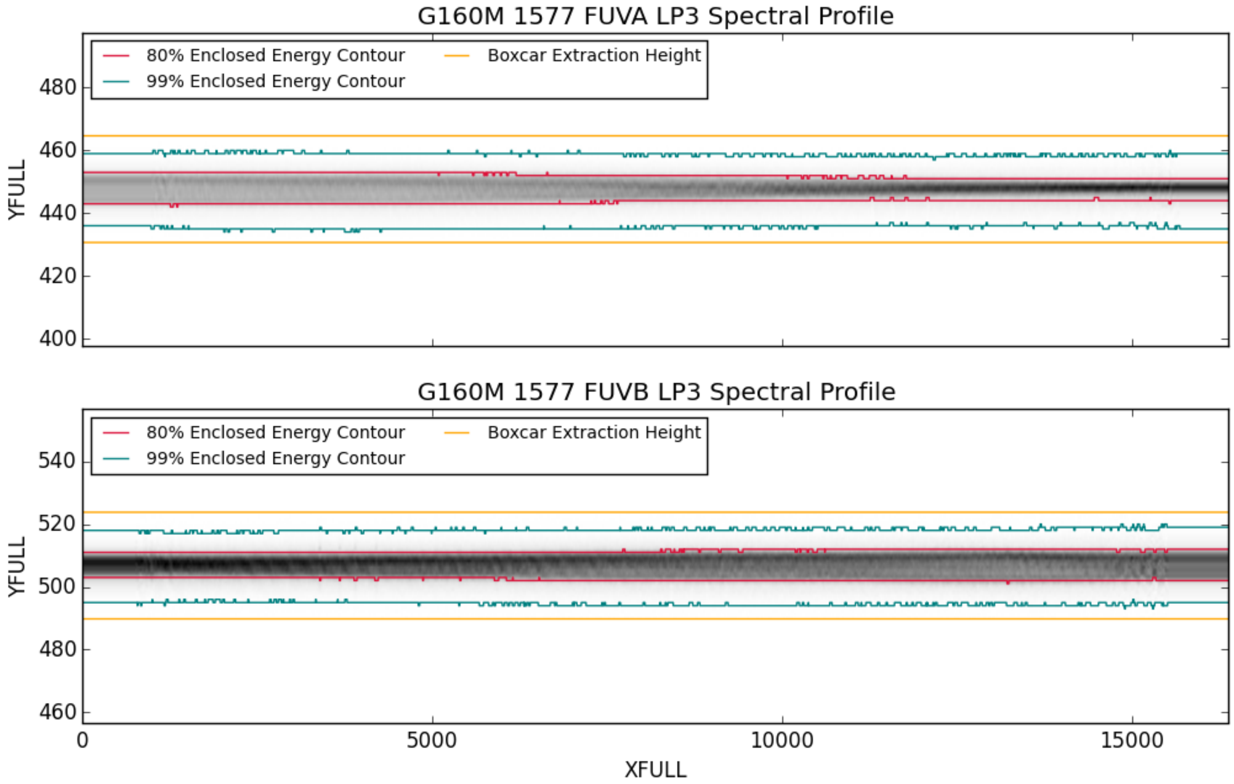
An image of the 2D reference profile in the PROFTAB for the G140L/c1280 setting at LP3 is show for each detector segment. The 80% and 99% enclosed energy contours that are currently used to define the inner and outer zones of the two zone extraction are also marked.

Figure 3.18: 2D Reference Profile for G130M/1291.



The same as Figure 3.17, expect that the profiles shown here are for the G130M/1291 setting. Note that for this example the cross-dispersion profile is double peaked at many wavelengths due to the cross-dispersion astigmatism of the G130M grating.

Figure 3.19: 2D Reference Profile for G160M/1577.



The same as Figure 3.17, expect that the profiles shown here are for the G160M/1577 setting.

3.7.22 TWOZXTAB: TWOZONE Spectral Extraction Table

- File Suffix: `_2zx`

The TWOZONE extraction table contains the starting values for the object center and background regions, as well as the cumulative flux boundary values for the TWOZONE extraction. This file is lifetime dependent, but currently only provided for LP3 and greater.

TWOZXTAB is a FITS table with a primary header and one data extension. The row to be used is selected on SEGMENT, OPT_ELEM, CENWAVE and APERTURE. Each row has 16 columns. B_SPEC is the center of the science extraction aperture, and is used by the ALGNCORR step to get an initial guess for the location of the spectral trace. B_BKG1 and B_BKG2 are the center of the background regions, HEIGHT is the height of the target extraction region, and BHEIGHT is the height of the background extraction regions. BWIDTH is the width of the smoothing box used to smooth the background region in the extraction step.

In the TWOZONE extraction step, the spectral profile in the PROFTAB is analyzed to determine the boundaries of INNER and OUTER zones. These boundaries are specified in terms of the cumulative flux enclosed. In the INNER region, the flux is summed within the region and any DQ flags are propagated to the extracted spectrum. In the OUTER region, the flux is also summed and added to the flux in the inner region, but any DQ flags in the outer region are not propagated to the final extracted spectrum unless they are in the DQ value SDQOUTER from the primary header. The columns LOWER_OUTER, UPPER_OUTER, LOWER_INNER and UPPER_INNER give the cumulative flux boundaries to be used in the two zone extraction. Typically the outer boundaries enclose 99% of the flux, while the inner boundaries enclose 80%.

The YERRMAX column is used in the ALGNCORR step to test the statistical error in the calculation of the flux-weighted centroid of the science data. If this measurement is greater than the value of YERRMAX for that setting, the spectrum is deemed 'not found,' and the location of the center of the reference profile is used instead. The PEDIGREE column gives the pedigree of the information in the row, with values that are typically INFLIGHT, GROUND or DUMMY.

Table 3.19 describes the column definitions.

Table 3.19: `TWOZXTAB` Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment name (FUVA or FUVB)
OPT_ELEM	String	Grating name
CENWAVE	Integer	Central wavelength (Angstrom)
APERTURE	String	Aperture name (PSA or BOA)
B_SPEC	Float	Location of center of object aperture
B_BKG1	Float	Location of center of lower background extraction aperture
B_BKG2	Float	Location of center of upper background extraction aperture
HEIGHT	Integer	Height of the spectral extraction window
BHEIGHT	Long	Height of background spectral extraction aperture
BWIDTH	Long	Width of the boxcar filter used to smooth the backgrounds
LOWER_OUTER	Float	Fraction of flux below lower outer boundary
UPPER_OUTER	Float	Fraction of flux below upper outer boundary
LOWER_INNER	Float	Fraction of flux below lower inner boundary
UPPER_INNER	Float	Fraction of flux below upper inner boundary
YERRMAX	Float	Maximum allowed error in centroid
PEDIGREE	String	Pedigree

3.7.23 SPWCSTAB: Spectroscopic WCS Parameters Table

- File Suffix: `_spwcs`

The spectroscopic SPWCS table gives the parameters needed to populate the world coordinate keywords in the `corrtag`, `counts`, and `flt` files. There are entries for each `SEGMENT`, `OPT_ELEM`, `CENWAVE`, and `APERTURE` . The columns (see Table 3.20) are interpreted as follows. The detector coordinate system has two dimensions. Let the more rapidly varying axis be X and the less rapidly varying axis Y. The world coordinate system has three dimensions, the spectral coordinate, right ascension, and declination. The reference pixel is at approximately the middle of the detector. `CTYPE1` can be `WAVE` to indicate that the wavelength is a linear function of pixel number, or it can be `WAVE-GRI` to indicate that the wavelengths should be computed by using the grating ("grism") equation. In either case, the wavelengths are in vacuum. `CRVAL1` is the wavelength at the reference pixel. `CRPIX1` is the location of the reference pixel in the first axis (X); the location of the reference pixel in the second axis (Y) is obtained separately from the 1-D Extraction Parameters Table (`XTRACTAB`). `CDELT1` is the dispersion in Angstroms per pixel at the reference pixel. At a single wavelength (nominally the wavelength at the reference pixel), a pixel when projected onto the sky would be approximately a rectangle. `CDELT2` and `CDELT3` are the sizes of that rectangle in the X and Y directions. `SPECRES` is the spectral resolution; this is only used for updating the archive search keyword of the same name. `G` is the groove density of the grating, e.g., 3.8E6 grooves per meter for G130M. `SPORDER` is the spectral order. This will usually be 1, but for G230L, stripe NUVC, `SPORDER` will be 2. `ALPHA` is the angle between the normal to the grating and the light that is incident onto the grating. `THETA` is the angle between two lines from the grating to the detector, the line to the reference pixel and the line that is perpendicular to the detector. Since the reference pixel is close to the middle of the detector, `THETA` will probably be close to zero.

Table 3.20: SPWCSTAB Table Format.

Column Name	Data Type	Description
SEGMENT	String	Segment Name
OPT_ELEM	String	Name of optical element
CENWAVE	Integer	Central wavelength (Angstroms)
APERTURE	String	NPSA. BOA, WCA
CTYPE1	String	Type of world coordinate on spectral axis
CRPIX1	Double	Reference pixel number for spectral axis (X)
CRVAL1	Double	Wavelength at the reference pixel (Ang)
CDELT1	Double	dispersion at reference pixel (Ang/pixel)
CDELT2	Double	Size of a pixel in dispersion direction (deg/pixel)
CDELT3	Double	Size of a pixel perpendicular to dispersion direction (deg/pixel)
SPECRES	Double	Spectral resolution
G	Double	Groove density of grating (grooves/m)
SPORDER	Integer	Spectral order
ALPHA	Double	Incident angle from aperture onto grating (degrees)

THETA	Double	Angle from reference pixel to base of normal from grating to detector (degrees)

Chapter 4: COS Error Sources

Chapter Contents

4.1 Overview

In this chapter we describe various properties of the COS detectors that affect the final quality of the science data products, in particular the various sources of error affecting the accuracy of the flux-calibrated and wavelength-calibrated spectra. Note that several of the effects outlined below are the subjects of ongoing study.

✓ ***Always check the COS Web pages for the latest Instrument Science Reports (ISRs) which describe the results of various studies intended to better characterize COS.***

4.2 Error Sources Associated with Pipeline Processing Steps

In this section, we discuss sources of error that are associated with major steps in the COS calibration pipeline (**calcos**). Note that these steps themselves were already described in Chapter 3 and will not be repeated here; this section will only describe specific issues related to the error budget of the resulting data which were not described before.

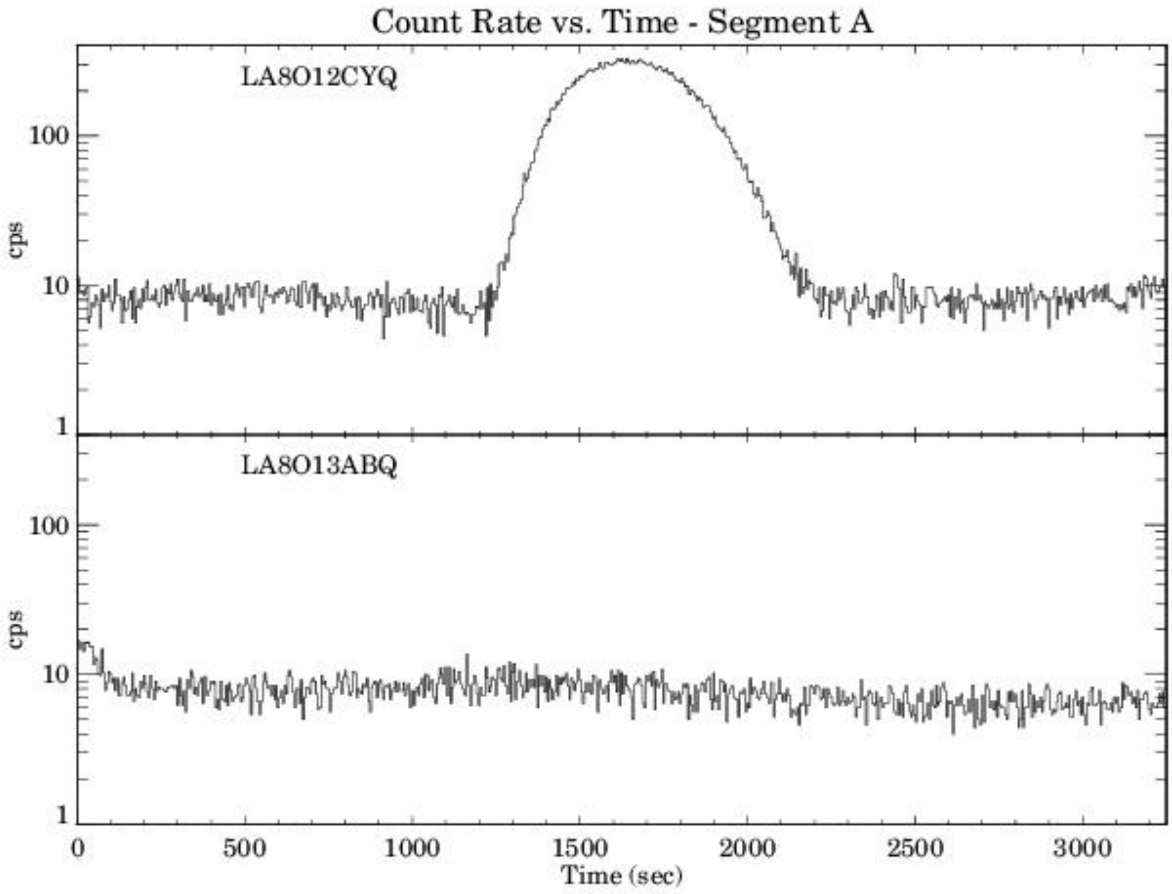
4.2.1 FUV Dark Count Rate

Dark counts arise from a combination of detector effects and external sources. **Calcos** will remove the effects of detector background (which includes dark, scattered light, etc.) in the BACKCORR module. This is done after the X1DCORR converts the detector image to a 1D extracted spectrum. Here, we discuss the instrumental contribution, since it can be the limiting factor in the error budget for very faint sources.

FUV-XDL Dark Count Rate

The FUV detector dark rates measured on the ground were very low, of order 0.4 counts cm^{-2} s^{-1}. Typical dark rates on-orbit away from the South Atlantic Anomaly (SAA) are several times higher. The dark rates based on the 95th percentile of dark values obtained in 2021 are 2.01×10^{-6} counts $pixel^{-1}$ s^{-1} for FUVA (corresponding to 1.40 counts cm^{-2} s^{-1}) and 1.70×10^{-6} counts $pixel^{-1}$ s^{-1} for FUVB (1.18 counts cm^{-2} $s^{-1)}$. This is equivalent to 1.21×10^{-4} counts s^{-1} per resolution element (FUVA) and 1.02×10^{-4} counts s^{-1} per resolution element (FUVB). These rates have remained relatively stable since SM4, although there are sometimes large variations seen on short timescales. The behavior of the dark rates versus time can be seen on these monitoring pages for FUVA and FUVB. For the most up to date numbers, please refer to the current version of the *COS Instrument Handbook*, Section 7.4.1.

Figure 4.1: Dark Rates.



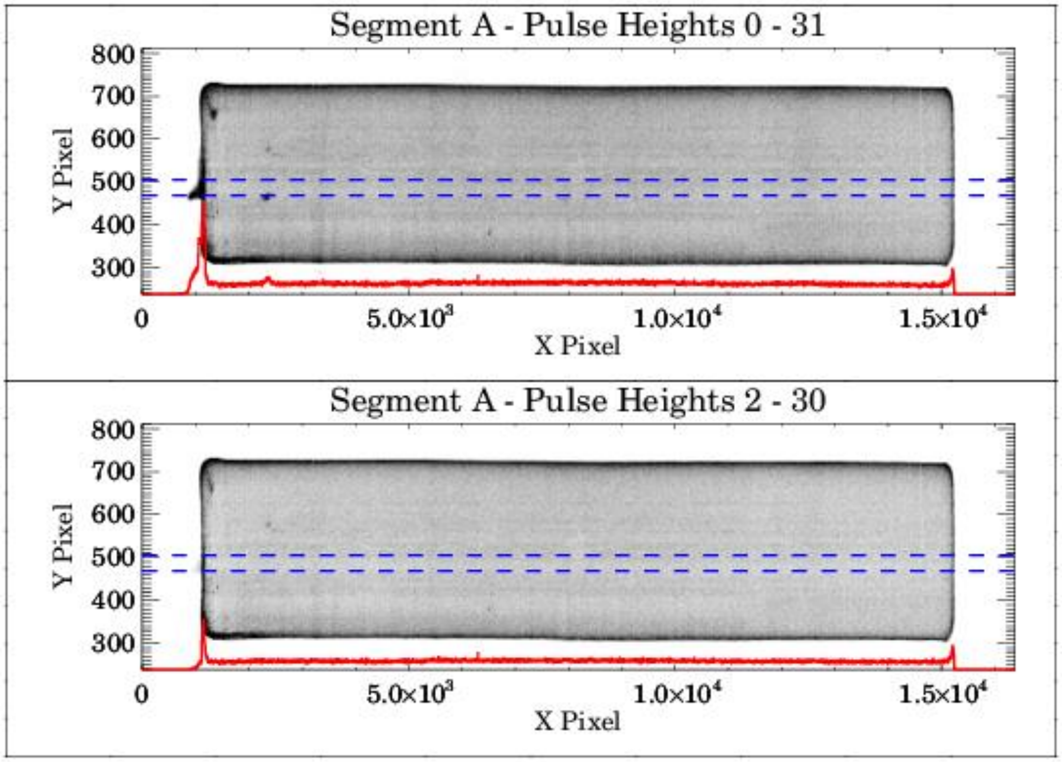
FUV Segment A count rate as a function of time during an orbit which skims the SAA (top), and during one which is further from the SAA (bottom).

When *HST* passes though the SAA (South Atlantic Anomaly, a region of high particle background), observations stop and the detector high voltage is lowered in order to prevent damage to the detector. Elevated dark rates of up to 30 times higher than normal (Figure 4.1) have been observed during exposures made when skimming the edge of the SAA. To minimize the observing time with higher background, the SAA model was shifted 6 degrees to the west in May 2010.

The spatial distribution of background counts on Segment A is fairly uniform, independent of pulse height thresholding or proximity to the SAA (Figure 4.2). For segment B, however, there are a number of features in the region where the spectra fall when all pulse heights are included. Most of these features disappear when the appropriate pulse height thresholding (used by default in the calcos pipeline for TIME-TAG data) is applied, as shown in Figure 4.3.

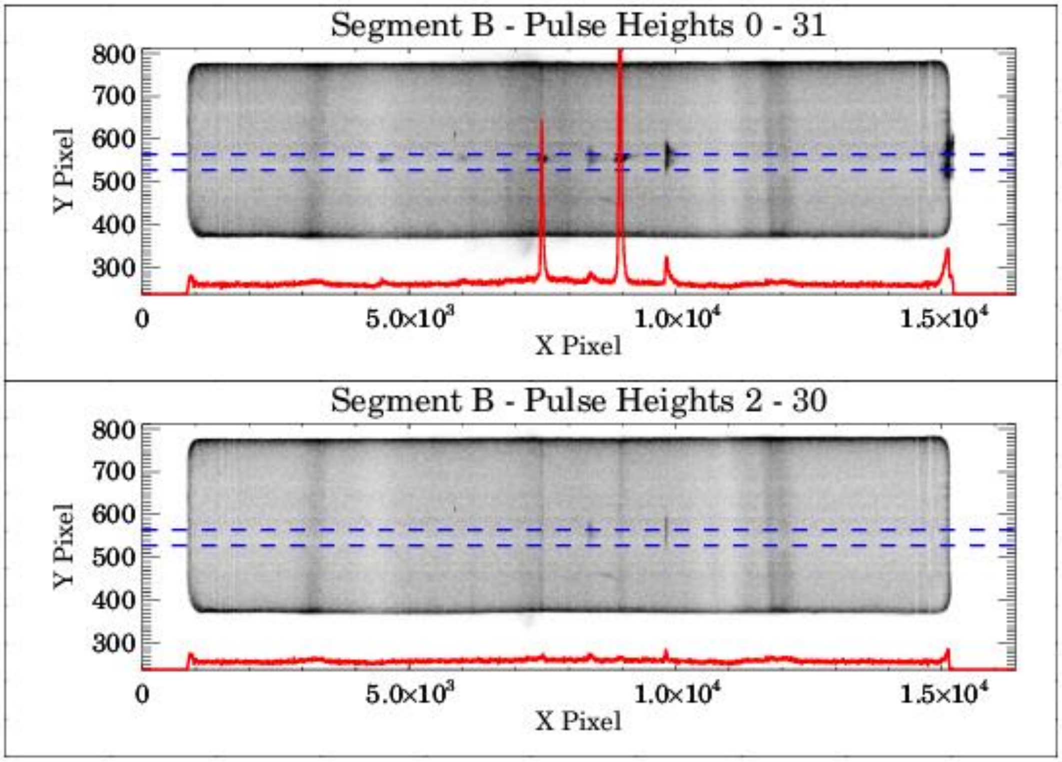
There is an additional complication to the FUV dark correction. As the FUV detectors have been exposed to more light, the portion of the detectors where the spectrum falls has become less sensitive due to gain sag. This sensitivity loss even affects the dark count rate. As a result, the background, which is estimated from rarely illuminated regions on either side of the science spectrum, tends to overestimate the dark rate at the location of the spectrum and, therefore, to overcorrect the spectrum. This effect is small, and only affects very faint objects. Nevertheless, one should be aware of it. We note that the darks are time and LP dependent.

Figure 4.2: FUVA Dark.



Dark rate for FUV Segment A with no pulse height thresholding (top), and with the default thresholding used by calcos (bottom). The background is spatially uniform at all pulse heights. The dashed blue lines show the LP1 extraction box and the red lines show the dark rate summed over the region between the blue lines.

Figure 4.3: FUVB Dark.



Dark rate for FUV Segment B with no pulse height thresholding (top), and with the more aggressive thresholding used by calcos (bottom). Using the appropriate thresholding minimizes the effects of the extra features near the middle of the segment. The dashed blue lines show the LP1 extraction box and the red lines show the dark rate summed over the region between the blue lines.

4.2.2 Flat Fields

NUV-MAMA Flat Fields

The STIS MAMA flat fields are dominated by a fixed pattern that is a combination of several effects including "beating" between the micro-channel plates and the anode pixel array and variations in the charge cloud structure at the anode. Similar effects are present in the COS MAMA. Intrinsic pixel-to-pixel variations measured on the ground for the COS NUV-MAMA were 5.2% rms. Analysis of the COS NUV flat-field taken during servicing mission orbital verification (SMOVs) by Ake et al. (COS ISR 2010-03) found that it aligned to within one pixel of the flat field created during ground testing. Consequently, all SMOV and ground data were combined to produce a single flat field reference file for pipeline processing.

The reference file does not correct vignetting, which affects X pixels with values between 0 and 200. The vignetting can eliminate as much as 20% of the flux from X = 0 to 100, and then the fraction of flux vignetted slowly decreases to 0 between X = 100 and 200. Since the amount of vignetting depends on the angle of illumination, and because the OSM positions are not perfectly repeatable, simple corrections are inadequate. Due to the low usage of the NUV channel, a more complex solution has not been pursued.

Studies of the on-orbit S/N achievable indicate that the Poisson limit can be reached for S/N < 70 and that an S/N > 150 can be achieved by combining high S/N exposures obtained at different FP-POS settings over most of the detector. However, the variable vignetting can introduce large, spatially coherent errors over the first 200 pixels of each stripe of the NUV spectra.

FUV-XDL Flat Fields

The FUV XDL detector has considerable fixed-pattern noise. These include dead spots, variable hot spots, a honeycomb pattern due to the manufacturing process used to produce the MCP, and shadows from the repeller grid wires. A full, two-dimensional flat field obtained during internal ground tests did not produce the signal-to-noise needed for a useful flat, and it has been deemed too costly in terms of exposure time and impact on detector lifetime to fully characterize the COS flat field using on-orbit observations.

Nevertheless, some progress has been made. The FUV flat field corrects for the grid wire shadows, which are the largest single source of fixed-pattern noise, some features near the edges of segment B (the so-called "imposter" grid wires), and low-order variations in the detector response.

Note that even with the correction of the grid wire shadows, other large amplitude (up to 10%) fixed-pattern features remain in the spectra. At present, the best approach to mitigate these is to combine observations obtained at different FP-POS settings. A complete description of the G130M and G160M grid wire flats, and estimates for the achievable S/N for these gratings from normally processed data, are given in COS ISR 2011-03.

4.2.3 Gain Sag

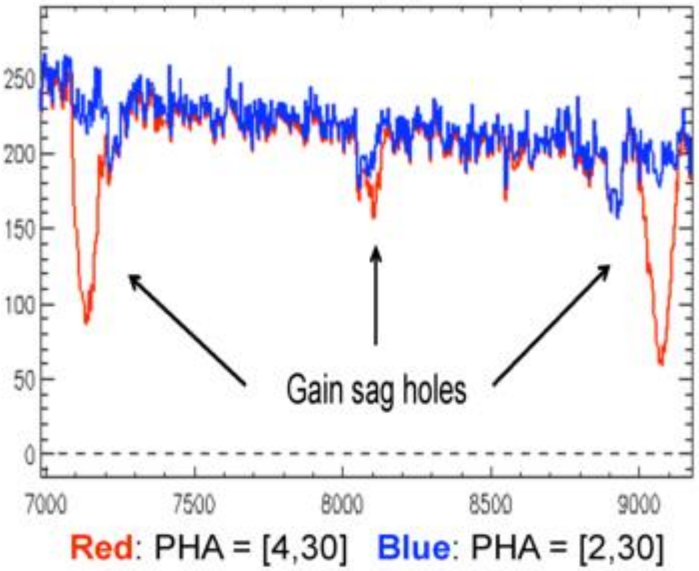
As described in Chapter 1, the pulse height for photons incident on the COS XDL FUV detector varies with position on the detector and changes with time. The typical peak pulse height (modal gain) decreases as the total exposure at that location increases; this effect is known as gain sag (see COS ISR 2011-05). The regions of the detector which have collected the most photons are the parts of Segment B that are exposed to the bright Lyα airglow line while using the G130M grating. In particular, regions near pixels 7150 and 9100 are most strongly affected. When the sag is large, PHA values can fall below the **calcos** thresholds and are discarded by the pipeline. Figure 4.4 shows the effect of changing the lower PHA cutoff from 4 to 2 on these features. With a PHA cutoff of 4, the total number of counts in the most heavily sagged regions is depressed by nearly 50%. In contrast, with a PHA cutoff of 2, the gain sag regions are depressed by approximately 10%.

The MCP gain can be increased by raising the high voltage on the affected detector segment. This is done at regular time intervals in order to keep the loss of flux due to gain sag to within data quality requirements. Nominal values of detector high voltage used on orbit are listed in Appendix B.

The gain in the spectral region can also be increased by moving the spectra to a part of the detector which has seen fewer counts. These moves of the Lifetime Position (LP) have been made several times since SMOV. In addition, the COS2025 policy limits the cenwave and FP-POS selections for G130M segment B observations. See Appendix A for details.

⚠ *In order to address the effects of gain sag, the COS FUV detector high voltage has been raised numerous times since the instrument was installed, and the spectra have been shifted to new lifetime positions. For details on these adjustments consult Appendix A and Appendix B.*

Figure 4.4: Gain Sag Effects.



This figure shows two versions of the same NET spectrum. The data displayed were taken in December 2010 at LP1 with the G160M grating. Red eliminates events with PHAs less than 4 and blue eliminates events with PHAs less than 2.

4.2.4 FUV XDL Thermal Drifts

The XDL centroiding electronics are sensitive to thermal effects. The `TEMPCORR` module of **calcos** measures the location of the stim pulses in order to determine the shift and stretch of the detector format and correct for any changes; `TEMPCORR` applies a linear correction based on the position of these stim pulses. The accuracy of this correction will influence the ability to properly register the flat field corrections and may influence the final error budget.

4.3 Factors Limiting Flux and Wavelength Accuracy

The COS calibration accuracies are given in Table 1.3 of the COS Instrument Handbook. In this section, several factors limiting those accuracies are discussed.

4.3.1 Flux Accuracy

The accuracy of the absolute flux calibration of COS spectroscopic data is limited by several factors including:

- The presence of fixed pattern noise in the FUV detectors. Although the grid wire shadows are corrected, several other artifacts remain, some of them with amplitudes up to 10%. Because the flux calibration is intended to be a smooth function, it interpolates through such small scale features, which can result in localized errors of ±5% (see COS ISR 2010-02 and COS ISR 2011-03).
- For the G140L settings, the initial wavelength calibrations were not as accurate as they are now. This, coupled with the fact that the instrumental response drops rapidly below about 1200 Angstroms, and that individual FUVB observations may not be properly aligned (see below), results in rather large flux calibration uncertainties (5–10%) below 1200 Angstroms for G140L data.
- The time-dependent sensitivity (TDS) correction to the FUV flux calibration may not be exact because it is determined empirically from available calibration observations and is determined after-the-fact as new calibration observations become available. The FUV sensitivity varies with time, and the rate depends on the grating, segment, and wavelength region considered. Based on our long baseline for COS, the degradation is likely dominated at early times by an outgassing product and at later times by atomic oxygen in the residual Earth atmosphere at *HST* altitude. Regular monitoring of the time-dependent sensitivity captures changes in the sensitivity due to varying atmospheric conditions stemming from variations in the solar cycle. The decline at early times is characterized in COS ISR 2011-02; the complex behavior at early times may add an additional small error to the calibration of data early in the mission. Information about the current time-dependent sensitivity monitoring and modeling is described in COS ISR 2022-08.
- Due to on-board Doppler corrections, a given pixel in `ACCUM` data will contain data from nearby pixels, which will cause a slight smearing of the fixed pattern noise.
- Because no PHA filtering is done onboard, FUV ACCUMs include events for all PHA values. This has two minor effects. First, background counts are included. However, since objects observed in ACCUM mode are bright, this should not be a practical issue. Second, because the absolute flux calibrations are derived from PHA filtered TIME-TAG data, this can result in small, systematic effects in the flux calibration, but these should be less than a few percent. In addition, ACCUM data do not include walk corrections.
- Both of the COS low resolution gratings are affected by order overlaps. For the NUV G230L, wavelengths longer than about 3200 Å (which primarily affects the NUVB stripe of CENWAVE 3360), second order light from wavelengths longer than 1600 Å can contaminate the result. For the second order G230L spectra (stripe NUVC), first order light from wavelengths at twice the observed wavelength can affect the spectra (see COS ISR 2010-01). For the FUV G140L, spectra longward of about 2150 Å can be contaminated by an overlapping second order spectrum (see COS ISR 2010-02). The exact extent of the contamination depends on the SED of the object being observed.
- For low signal-to-noise observations (particularly for faint sources), the signal-to-noise is best determined by measuring the dispersion around the continuum. At high signal-to-noise, this will match what can be determined from the ratio of the flux and error arrays in the `x1dsum` files.

4.3.2 Wavelength and Spectral Resolution Accuracies

There are several issues that may affect the COS wavelength calibration and spectral resolution, and these are explained in detail in COS ISRs 2010-05, 2010-06, 2009-01, and 2010-09. Some of these issues are outlined here.

- Because the COS optics corrects the *HST* spherical aberration after light passes through its large aperture, it accepts all of the uncorrected light from the *HST* telescope beam. Consequently, its image quality is subject to mid-range polishing errors which create broad wings on the PSF (see COS ISR 2009-01). Other spectrographs such as STIS can eliminate the effects of these wings by inserting a small aperture into the beam. Because COS cannot do that, its spectral purity is affected by the wings.
- Small, localized deviations from the dispersion relations determined by a low order polynomial have been reported for FUV XDL data. These deviations most probably result from localized inaccuracies in the geometric correction.
- For the FUVB segment of the G140L CENWAVE = 1280 setting, the wavecal lamp does not have detectable lines. As a result, the wavelength calibration from the FUVA side is applied to FUVB. However, for some observations, the FUVA is turned off, to avoid an over-bright condition. In these cases, a default wavelength calibration is applied. Note that the wavecal not only affects the wavelength calibration itself, but also the determination of where the PSA or BOA spectrum is expected to be. These same comments apply to FUVB observations obtained with the G130M CENWAVE = 1055 and 1096 settings. For more information, see section 5.7.5 of the *COS Instrument Handbook*.
- OSM motions, or drifts, can cause the spectrum to shift in the dispersion direction by as much as 2–3 pixels (~1 resolution element for NUV, approximately one-half resolution element for FUV) in the first 20 minutes after an OSM is moved. `TAGFLASH` wavecals correct for these motions to accuracies ≤0.5 pixel. However, it is only possible to correct `ACCUM` data for the mean OSM motion that occurred during the exposure and, in rare circumstances, this may result in a slight degradation in the spectral resolution of `ACCUM` data.
- The accuracy to which the source is centered in the science aperture along the dispersion direction can result in small displacements in the absolute wavelength scale corresponding to the plate-scales of 0.22 arcsec per FUV pixel and 0.25 arcsec per NUV pixel. Measurements for `ACQ/IMAGE` centering accuracies are of the order of 0.05 arcsec, and accuracies of other types of COS acquisition can be of the order of 0.1 arcsec or more. One can calculate the resulting wavelength accuracy using the plate-scale and dispersion given in Table 1.4 and Table 1.1 respectively.
- As discussed in the *COS Instrument Handbook*, the BOA degrades the target image, resulting in a reduction of the spectral resolution by a factor of three or more.
- Exposures longer than 960s taken at LP6 with SPLIT wavecals have a shift applied to the wavecal calibration to account for the fact that wavecals cannot be taken at the same time as the science observations. This shift makes the wavelength calibration more accurate than a straight line time interpolation between the SPLIT wavecals, and is estimated to result in an increase of < 0.5 pixel in the uncertainty of the wavelength calibration.

Chapter 5: COS Data Analysis

Chapter Contents

5.1 Data Reduction and Analysis Applications

Most of the software tools for operating on COS FITS files are contained in three **Python** packages that come with the **stenv** python distribution: https://stenv.readthedocs.io

The three **Python** packages used with COS are:

- **calcos**: Contains the calcos package, used for processing data through the calibration pipeline. Section 3.6 describes how to run **calcos**.
- **costools**: Contains COS-specific tasks to calibrate and interpret data. Many of these tasks are described in Chapter 3. A complete listing is presented in Section 5.1.1.
- **astropy**: Contains packages and modules to read and operate on FITS tables, which is the format used for many COS products including 1-D spectra and most COS reference files. We provide specific examples of their use here in Section 5.1.2.

We also present a brief summary of spectral display and analysis tasks in Section 5.1.3. To run the examples in Sections 5.1.1 to 5.4, one must activate the **stenv** environment before starting Python.

In addition to what is below, users are encouraged to use the Jupyter Notebooks for COS. Those available include:

Name	Topic
Setup	Setting up your computer environment to work with COS data
DataDl	Downloading COS Data from the archive
ViewData	Beginning to work with COS data in Python: *plotting, binning, calculating SNR, & evaluating* a spectrum
AsnFile	Modifying or creating an association file
CalCOS	Running the COS pipeline
DayNight	Filtering out COS data taken during the day or night
LSF	Working with the COS Line Spread Function (LSF)
Extract	Editing the extraction boxes in a BOXCAR-method spectral extraction file (XTRACTAB)

5.1.1 COS-Specific Tasks

In Chapter 3, we gave detailed discussions of the use of the data reduction pipeline **calcos**. This task is contained in the **PyPI** package, **calcos**. Other modules useful for reducing and analyzing COS data are contained in a separate package called **costools**. A complete listing and brief description of these modules is given in Table 5.1. All of these tasks are run in **Python**. Consult the online help for each module for more information. Some of these modules will be discussed in greater detail in the remainder of this chapter.

Table 5.1: COS-Specific Modules.

Module	Description	Package
calcos	Process COS data through the calibration pipeline	**calcos**
saamodel	Get vertices for SAA model by model number	**costools**
splittag	Split a `corrtag` file into multiple sub-files by time interval	**costools**
timefilter	Filter a corrtag table based on the TIMELINE extension	**costools**

5.1.2 FITS Table Tasks

COS 1-D spectra and most COS reference files are stored in FITS tables (see Section 2.4.1 for a description of the structure of the table extension files for 1-D spectra and `TIME-TAG` data). Tasks designed to handle this format can be used to examine and extract information from the tables. Here we give specific examples of the use of routines in **astropy.table.Table** and **astropy.io.fits** to help you get started. A sample output is given after each command.

Use the **Table** module in the **astropy.table** package to find out what information is given in the columns of a FITS table.

```
> from astropy.table import Table
> t = Table.read('ldel05k2q_x1d.fits')
> t.info # to print the column info
<Table length=2>
        name           dtype   shape            unit           format
-------------------- ------- ------- ---------------------- -------
             SEGMENT  bytes4
             EXPTIME float64                               s {:8.3f}
               NELEM   int32                                   {:6d}
          WAVELENGTH float64 (1274,)                Angstrom
                FLUX float32 (1274,)  erg / (Angstrom cm2 s)
               ERROR float32 (1274,)  erg / (Angstrom cm2 s)
         ERROR_LOWER float32 (1274,)  erg / (Angstrom cm2 s)
       VARIANCE_FLAT float32 (1274,)
     VARIANCE_COUNTS float32 (1274,)
        VARIANCE_BKG float32 (1274,)
               GROSS float32 (1274,)                  ct / s
             GCOUNTS float32 (1274,)                      ct
                 NET float32 (1274,)                  ct / s
          BACKGROUND float32 (1274,)                  ct / s
                  DQ   int16 (1274,)
              DQ_WGT float32 (1274,)
            DQ_OUTER   int16 (1274,)
BACKGROUND_PER_PIXEL float32 (1274,)            ct / (pix s)
    NUM_EXTRACT_ROWS   int16 (1274,)
           ACTUAL_EE float64 (1274,)
       Y_LOWER_OUTER float64 (1274,)
       Y_UPPER_OUTER float64 (1274,)
       Y_LOWER_INNER float64 (1274,)
       Y_UPPER_INNER float64 (1274,)
```

Use **print()** to look at the contents of the table:

```
> from astropy.table import Table
> t = Table.read('ldel05k2q_x1d.fits')
> print(t)
SEGMENT EXPTIME ... Y_UPPER_INNER [16384]
           s    ...
------- ------- ... ---------------------
   FUVA 503.744 ...        421.0 .. 423.0
```

Note that the number of columns displayed is limited by the width of the window that you are working in. To see more columns, you can adjust your window to be wider and re-type the command:

```
> print(t)
SEGMENT EXPTIME NELEM ... Y_UPPER_OUTER [16384] Y_LOWER_INNER [16384] Y_UPPER_INNER [16384]
           s          ...
------- ------- ----- ... --------------------- --------------------- ---------------------
   FUVA 503.744 16384 ...        427.0 .. 430.0        414.0 .. 410.0        421.0 .. 423.0
```

The table output indicates that some of the columns contain arrays of 16384 elements rather than a single value. For those columns, printing the table element t displays the value of the first and last elements in the array.

The user may wish to look at values in a COS reference file `FITS` table, which generally have many rows. Each row typically characterizes a specific operating mode, location on the detector, a value of a parameter to be used in the reduction, etc.

```
> from astropy.io import fits
> from astropy.table import Table
> hdulist = fits.open('s7g1700hl_disp.fits')
> tbdata = hdulist[1].data
> print(Table(tbdata))
SEGMENT OPT_ELEM APERTURE CENWAVE NELEM       COEFF [4]
------- -------- -------- ------- ----- ---------------------
   FUVA G130M        PSA     1291     2 1272.64958352 .. -0.0
   FUVA G130M        WCA     1291     2 1272.21660012 .. -0.0
   FUVA G130M        BOA     1291     2 1272.64958352 .. -0.0
   FUVB G130M        PSA     1291     2 1119.38291962 .. -0.0
   FUVB G130M        WCA     1291     2 1118.98191865 .. -0.0
   FUVB G130M        BOA     1291     2 1119.38291962 .. -0.0
   FUVA G130M        PSA     1300     2 1282.84550637 .. -0.0
   FUVA G130M        WCA     1300     2 1282.41252297 .. -0.0
   FUVA G130M        BOA     1300     2 1282.84550637 .. -0.0
   FUVB G130M        PSA     1300     2 1129.58656362 .. -0.0
   FUVB G130M        WCA     1300     2 1129.18055776 .. -0.0
   FUVB G130M        BOA     1300     2 1129.58656362 .. -0.0
   ...
> hdulist.close()
```

In **Python**, there are often many methods to accomplish the same goal. For instance, in the above example, one could have used **fits.getdata()** instead of **fits.open()** (as shown in examples below). When using **fits.open()** , it is important to close the fits file afterwards as shown in the last line of this example.

5.1.3 General Spectral Display and Analysis Tasks

Table 5.2 lists several useful **Python** packages for displaying and analyzing COS spectral data.

Table 5.2: Spectral Analysis Tasks.

Task	Input Formats	Purpose
specviz	1-D fits or ASCII files	Generating 1-D plots, fitting curves to spectral features, and much more with a graphical user interface
matplotlib.pyplot.plot()	1-D arrays	Generating 1-D plots from columns in tables
matplotlib.pyplot.imshow()	2-D image arrays	Generating 2-D plots from 2-D image arrays
astropy.modeling	1-D and 2-D arrays	Fitting curves to spectral features
scipy.optimize.curve_fit()	1-D arrays	Fitting curves to spectral features

Specvis is a powerful spectral plotting and fitting package that provides the user with many options for modeling emission and absorption features, continuum, and much more. It is controlled via a graphical user interface, and has documentation and examples available: https://specviz.readthedocs.io/. Please note that this is a new package which has not been fully tested and verified as of May 2018.

Plotting COS Spectral Images (`flt`)

The COS FUV data consist of separate files for each of the two FUV detector segments, segment A and segment B. In general, each segment contains one target spectrum and one wavecal. On the other hand, for the NUV data there are three disjointed portions of the spectrum present on the image for both the target and the wavecal. The following examples illustrate the use of the **imshow()** function in **matplotlib.pyplot** module to plot COS FUV and NUV spectra from the 2-D `flt` files. The parameters in **plt.ylim()** refer to the Y location of the spectrum for each particular grating/segment combination. One may need to adjust the scaling factors, vmin and vmax, to suit the data range of any particular dataset. A range of different colormaps may be used, and can be found here:
https://matplotlib.org/tutorials/colors/colormaps.html.

The parameters in plt.ylim() refer to the Y location of the spectrum for each particular grating/segment combination. For the FUV detector, the Y location also depends on lifetime position (LP). The example below shows both segments of a lifetime position 4, G160M spectrum.

```
# target spectrum, FUV segment A:
> import matplotlib.pyplot as plt
> from astropy.io import fits
> fltdata = fits.getdata('ldel05ivq_flt_a.fits')
> plt.imshow(fltdata, cmap=plt.get_cmap('gist_yarg'), \
>          aspect='auto', vmin=0.0, vmax=0.2, origin='lower')
> plt.ylim([410., 430.])
> plt.xlabel('X FULL')
> plt.ylabel('Y FULL')
> plt.colorbar()
> plt.show()

# wavecal spectrum, FUV segment A:
> import matplotlib.pyplot as plt
> from astropy.io import fits
> fltdata = fits.getdata('ldel05ivq_flt_a.fits')
> plt.imshow(fltdata, cmap=plt.get_cmap('gist_yarg'), \
>         aspect='auto', vmin=0.0, vmax=0.01, origin='lower')
> plt.ylim([510., 535.])
> plt.xlabel('X FULL')
> plt.ylabel('Y FULL')
> plt.colorbar()
> plt.show()
```

```
# target spectrum, NUV stripes A, B, C:
> import matplotlib.pyplot as plt
> from astropy.io import fits
> fltdata = fits.getdata('lcwj10qtq_flt.fits')

# stripe A
> plt.imshow(fltdata, cmap=plt.get_cmap('gist_yarg'), \
>         aspect='auto', vmin=0.0, vmax=0.01, origin='lower')
> plt.xlabel('XFULL')
> plt.ylabel('YFULL')
> plt.ylim([175.0, 190.0])
> plt.colorbar()
> plt.show()

# stripe B
> plt.imshow(fltdata, cmap=plt.get_cmap('gist_yarg'), \
>      aspect='auto', vmin=0.0, vmax=0.01, origin='lower')
> plt.xlabel('XFULL')
> plt.ylabel('YFULL')
> plt.ylim([275.0, 290.0])
> plt.colorbar()
> plt.show()

# stripe C
> plt.imshow(fltdata, cmap=plt.get_cmap('gist_yarg'), \
>         aspect='auto', vmin=0.0, vmax=0.01, origin='lower')
> plt.xlabel('XFULL')
> plt.ylabel('YFULL')
> plt.ylim([420.0, 435.0])
> plt.colorbar()
> plt.show()
```

Plotting COS Tabular Spectra (`x1d`)

The following example illustrates the use of the **Python** package **matplotlib.pyplot** to plot COS tabular data. See Section 5.3.1 for more details and examples.

```
> import matplotlib.pyplot as plt
> from astropy.io import fits

> x1ddata = fits.getdata('ldel05k2q_x1d.fits')
> # select only segment A
> x1d_fuva = x1ddata[x1ddata['segment'] == 'FUVA']
> wavelength = x1d_fuva['wavelength'][0]
> flux = x1d_fuva['flux'][0]

> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

The above example selects the portion of the data that corresponds to the segment we want to plot. The segment selector can be either 'FUVA' or 'FUVB' for FUV data and 'NUVA', 'NUVB', or 'NUVC' for NUV data.

The next example shows how one would plot both FUV segments by using the **concatenate()** module in **numpy**.

```
> import matplotlib.pyplot as plt
> from astropy.io import fits
> import numpy as np
> x1ddata = fits.getdata('ldel02cxq_x1d.fits')
> wavelength = np.concatenate((x1ddata["wavelength"][1],\
              x1ddata["wavelength"][0]))
> flux = np.concatenate((x1ddata["flux"][1], \
         x1ddata["flux"][0]))
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

We note that instead of using **concatenate()**, one could also use the **flatten()** module as shown in the example in Section 5.3.1.

5.2 Evaluating Target Acquisitions and Guiding

COS target acquisitions and the options available to the observer are fully described in the *COS Instrument Handbook*. If you are examining COS observations that were specified by another observer, please refer to the instrument handbook to understand the options and parameters that may have been used.

Virtually all COS observations start with one or more acquisition exposures. The purpose of the acquisition is to ensure that the object observed is well centered in the COS aperture being used so as to avoid throughput losses and to produce a reliable wavelength zero point. Examining the acquisition data should allow you to detect significant errors in the centering of the target. Note that target acquisition data are always uncalibrated.

5.2.1 Types of Target Acquisitions

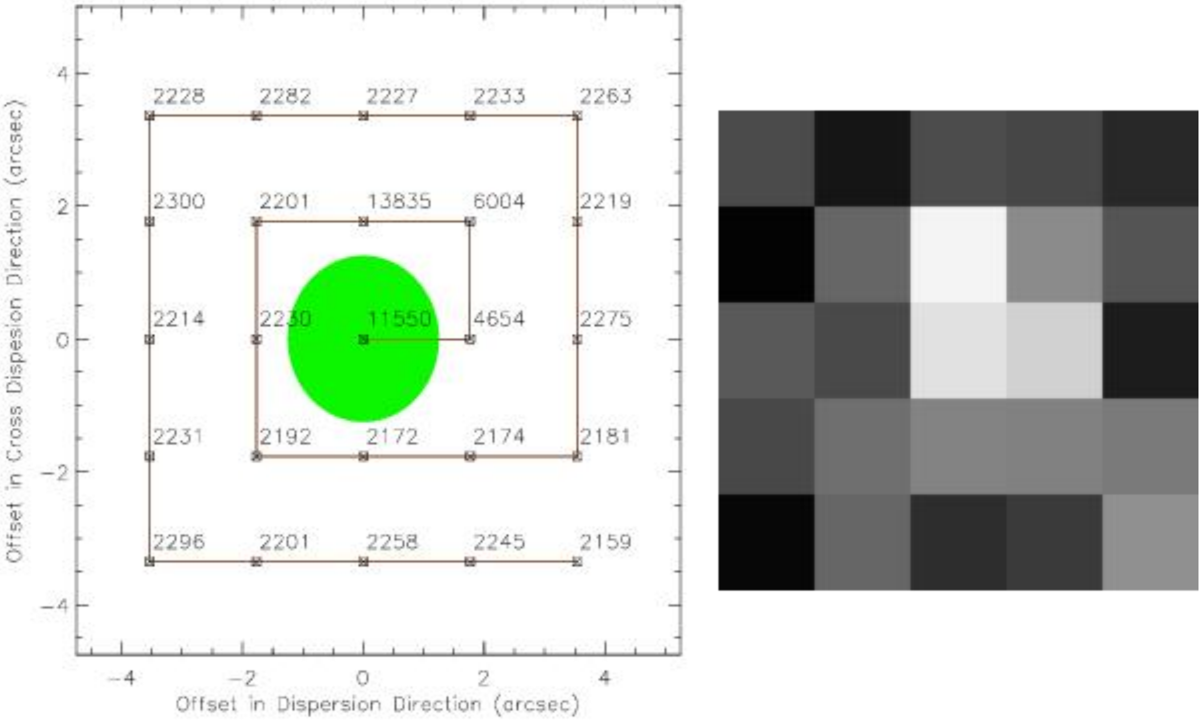
There are two types of COS acquisitions: imaging and dispersed-light. In an imaging acquisition, the COS NUV channel is used to obtain an image of the target in the COS aperture. This image is then analyzed by the COS flight software, the object's centroid is calculated, and the object is centered in the aperture. A dispersed-light acquisition directly analyzes the spectrum of the object being acquired and determines how to center the object so as to maximize throughput. Both types of COS acquisitions are intended to work on point sources or point-like sources.

`ACQ/SEARCH`

A sequence of observations may begin with an `ACQ/SEARCH`, either in imaging mode or in dispersed-light mode (see the *COS Instrument Handbook* for a full description). The optical element selected will appear in the ACQ file header: either a grating (and central wavelength) for dispersed light, or a mirror (`MIRRORA` or `MIRRORB`) for imaging. In either case, the telescope is commanded to move in a square spiral pattern, and at each dwell point an exposure is taken. The `STEP-SIZE` parameter sets the spacing between dwell points; the default is 1.767 arcsec, the optimum size to ensure that no area of sky is missed. The `SCAN-SIZE` parameter sets the number of dwells on each side of the square; the choices are 2, 3, 4, or 5. If an even number of points is used (`SCAN-SIZE` = 2 or 4), the first point is offset by half the `STEP-SIZE` in both directions so that the overall pattern remains centered on the initial pointing.

The data from an `ACQ/SEARCH` exposure consists of a header and a binary table data extension which contains the accumulated counts at each dwell point, see Table 2.11. This array of counts was processed by the flight software to calculate the centroid and the telescope was then commanded to move to that centroid. A *quick verification* that an `ACQ/SEARCH` exposure was successful would be to find the values of the `XDISP_OFFSET` and `DISP_OFFSET` columns of the `ACQ/SEARCH` data table corresponding to the maximum counts value at a single dwell point. Then, compare the `XDISP_OFFSET` and `DISP_OFFSET` values to the `ACQSLEWX` and `ACQSLEWY` header keyword values (see Table 2.7). Similarly, the data can be easily plotted for quick visual verification (see Figure 5.1).

Figure 5.1: Example of an ACQ/SEARCH exposure.



Left: An example of an `ACQ/SEARCH` spiral pattern showing the offsets in the dispersion and cross dispersion directions from the initial pointing for each dwell point with the counts at each dwell point overplotted. The green circle in the center represents the science aperture and the initial pointing. Right: Linearly scaled image of the counts at each dwell point for the same 5 × 5 `ACQ/SEARCH` exposure (where white shows the most counts).

Undispersed Light (Imaging) Acquisitions (`ACQ/IMAGE`)

When the `ACQ/IMAGE` command is used, two `ACCUM` exposures in imaging mode are taken for the specified exposure time, using the NUV channel of COS. The first exposure is taken after the initial pointing by *HST* and is used by the flight software to determine the centroid of the object and the amount of pointing change needed to center the object in the aperture (`PSA` or `BOA`). The center of the aperture is computed from the image of the Wavelength Calibration Aperture (`WCA`) and the known `WCA-PSA` offset specified on the flight software. The second image is taken after the object is centered to confirm that proper centering occurred. Each of the two images uses a sub-array of size 816 × 345 (in user coordinates) on the COS NUV MAMA. The commanded motions of the telescope in x and y are provided in the `ACQ/IMAGE` header. The `_rawacq` file contains the initial target image as a 1024 × 1024 array, followed by the confirmation image, another array of the same size.

The appearance of the image of a point source recorded by COS in `ACQ/IMAGE` mode will depend on the aperture used (`PSA` or `BOA`) and the mirror (`MIRRORA` or `MIRRORB`). The best optical quality is achieved with the `PSA` used with `MIRRORA`, in which case a diffraction-limited image is created with a tight core. If `MIRRORB` was used instead to attenuate the source, two images of the source are produced (Figure 5.2). If the `BOA` was used, a neutral-density filter attenuates the source, but that filter has a slight wedge shape that degrades optical quality. Figure 5.3 shows images of point sources obtained with the `BOA` using `MIRRORA` and `MIRRORB`. Profiles of images taken with various combinations of (`PSA`, `BOA`) and (`MIRRORA`, `MIRRORB`) are shown in the *COS Instrument Handbook*.

The data produced by `ACQ/IMAGE` can be used to confirm proper acquisition of an object, by direct comparison of the two images.

Figure 5.2: Example of an image using the **`PSA`** and **`MIRRORB`**.

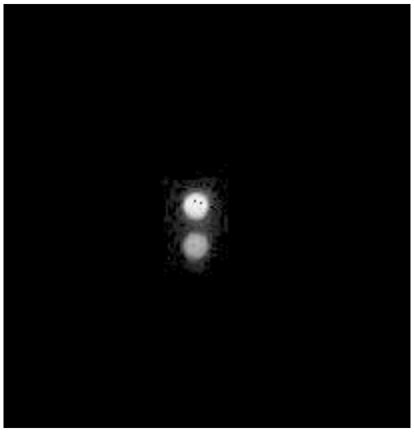
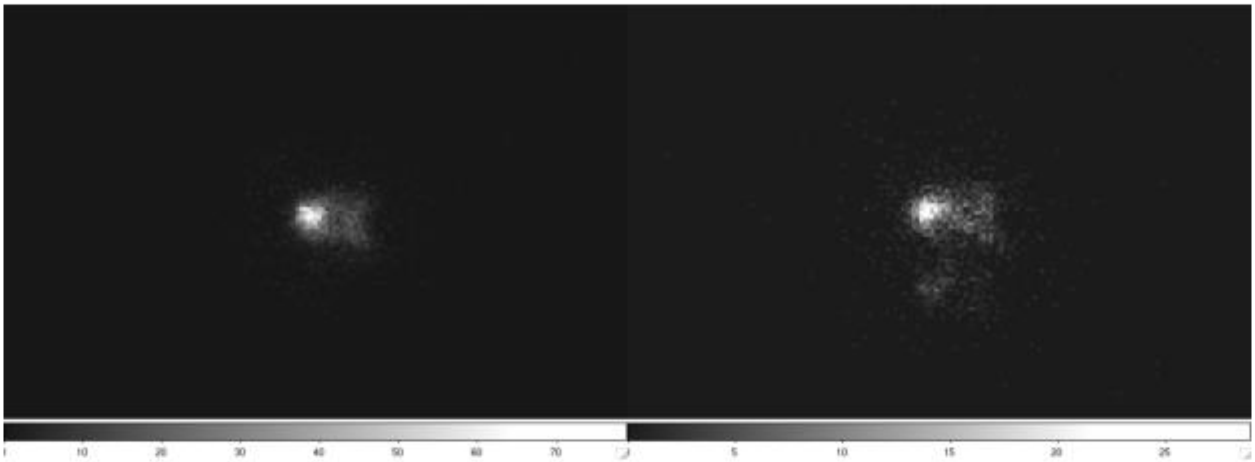


Figure 5.3: `BOA` and `MIRRORA` (left), and the `BOA` and `MIRRORB` (right).



Example of an image using the `BOA` and `MIRRORA` (left), and the `BOA` and `MIRRORB` (right).

Dispersed-Light Acquisitions (`ACQ/PEAKD` and `ACQ/PEAKXD`)

As noted above, an `ACQ/SEARCH` exposure can be performed in dispersed light. In that case, the file header will show a grating and central wavelength for the optical element chosen. As for an `ACQ/IMAGE`, any acquisition performed in dispersed light can use either aperture: the `PSA` or `BOA`. In addition to `ACQ/SEARCH`, two other commands are available to improve the centering of an object in dispersed light: `ACQ/PEAKXD` and `ACQ/PEAKD`.

An `ACQ/PEAKXD` should always precede an `ACQ/PEAKD` if both are performed. An `ACQ/PEAKXD` centers the spectrum in the cross-dispersion direction by obtaining a short exposure, calculating the centroid, and moving the telescope by that amount. Users will only receive files with headers containing the commanded movement of the telescope for `ACQ/PEAKXD` exposures. A *quick verification* that an `ACQ/PEAKXD` exposure was successful would be to compare the `ACQSLEWY` and (`ACQPREFY` –`ACQMEASY`) header keyword values (see Table 2.7).

An `ACQ/PEAKD` centers the spectrum along the dispersion direction by executing a series of short exposures with the telescope moving the source in a line for a specified number of points (`SCAN-SIZE`), spaced by `STEP-SIZE` arcsec (effectively a 1-D `ACQ/SEARCH`). A centroid is calculated, and the same options available for an `ACQ/SEARCH` are also available for an `ACQ/PEAKD`. Following the centroid calculation, the telescope is moved to center the source, and the counts at each dwell point are recorded in a table, see Table 2.11. Users may compare the offsets associated with the dwell point containing the maximum counts to the telescope slews recorded in the header. A *quick verification* that an `ACQ/PEAKD` exposure was successful would be to find the value of the `DISP_OFFSET` column of the `ACQ/PEAKD` data table corresponding to the maximum counts value at a single dwell point. Then, compare the DISP_OFFSET value to the `ACQSLEWX` header keyword values (see Table 2.7). The data can also be easily plotted for a quick visual verification, similar to what is shown in Figure 5.1 for the `ACQ/SEARCH` example.

5.2.2 Guiding Errors for Single-Guide-Star Mode

Tracking on two guide stars should provide pointing accuracy sufficient to keep targets centered in the COS aperture for several orbits. However, in some cases, observations are made using only a single guide star instead of the usual two. Either the General Observer has consented to this in consultation with the Program Coordinator when two suitable guide stars could not be found, or one Fine Guidance Sensor failed to acquire its guide star during the guide star acquisition/reacquisition. In this situation, the roll of the telescope is under GYRO control, which may allow a slow drift of the target on a circular arc centered on the single guide star. The drift rate along this arc (rate of rotation) depends on the characteristics of the pointing for any particular observation, but typical values are expected to be in the range of 1.0 to 1.5 milliarcsec/sec (possibly, but very rarely, as large as 5 milliarcsec/sec).

To calculate the approximate magnitude of the drift of the target on the detector, you will need to find the distance of the target from the acquired guide star. The primary header of the observation log file `jif` identifies the acquired guide star (`GSD_ID`) and gives its right ascension (`GSD_RA`) and declination (`GSD_DEC`) in degrees. For example, for a target 10 arcmin from the guide star, a drift of the target around the guide star of 1 milliarcsec/sec during a 1,000 second exposure would cause the target to move 0.0029 arcsec on the detector. The direction of the motion on the detector can be deduced from header keywords in the science data describing the position angle of the detector (e.g., `PA_APER`) in combination with the direction perpendicular to the radiant. In many cases, the drift will be a small fraction of a pixel, although in some cases an image exposure may appear smeared.

5.3 Working with Extracted Spectra

Here we discuss how to customize the extraction of spectra and modify reference files that control the extraction process with **calcos**. Please note that STScI supports the **FITS** module located in the **Python** **astropy.io** package. This module provides the same core functionality as the earlier module **PyFITS**. Any script that is uses **PyFITS** can be modified by importing **FITS** into **Python** as follows:

```
from astropy.io import fits as pyfits
```

Information on the **FITS** module can be found on the astropy webpage.

5.3.1 Working with `x1d` Files in Python

When calibrating a single spectroscopic exposure, the **calcos** pipeline creates a one-dimensional extracted spectrum file, with suffix `x1d` and a filename such as "`l9v220eqq_x1d.fits`."

COS `x1d` files are MEF format files and their data contents and extension formats are discussed in Section 2.4.3. As with other COS data files, the primary [0] header will contain only header information and no data. The extracted spectra are stored in a single [SCI] extension as a *multi-dimensional* binary table. 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.

Using the "Selectors Syntax" to work with tables

In order to analyze COS tabular spectral data with most **Python** modules, one can use the selectors syntax to specify the desired row and column (e.g., the wavelength or flux). The general syntax for selecting a particular cell is to first select the desired segment, and then select the column:

```
> import matplotlib.pyplot as plt
> from astropy.io import fits
> x1ddata = fits.getdata('ldel02cxq_x1d.fits')
> # select segment A or segment B data
> x1d_fuva = x1ddata['segment'] == 'FUVA'
> x1d_fuvb = x1ddata['segment'] == 'FUVB'
> # select the wavelength and flux columns for each segment
> wave_fuva = x1ddata[x1d_fuva]['wavelength'].flatten()
> flux_fuva = x1ddata[x1d_fuva]['flux'].flatten()
> wave_fuvb = x1ddata[x1d_fuvb]['wavelength'].flatten()
> flux_fuvb = x1ddata[x1d_fuvb]['flux'].flatten()
> # plot segment A
> plt.plot(wave_fuva, flux_fuva)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

Similarly, one could use the **where()** module in **numpy** to select the segment:

```
> from astropy.io import fits
> from astropy.table import Table
> hdulist = fits.open('s7g1700hl_disp.fits')
> tbdata = hdulist[1].data
> print(Table(tbdata))
```

```
SEGMENT OPT_ELEM APERTURE CENWAVE NELEM       COEFF [4]
------- -------- -------- ------- ----- ---------------------
   FUVA G130M        PSA     1291     2 1272.64958352 .. -0.0
   FUVA G130M        WCA     1291     2 1272.21660012 .. -0.0
   FUVA G130M        BOA     1291     2 1272.64958352 .. -0.0
   FUVB G130M        PSA     1291     2 1119.38291962 .. -0.0
   FUVB G130M        WCA     1291     2 1118.98191865 .. -0.0
   FUVB G130M        BOA     1291     2 1119.38291962 .. -0.0
   FUVA G130M        PSA     1300     2 1282.84550637 .. -0.0
   FUVA G130M        WCA     1300     2 1282.41252297 .. -0.0
   FUVA G130M        BOA     1300     2 1282.84550637 .. -0.0
   FUVB G130M        PSA     1300     2 1129.58656362 .. -0.0
   FUVB G130M        WCA     1300     2 1129.18055776 .. -0.0
   FUVB G130M        BOA     1300     2 1129.58656362 .. -0.0
   ...
> desired_row = ((tbdata['SEGMENT'] == 'FUVB') & \
                        (tbdata['OPT_ELEM'] == 'G130M') & \
                        (tbdata['APERTURE'] == 'BOA') & \
                        (tbdata['CENWAVE'] == 1300))
> coeffs = tbdata[desired_row]['COEFF']
> print(coeffs)

array([[  1.12958656e+03,   9.96449962e-03, 0.00000000e+00, -0.00000000e+00]])

> hdulist.close()
```

Dumping x1d data to an ASCII File

It is possible to dump the arrays of an `x1d` file to an `ASCII` file. For example, to extract the WAVELENGTH, FLUX, ERROR, and DQ_WGT columns of FUV `x1d` file `ldel02cxq_x1d.fits`:

```
> from astropy.io import fits
> import numpy as np
> x1ddata = fits.getdata('ldel02cxq_x1d.fits')
> wavelength = np.concatenate((x1ddata["wavelength"][1],\
>              x1ddata["wavelength"][0]))
> flux = np.concatenate((x1ddata["flux"][1], x1ddata["flux"][0]))
> err = np.concatenate((x1ddata["error"][1], x1ddata["error"][0]))
> dq_wgt = np.concatenate((x1ddata["dq_wgt"][1], \
>         x1ddata["dq_wgt"][0]))
>with open('data.txt','w') as f:
>    for row in zip(wavelength, flux, err, dq_wgt):
>        f.write('{}\t{}\t{}\t{}\n'.format(row[0], row[1], row[2], row[3]))
```

This will create a new 2-D `ASCII` text file, `data.txt`.

Plotting COS `x1d` Data

Each row of the science extensions in an `x1d` file will contain the columns listed in Table 2.5; a similar table, including array dimensions, can be displayed by using the **Table** module in **astropy.table**.

When using many **Python** routines with `x1d` files as input, it will be necessary to specify the extension number of the file. For example, to plot flux vs. wavelength for both segments in an `x1d` file using the **plot** module in **matplotlib.pyplot**:

```
> import matplotlib.pyplot as plt
> from astropy.io import fits
> import numpy as np
> x1ddata = fits.getdata('ldel02cxq_x1d.fits')
> # Print segments associated with different indexes
> print(x1ddata["segment"][0], x1ddata["segment"][1])
> wavelength = np.concatenate((x1ddata["wavelength"][1],\
              x1ddata["wavelength"][0]))
> flux = np.concatenate((x1ddata["flux"][1], \
        x1ddata["flux"][0]))
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

In this example, we explicitly reference extension 0 and 1 to obtain segment FUVB and FUVA respectively. One can verify this by printing the segment names:

```
> # Print segments associated with different indexes
> print(x1ddata["segment"][0], x1ddata["segment"][1])
```

For FUV data, the `x1d` files contain both[1] segments A and B. To plot flux vs. wavelength in an FUV `x1d` file for segment A or segment B, using the **plot** module:

```
> import matplotlib.pyplot as plt
> from astropy.io import fits
> import numpy as np
> x1ddata = fits.getdata('ldel02cxq_x1d.fits')
# select only segment A
> x1d_fuva = x1ddata[x1ddata['segment'] == 'FUVA']
> wavelength = x1d_fuva['wavelength'].flatten()
> flux = x1d_fuva['flux'].flatten()
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
# select only segment B
> x1d_fuvb = x1ddata[x1ddata['segment'] == 'FUVB']
> wavelength = x1d_fuvb['wavelength'].flatten()
> flux = x1d_fuvb['flux'].flatten()
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

For NUV data, the `x1d` files contain the three stripes A, B, and C. To plot flux vs. wavelength in an NUV `x1d` file for all three stripes sequentially:

```
> import matplotlib.pyplot as plt
> from astropy.io import fits
> x1dfile = fits.getdata('lcwj10qtq_x1d.fits')
# select only stripe A
> x1d_nuva = x1ddata[x1ddata['segment'] == 'NUVA']
> wavelength = x1d_nuva['wavelength'].flatten()
> flux = x1d_nuva['flux'].flatten()
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()

# select only stripe B
> x1d_nuvb = x1ddata[x1ddata['segment'] == 'NUVB']
> wavelength = x1d_nuvb['wavelength'].flatten()
> flux = x1d_nuvb['flux'].flatten()
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()

# select only stripe C
> x1d_nuvc = x1ddata[x1ddata['segment'] == 'NUVC']
> wavelength = x1d_nuvc['wavelength'].flatten()
> flux = x1d_nuvc['flux'].flatten()
> plt.plot(wavelength, flux)
> plt.xlabel('Wavelength')
> plt.ylabel('Flux')
> plt.show()
```

5.3.2 Redoing Spectral Extraction

The **x1dcorr** module in **calcos** is designed to extract flux calibrated 1-D spectra from corrected COS event lists (`corrtag` files). This module is called by **calcos** as part of standard pipeline processing; its functioning in that role is described in Section 3.4.18.

Correcting for Shifts Along the Dispersion Direction

Properly aligning the spectrum along the dispersion direction is important not only for obtaining the correct wavelength solution, but also for properly applying the flux calibration. Incorrect registration of the spectrum will result in the wrong sensitivity being applied at each wavelength. This is especially important for low resolution spectra, since at some wavelengths the sensitivity changes rapidly with wavelength. The throughput also changes rapidly for the blue modes.

For `rawtag` exposures the wavecal lamp exposures are taken either concurrently with the science `rawtag` spectra (`TAGFLASH`) or they are acquired as separate `rawtag` spectra (`AUTO` or `GO` wavecals). For all science `rawaccum` exposures the wavecals are acquired as separate `rawtag` exposures.

The wavecal exposures are used by **calcos** to determine the location of both the wavecal image and the corresponding science image on the detector. The locations may vary in a non-repeatable manner due to non-repeatability of the COS grating positions, but are always at a fixed position from one another. When auto-wavecals are acquired as separate exposures they are taken close in time to the science exposures, with the grating in the same position as during the science exposure.

After processing data through **calcos**, you may decide that you need to shift the spectrum along the dispersion direction to correct offsets in the wavelength calibration. For example, wavelength calibration offsets may occur due to offsets of the target from the center of the PSA aperture (which can occur if the target acquisition was imperfect), or from drift of the grating due to thermal flexure. Assuming that **calcos** has been run on the data and a residual wavelength offset has been found in the calibrated spectrum, the offset can be corrected by first calculating the number of pixels corresponding to the offset, then subtracting it from the CORR or FULL position of coordinates along the dispersion direction. In the example below, the shift is applied to the XFULL column of the `corrtag` or `corrtag_(a,b)` file. In **Python**, the corrtag can be modified to apply the shift as follows:

```
> from astropy.io import fits
> hdulist = fits.open('ldel05ivq_corrtag_a.fits', mode='update')
> corrdata = hdulist[1].data
> corrdata['XFULL'] = corrdata['XFULL'] + SHIFT
> hdr = hdulist[0].header
> hdr['X1DCORR'] = 'PERFORM'
> hdulist.close()
```

In the example above, the XFULL positions of the science spectrum in an FUV A `corrtag` file have been moved by "SHIFT" pixels. Also, the header was updated such that `X1DCORR` will be performed again. The `corrtag` can then be run through **calcos** again if desired.

There is also a terminal command line option to apply a SHIFT1 value that is different than the one **calcos** finds internally. The values need to be in a separate file with the following columns included in the file: Dataset rootname, fpoffset, flash number, stripe or segment, desired SHIFT1 value, and desired SHIFT2 value. **Calcos** will recognize "any" as a value for fpoffset if the rootname is unique (not the association rootname). It will also recognize "any" for the flash number and stripe and segment. While SHIFT1 values (dispersion direction shift) must be specified in the file, specifying SHIFT2 values (cross-dispersion direction shift) is optional. This option can be called in the terminal as:

```
calcos —shift shift_filename.txt rawtag.fits
```

or in **Python** after importing **calcos** as:

```
calcos.calcos("rawtag.fits",shift_file="shift_filename.txt").
```

Adjusting the Background Subtraction

For spectra, background regions offset from the extraction region are used to determine the background. You can adjust the default parameters for this background region by first copying the `_1dx` reference file listed under the `XTRACTAB` keyword in the primary header (or `_2zx` reference file if performing a two-zone extraction) to a local directory, then adjusting the background parameters within the local version of the `_1dx` or `_2zx` reference file. Once you have adjusted the parameters to your satisfaction, edit the primary header of the `rawtag` file to indicate the path to the local version of the `_1dx` or `_2zx` file. You can then run **calcos** with the updated background subtraction parameters.

Examples of customizing extraction box heights are provided in Appendix A of COS ISR 2017-03. The background parameters available for editing in the `_1dx` and `_2dx` files are described in Section 3.4.19.

The header keywords in the data files to be processed need to point to the updated `_1dx` and `_2zx` reference files, and the headers scan be edited with the **setval()** function, part of the **astropy.io.fits** module in **stenv**. Below is an example modifying the header to point to a custom `_1dx` file:

```
> from astropy.io import fits
> fits.setval('filename_rawtag_a.fits', 'XTRACTAB', \
      value='custom_file_1dx.fits', ext=0)
```

5.3.3 Splicing Extracted Spectra

The **IRAF** task **splice** can be applied to combine overlapping extracted COS spectra (e.g., spectra taken with different central wavelengths). Information about using **IRAF** and the **STSDAS** analysis software can be found in Chapter 3 of the *Introduction to HST Data Handbooks*. Users should be aware of differences in spectral resolution at a given wavelength between different cenwaves. **Splice** takes into account the error (ERR) array as well as the data quality (DQ) array. Handling of the DQ array is important as it helps **splice** perform the combination properly and avoid bad or noisy data in the output file arising from the large changes in throughput at the edges of the detector.

```
cl> splice obs1_x1d.fits,obs2_x1d.fits output_splice.fits
```

Please refer to the **splice** task help file for more useful information. If a multispec format spectrum is preferred for further analysis, the task **tomultispec** can be run on the output file of the **splice** task.

Running **splice** as mentioned above (rather than transforming individual `x1d` fits tables into multispec format before combining them) has important advantages: it keeps the science data, error, and DQ arrays intact allowing for easier error analysis, and it does not have a limitation on the number of segments or wavelengths to include, which is the case with the multispec format. This limitation is caused by the size limit of the FITS header, which requires the wavelength scale to be fit with a function. We note that a **Python** replacement for the **IRAF** task splice is under development, and **IRAF** is no longer actively maintained.

---

[1] For FUV `x1d` files, both segments A and B will be present as long as the individual raw data from both segments were available at the time of processing. If only one segment was present during processing, then a row selector of row=0 will point to the data from that segment. Similarly, a row selector of row=1 will result in an error.

5.4 Working with TIME-TAG Data

COS detectors can be used in `ACCUM` or `TIME-TAG` modes, as described in Chapter 5 of the *COS Instrument Handbook*. In `TIME-TAG` mode, the position and detection time of every photon is recorded in an events list. Detection times are recorded with 32 millisecond precision, although events may be buffered for as long as 32 milliseconds prior to assignment of a detection time. **This 32 millisecond rate may lead to a spurious four-second periodicity in plots of count rate vs. time.**

For `TIME-TAG` datasets, the *HST* archive returns a raw events list in a file with a `rawtag` suffix. The `rawtag` file is a FITS file with two binary table extensions. The first extension contains the events list and the last extension a list of good time intervals, indicating time intervals when events are valid. More details are given in Section 3.2.

An events list in a `rawtag` file is a FITS binary table extension named EVENTS, containing four columns named TIME, RAWX, RAWY, and PHA. Note only FUV data will include the PHA columns in the `_rawtag` files.

The TIME in the events extension contains the time when each event was recorded, relative to the start time (MJD) of the exposure given in the `EXPSTART` keyword of the primary FITS header.

In `TIME-TAG` the RAWX column contains the pixel coordinate along the spectral axis where each event was recorded. Corrections to remove Doppler shifts introduced by the orbital motion of *HST* are applied by **calcos** and placed in the `corrtag` file. The correction depends on optical element and the projected orbital velocity of *HST*, which varies over the course of an observation. In `ACCUM` mode, this Doppler compensation is applied on orbit during an observation and is included in the RAWX column, but in `TIME-TAG` mode the uncorrected positions are downlinked and Doppler compensation is applied during ground processing. The RAWY column contains the pixel coordinate along the spatial, or cross-dispersion, axis. No Doppler compensation is applied. The PHA column (for FUV data only) contains the pulse height amplitude for each event as an integer on a 5-bit scale.

After all EVENTS extensions in a `rawtag` file, there will be one final binary table extension named GTI, containing columns named START and STOP. There will be associated start and stop times for every uninterrupted observing interval during a planned exposure. For most datasets, there will be only one START and one STOP time encompassing all buffer dumps in an exposure. Multiple good time intervals are possible, however—or example, if guide star lock is lost. Times in START and STOP are expressed in seconds since the start time (MJD) of the exposure given in the `EXPSTART` keyword of the primary FITS header is also expressed in seconds. The exposure start time (JD) is also provided in the `EXPSTARTJ` keyword of the primary FITS header. In **Python**, good time intervals can be examined using the **fits** module in the **astropy.io** package:

```
> from astropy.io import fits
> gti = fits.getdata('rootname_rawtag_a.fits', ext=2)
> print(gti)
[( 0.,  289.024)]
```

where rootname must be replaced by the rootname of the `rawtag` file being examined. Note, the 2nd extension of the `rawtag` files includes all the GTI. In this example, there is just a single GTI.

5.4.1 Displaying `TIME-TAG` Data in DS9

To view a `TIME-TAG` file (`rawtag_(a,b)` in the FUV, `rawtag` in the NUV), open ds9, then choose ‘open’ from the menu bar at the top. The image will load but, save for a few pixels registering a value of 1, the remaining pixels will be zero.

Once the image is loaded, go to the menu item ‘bin’ and open the pull-down menu from that. From that pulldown menu you can choose the size of the image to view—generally you should make it as big as possible: 8192 × 8192 pixels for FUV data, 1024 × 1024 pixels for NUV.

NOTE: for the instructions below, the changes will not take effect until you click on the 'Apply' button.

Now choose 'Binning Parameters' from the 'bin' pulldown menu. This will open a new window with the binning parameters listed. You will notice right away that the bin columns are listed as TIME and RAWX. These are what is currently being displayed by **ds9** (which is why the image looks so strange when initially loaded). However, what you really want is RAWX vs. RAWY, so change that in the pulldown menu under bin columns.

You can also set the blocking size of the image in the 'Binning Parameters’ window—just type in '2' in the Block field next to RAWX. By blocking this way along the dispersion direction, you can now see virtually all of the 16,384 pixels along the dispersion direction. If you are looking at NUV data, then no additional blocking is needed—just leave the blocksize as 1, but choose the image size as 1024 pixels from the 'bin' pulldown menu.

Spectroscopic Data

Next, from the 'Binning Parameters' window choose the part of the spectrum to be centered on the middle of the dispersion direction by clicking on the button marked 'or center of data'. Now press 'Apply' on the binning parameters window to update the **ds9** display.

The spectrum should now be displayed, with the dispersion direction running from left to right. To better see the data, choose 'scale' under the main **ds9** menu bar, and from that pulldown menu choose a square root stretch and min/max range. You can now pan your cursor over the image, while holding the right button down on your cursor, until the contrast looks just right. If you would like to smooth the data a bit (this can be useful for bringing out fainter features and increasing signal to noise along the display), choose the 'Analysis' menu item under the main **ds9** menu bar and select 'smooth parameters.' A dialogue box will open, and from there you can set the number of pixels to smooth. Finally, you can also click on the 'Color' item on the **ds9** menu bar and choose 'invert color map' to get an inverted color map.

You can also load a `corrtag_(a,b)` table in **ds9**, but in this case the appropriate columns to display are XFULL and YFULL. Otherwise, the same **ds9** commands apply as for `rawtag` files. For both `TIME-TAG` and `ACCUM` spectroscopic data the `flt` and `counts` spectral images will load as simple 2-D images in **ds9**.

Imaging Data

For both `TIME-TAG` and `ACCUM` imaging data the `flt` and `counts` images will load as simple 2-D images in **ds9**.

`TIME-TAG` Animation

You can assign events registered during each time interval to a separate image in **ds9**, thereby creating a sequence of images which can be played as an animation. This can be useful in verifying the occurrence of lamp flashes in `TAGFLASH` data, in searching for the appearance of bursts in raw data (although bursts have not yet been seen with COS), and so on. To bin the images in time, set up the image as described above—with RAWX and RAWY chosen in the 'Binning Parameters' dialogue box. At the bottom of the 'Binning Parameters' box is a parameter called 'Bin 3rd Column,' Set the value of this parameter to TIME. Next, choose the number of bins you would like to divide the event file into under the 'Depth' parameter. Setting this value to 10, for example, will create 10 separate images, with the first one showing all events registered during the first (`EXPTIME`/10) seconds, the next one showing all events registered between (`EXPTIME`/10) and (2*`EXPTIME`/10) seconds, the next showing all events registered between (2*`EXPTIME`/10) and (3* `EXPTIME`/10), and so on up to `EXPTIME`. The 'Min' and 'Max' parameters let you choose the range of values in time to display—usually this is pre-set to 0 and `EXPTIME`, and can be left unchanged to bin the entire image as above. Select 'Apply' to do the binning.

Note that some time will be required to create the sequence of images, and that binning the events in time in **ds9** is very memory intensive, and that **it is easy to make ds9 crash if `EXPTIME` is large (for example >1000 seconds) and the number of bins in 'Depth' is set to a large value (for example 30)**. It is best to start with a small value for 'Depth' that works, then increase the value if needed.

After the binning is done, a new dialogue box will appear called 'Data Cube.' Numbered from left to right will be the enumeration of the bins (in the example above from 1 to 10), along with a slider underneath. Click on 'Play' in that window to start the animation—it will play each of the binned images sequentially in the ds9 window. Again, the spacing between each of the bins will be (`EXPTIME`/10) in seconds, or (`EXPTIME`/Nbin), where Nbin is the number of bins.

In the animation, it should be possible to see the `TAGFLASH` spectrum appear and disappear as the sequence progresses. Obviously the sequence will show the flashes only if the keyword `TAGFLASH=AUTO` or `TAGFLASH=" UNIFORMLY SPACED"` is in the header of the event file.

To exit from the animation, close the 'Data Cube' window, and then set the 'Depth' parameter in the 'Binning Parameters' dialogue box to zero, and click 'Apply.' That will reset the image in **ds9** to show all of the data again.

It is possible to bin in other parameters as well, such as PHA. The logic is the same as above.

5.4.2 Filtering Time-Tag Data

Filtering Events in the Timeline Extension

All `corrtag` files processed with **calcos** version 2.14 or later contain a `timeline` extension. The `timeline` extension can be operated on by the **timefilter** module to exclude photon events that match user-specified patterns. The **timefilter** module is available as part of **costools**, and requires **calcos** version 2.14 or later to work. One common use of **timefilter** is to exclude daytime events in order to minimize the contribution of geocoronal Lyman alpha or O I emission lines to your data. **Timefilter** will filter events according to a filter string passed to it.

The filter string consists of one or more filter conditions, separated by "and," "or," or "xor" (parentheses are currently unsupported). Each filter condition consists of a column name, a relation, and a cutoff value. Valid column names are "time," "longitude," "latitude," "sun_alt," "target_alt," "radial_vel," "shift1," "ly_alpha," "OI_1304," "OI_1356," and "darkrate" (see Table 2.3 for a description of the columns). Valid relations are '>', '>=', '<', '<=', '==', and '!='. Cutoff values are numerical values. In addition, it is possible to flag events based on one of the 32 SAA model contours with the filter condition "SAA #" where # is a number from 1 to 32. Events which match the filter string will be marked with the DQ flag 2048 (bad time interval), and will be excluded in the creation of `flt` and `x1d` files.

Timefilter can either modify an existing `corrtag` file in place, or create a new one, and it can be run in conjunction with **splittag** (although in that case, it is possible that some output files will contain no valid events at all). The produced corrtag file may be extracted with the **calcos** package as usual. It is possible to remove any events filtered with **timefilter** by setting the filter to "reset" followed by setting the filter to ''clear."

The following examples show common uses of **Timefilter**:

- Take "test_corrtag_a.fits," flag all data taken during orbital day (sun_alt > 0), and save in the file "output_corrtag_a.fits"

```
> from costools import timefilter
> timefilter.TimelineFilter('test_corrtag_a.fits', \
                            'output_corrtag_a.fits', 'sun_alt > 0.')
```

- Filter "xyz_corrtag_b.fits" in place to remove data with (sun_alt > –10 AND ly_alpha > 2.5) OR taken in the SAA 31 profile

```
>from costools import timefilter
>timefilter.TimelineFilter('xyz_corrtag_b.fits', '', \
                      'sun_alt > -10 and ly_alpha > 2.5 or saa 31.')
```

- Remove filters from "xyz_corrtag_b.fits"

```
>from costools import timefilter
>timefilter.TimelineFilter('xyz_corrtag_b.fits', '', 'reset')
>timefilter.TimelineFilter('xyz_corrtag_b.fits', '', 'clear')
```

Manipulating `TIME-TAG` Data for Variability

Users may wish to process only sub-intervals of TIME-TAG events, to look for variability in the data. One way to do this would be to divide an exposure up into several sub-exposures before re-processing by using the **splittag** module. The **splittag** module is available as part the **costools** package within **stenv.**

Splittag is a useful tool for dividing a COS time-tag exposure (FUV or NUV) into a series of sub-exposures with time intervals specified by the user. The task operates on the **calcos** `corrtag` files, copying rows from a `corrtag` file into one or more output files. The number of files depends on the number of time intervals specified by the user. The resulting `corrtag` sub-exposures can then be run separately through **calcos** to extract one-dimensional, flux-calibrated spectra (`*_x1d.fits` files) for each file.

The following keywords are modified when **splittag** copies the time columns to the new `corrtag` files: `EXPTIME`, `EXPEND` and `EXPENDJ`. The keywords `EXPSTART` and `EXPSTRTJ`, on the other hand, are not changed. The `EXPTIME` keyword in each of the new `corrtag` files will be set to the duration of the time interval being extracted, while the modified Julian date and Julian date in `EXPEND` and `EXPENDJ` will be set to the following:

$$\mathtt{EXPEND} = \mathtt{EXPSTART} + t\_end(i)$$

$$\mathtt{EXPENDJ} = \mathtt{EXPSTARTJ} + t\_end(i)\ ,$$

where t_end(i) is the ending time of the i-th desired sub-exposure. In addition to the updated keywords, **splittag** also produces updated GTI (good time interval) tables for each of the output `corrtag` files. The GTI intervals are specified relative to the times of the original `corrtag` file, such that the split `corrtag` files will not include events outside the GTI values. The `EXPTIME` keyword written to the `corrtag` files affected by the GTI intervals is shortened accordingly.

There are two ways to run the **splittag** task: (1) specify a starting time, an increment, and an ending time, or (2) provide an explicit list of times (not necessarily adjacent to one another). In either case, the output `corrtag` files will have a root name specified by the user. If no root name is specified, the root name of the input `corrtag` will be used, appended with numbers 1,...N for N exposures.

The parameters input by the user for **splittag** include the following: an input `corrtag` file name, a root name for the output files, the starting time for the first event to be extracted, the time increment to be used in extracting the following intervals, and the ending time of the extraction. If option (1) from above is used, then the starting time and increment are specified, with the remaining parameters left at their indefinite values. This will extract however many `corrtag` files are needed until the ending time of the original exposure is reached. If option (2) is used, then the user can specify explicitly, in the form of start/stop pairs, which intervals are desired. For example, specifying time_list="0,20,100" will extract events in the range $0 < t < 20$ seconds and output that to a `corrtag` file, then extract events in the range $20 < t < 100$ seconds and write that to another file, and so on. **Splittag** can also read in a text file with the start/stop pairs entered (using the format in the example above). In that case, all the start/stop pairs would be listed in one line in the text file, separated by either commas or spaces. If this option is used (i.e., the time_list parameter is set to point to the text file), then the `starttime` and `increment` parameters are ignored.

For example, to split the exposure, `l61h9002r_corrtag.fits` into two sub-exposures, with an output of 'split':

```
> from costools import splittag
> splittag.splittag('rootname_corrtag_a.fits', 'split', \
             starttime=None, increment=None, \
             endtime=None, time_list='0, 60, 120')
```

Next, the two sub-exposures should be extracted with **x1dcorr** either as a separate task or by running **calcos** from the `corrtag` file. (Note, however, that the x1dcorr task is now broken; for more information, see *Running x1dcorr task in costools and custom COS spectral extraction*, in the HST Knowledge Base.) To instead split the exposure into 20-second increments, the following command would be used instead:

```
> from costools import splittag
> splittag.splittag('rootname_corrtag_a.fits', 'split', \
             starttime=None, increment=20., \
             endtime=None, time_list=None)
```

Appendix A: COS Lifetime Positions

Appendix Contents

A.1 COS Lifetime Positions

To mitigate the effect of gain sag, COS FUV spectra are obtained at multiple positions on the detector that are offset from each other in the cross-dispersion direction. These are known as Lifetime Positions (LPs). The introductions of LP2, LP3, LP4, LP5, and LP6 occurred on July 23, 2012, February 9, 2015, October 2, 2017, October 4, 2021, and October 3, 2022, respectively.

As of publication, G130M cenwaves 1055 and 1096 are observed at LP2. G130M cenwave 1222 is observed at LP4. Other G130M cenwaves are observed at LP5. G160M cenwaves are observed by default at LP6, with exposures shorter than approximately half an orbit eligible for LP4 upon request, to reduce overheads. All G140L cenwaves are observed at LP3. Any changes to this arrangement will be announced to the community in STScI Analysis Newsletters (STANs).

LP2 is located at +3.5 arcsec in the cross-dispersion direction, LP3 is located at –2.5 arcsec in the cross-dispersion direction, and LP4 is located at –5.0 arcsec in the cross-dispersion direction. LP5 is located at +5.4 arcsec in the cross-dispersion direction, and LP6 is located at +6.5 arcsec in the cross-dispersion direction. Figure A.1 illustrates the location of each Lifetime Position on a map showing the modal gain of Segment B of the FUV detector as of October 2022.

The LP4 move was accompanied by a new set of restrictions on detector segment usage, FP-POS selection, and target acquisition settings. These changes, which will remain in effect at subsequent LPs, are together known as the COS2025 policy, described at http://www.stsci.edu/hst/instrumentation/cos/proposing/cos2025-policies. The main culprit of gain sag in the COS/FUV detector is Ly α geocoronal emission at 1216 Å that fills the COS 2.5" aperture whenever the G130M grating is used with the 1291/1300/1309/1318/1327 cenwaves on Segment B. In order to mitigate this gain sag, the COS2025 policy minimizes the number of locations on the detector where the geocoronal Ly α emission falls by not supporting the Segment B G130M 1300/1309/1318/1327 cenwaves and reducing the number of FP-POS for 1291 from four to two. In this way the geocoronal Ly α emission falls on fewer locations on the detector, and each LP can be used until the continuum level reaches a low modal gain.

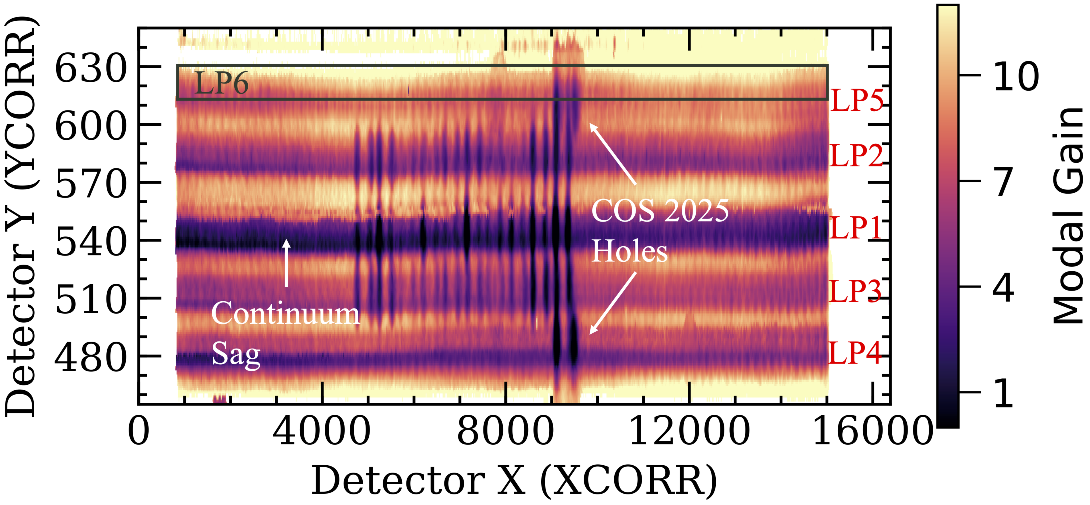
Each change in Lifetime Position has resulted in a change in the spectral resolution, and small changes in the core of the line profile and the wings of the line spread function (LSF). The spectral resolution at different Lifetime Positions has been characterized (COS ISR 2013-07, COS ISR 2017-06, and COS ISR 2018-07 for LP2 through LP4, with ISRs for LP5 and LP6 in preparation), and they find a decrease of 5–10% for LP2 and LP3 below LP1, and another 10–15% below LP3 for LP4. The resolution at LP5 is about 10% better than at LP4, and the resolution at LP6 is about 10% worse. The resolution tabulated in Table 1.1 corresponds to LP4 for the FUV cenwaves. The resolution as a function of Lifetime Position is shown in Figure 1.1. The shapes of the line profiles are well represented in the LSF models available to the community at http://www.stsci.edu/hst/instrumentation/cos/performance/spectral-resolution.

The flux and wavelength calibrations are performed in a similar way for all Lifetime Positions. Calibration observations are executed in preparation for lifetime moves, and updates to calibration reference files are made. These observations include measurements of the spatial and spectral resolution, verification of the wavelength scales, verification of the FUV BOA operations if available, and flux and flat field calibration. In addition, LP3 and beyond make use of profile calibration files for the two-zone extraction described in Section 3.2.1, while LP1 and LP2 use a boxcar extraction.

The calibration pipeline automatically uses the correct calibrations methods based on the Lifetime Position specified in the header keyword `LIFE_ADJ`, which also enables CRDS to determine the most up to date reference files for that Lifetime Position. `LIFE_ADJ` can have values of 0, 1, 2, 3, 4, 5, 6, –1, –11, and –999. A value of 0 corresponds to pre-launch data. Values of 1 through 6 correspond to those Lifetime Positions, and as new Lifetime Positions are added they will be designated 7 and higher. A value of –1 is specified if the aperture position is at a non-standard location. A value of –11 is only used in reference files and indicates that the file should be applied to all data with `LIFE_ADJ`=1 and –1. Lastly, a value of –999 is used for dark exposures, where there is no defined LP.

The varying Lifetime Positions are just for the FUV, and NUV spectroscopic observations remain at LP1 of the NUV detector.

Figure A.1: Location of the Lifetime Positions on Segment B.



The color map shows the modal gain of the FUVB segment as of October 2022. The allowed COS2025 Ly alpha airglow holes are indicated, along with an example of continuum gain sag. Note that the COS2025 policy, which took effect with the move to LP4 and will remain in effect for subsequent LPs, results in fewer gain sag holes due to the limited number of locations where Ly alpha can be placed on the detector. The dark box indicates where LP6 has been placed.

Appendix B: COS High Voltage History

Appendix Contents

B.1 COS High Voltage History

The high voltage on the two FUV detector segments has been adjusted numerous times since launch in order to optimize the performance. Table B.1 lists the nominal high voltage values used since COS installation in 2009.

The initial values used were identical to those used during ground testing. After early on-orbit tests showed that the gain of the MCPs was higher than expected on both segments, the voltages were lowered to return the gain to the prelaunch values. As described in Section 1.2.1, exposure to photon events lowered the gain in the spectral region at the original lifetime position, so the voltage was raised on several subsequent occasions as shown in Table B.1 to keep the gain high enough to minimize throughput loss. The voltage has been adjusted as needed since then to keep the gain in the spectral region at acceptable levels.

In addition to the nominal voltages listed in the table, other values have been used in a number of calibration activities, so some data files in the archive have nonstandard voltage levels. For a complete listing of the FUV high voltage, users should refer to: FUV High Voltage History. Also, the HV used in an individual observation is listed in its header.

Table B.1: Lifetime Position and Nominal High Voltage Values (Segment A/B) for the COS FUV Detector.

Grating:	G130M								G160M		G140L			
Central Wavelengths	1055, 1096		1222		1291		1300, 1309, 1318, 1327		1577, 1589, 1600, 1611, 1623		1105[1]		1280[2]	
5/11/2009	Not used		Not used		LP1	178/175	LP1	178/175	LP1	178/175	LP1	178/100	LP1	178/175
8/12/2009	LP1	169/167[3]	LP1	169/167	LP1	169/167	LP1	169/167	LP1	169/167	LP1	169/100	LP1	169/167
3/8/2011	LP1	169/175	LP1	169/175	LP1	169/175	LP1	169/175	LP1	169/175	LP1	169/100	LP1	169/175
3/26/2012	LP1	178/175	LP1	178/175	LP1	178/175	LP1	178/175	LP1	178/175	LP1	178/100	LP1	178/175
7/23/2012	LP2	167/163	LP2	167/163	LP2	167/163	LP2	167/163	LP2	167/163	LP2	167/100	LP2	167/163
6/24/2013	LP2	167/169	LP2	167/169	LP2	167/169	LP2	167/169	LP2	167/169	LP2	167/100	LP2	167/169
7/21/2014	LP2	167/175	LP2	167/175	LP2	167/175	LP2	167/175	LP2	167/175	LP2	167/100	LP2	167/175
11/3/2014	LP2	173/175	LP2	173/175	LP2	173/175	LP2	173/175	LP2	173/175	LP2	173/100	LP2	173/175
2/9/2015	LP2	173/175	LP3	171/167	LP3	167/163	LP3	167/163	LP3	167/163	LP3	167/100	LP3	167/163
1/18/2016	LP2	173/175	LP3	171/169	LP3	167/169	LP3	167/169	LP3	167/169	LP3	167/100	LP3	167/169

10/17/2016	LP2	173/175	LP3	171/175	LP3	167/175	LP3	167/175	LP3	167/175	LP3	167/100	LP3	167/175
10/2/2017	LP2	173/175	LP4	163/167	LP4 (LP3)[4]	163/163 (167/175)[4]	LP4 (LP3)[4]	167/100 (167/175)[4]	LP4	163/163	LP4	163/100	LP4	163/163
10/5/2020	LP2	173/175	LP4	167/169	LP4 (LP3)[4]	167/169 (173/175)[4]	LP5 (LP3)[4]	167/100 (173/175)[4]	LP4	167/169	LP3	173/100	LP3	173/175
10/4/2021	LP2	173/175	LP4	173/169	LP5 (LP3)[4]	167/169 (173/175)	LP5 (LP3)[4]	167/100 (173/175)	LP4	173/169	LP3	173/100	LP3	173/175

[1] A commanded high voltage of 100 means that no data is collected on that segment.

[2] In Cycle 17, the 1230 Central Wavelength setting was replaced by 1280.

[3] Segment High Voltage in Volts = –(Commanded voltage*15.69 + 2500).

[4] Configurations in parentheses are Available-but-Unsupported options, as described in the COS Instrument Handbook.

List of Figures

List of Tables