Instrument Science Report STIS 2019-02

Identifying Jitter Induced CCD CR-SPLIT Combination Errors

Joleen K. Carlberg[1]

[1] *Space Telescope Science Institute, Baltimore, MD*

17 June, 2019

ABSTRACT

By default, all STIS CCD exposures are split into a minimum of two subexposures to allow for cosmic ray removal. An underlying assumption of the cosmic ray removal algorithm is that the subexposures are well-registered. This assumption breaks down if the target wanders too much in the spectroscopic slit during the observation due to telescope jitter, particularly if the jitter occurs predominantly in one subexposure. In such cases, the algorithm may reject large fractions of valid data, leading to systematically underestimated flux, even for wide slit widths where slit losses from the jitter are negligible. Such datasets may or may not present with unusual line profiles. In this work, we present a technique for using the cosmic ray data quality flags to identify potentially problematic spectra and demonstrate its effectiveness in flagging observations of standard stars with apparent 5-15% flux losses.

Contents

1. Introduction

This project was initiated by an investigation of an apparent 5% throughput drop for dataset ODVKL1040, a G430L 52×2E1 observation of standard star AGK+81D266, which is observed three times per year to monitor changes in STIS's sensitivity as part of CAL/STIS program 15558. This exposure was one of six and occurred roughly in the middle of a single-orbit visit. All five of the other exposures of the same target had expected flux levels, ruling out problems in the target acquisition. There were no alerts for guide star problems during the observation, and spatial cuts through the _crj file showed no indication of focus issues. The trace parameters reported in the _sx1 file also looked appropriate.

Further investigation revealed that the telescope pointing along the V2 axis was offset by ~0.03″ for the first of the two CR-SPLIT subexposures in the dataset but normal for the remaining subexposure. Although such large jitter is unusual, it is still quite small compared to the 2″ slit width, and any jittering of the target perpendicular to the slit should have negligible impact on the recorded flux. Instead, the apparent flux loss in the extracted spectrum results from the jitter parallel to the slit and how the cosmic ray rejection algorithm behaves when the input images to be combined are not well-aligned spatially.

In their paper on the **calstis** cosmic ray (CR) rejection algorithm, Shaw and Hodge (1996) estimated that mis-registration of images $\gtrsim$ 0.1 pixel can lead to substantial algorithmic failure. At the CCD plate scale (0.050872″/pix), this translates into 0.005″ of movement in the spatial direction. The STIS slits are rotated roughly 45° in the V2/V3 plane of HST; therefore, jitter at the level of $0.005'' \times 2/\sqrt{2}$ = 0.007″ along either axis is sufficiently large to create 0.1 pixel misalignment. Algorithmic failures may occur at even smaller misalignments, since the exact response of the CR algorithm to misaligned spectra also depends on the sharpness of the line profile, the number of spectra to be combined, and the S/N properties of the data.

Although the pitfalls of performing CR rejection on spatially misaligned images are well-known and documented (see, e.g., The STIS Data Handbook 3.5.4, Sohn et al. 2019), typical users may not expect problems under normal observing conditions (two guide star fine-locked observations with no dithering). However, in the current gyro configuration, large but short-lived pointing excursions can occur, particularly along the V2 axis (see March 2019 STAN[1]), and Figure 1 illustrates the large V2 jitter that affected dataset ODVKL1040 during the first half of the exposure. Therefore, this ISR revisits the root cause of CR rejection errors and provides a simple tool for identifying affected datasets.

[1] http://www.stsci.edu/contents/news/stis-stans/march-2019-stan.html

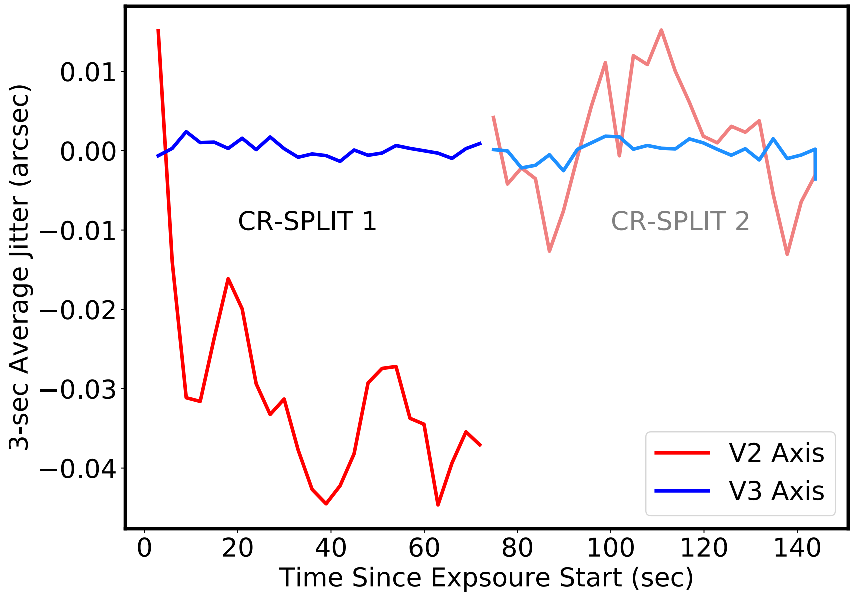

Figure 1. Time averaged jitter during the observation of ODVKL1040. The jitter along the V2 and V3 axes are shown in red and blue, respectively. For additional clarity, the jitter in the first and second subexposures are shown in darker and lighter shades, respectively. Not only is the jitter along the V2 axis larger and noisier than along the V3 axes, but it is also systematically offset between the two CR-SPLIT subexposures by ~0.03″.

2. The Subtle Failure of Mis-Registered Spectra

When the average spatial position of the target differs between co-added exposures, the peaks of the line spread functions in each column may not occur at the same y-location on the detector. If there are only two subexposures, the CR algorithm may reject the data from whichever subexposure has the locally higher counts and scale up the lower counts to compensate for the rejected data. In other words, it systematically underestimates the net counts. Figure 2 demonstrates this failure. The solid black and purple lines show slices through rows in the flat-fielded subexposures (extensions 1 and 4 of the _flt file). The dotted green line shows a slice through the same row in the _crj file, divided by two. If no CR rejection occurred, the green line would be the average of the two input exposures. Instead, it always tracks the lower of the two spectra.

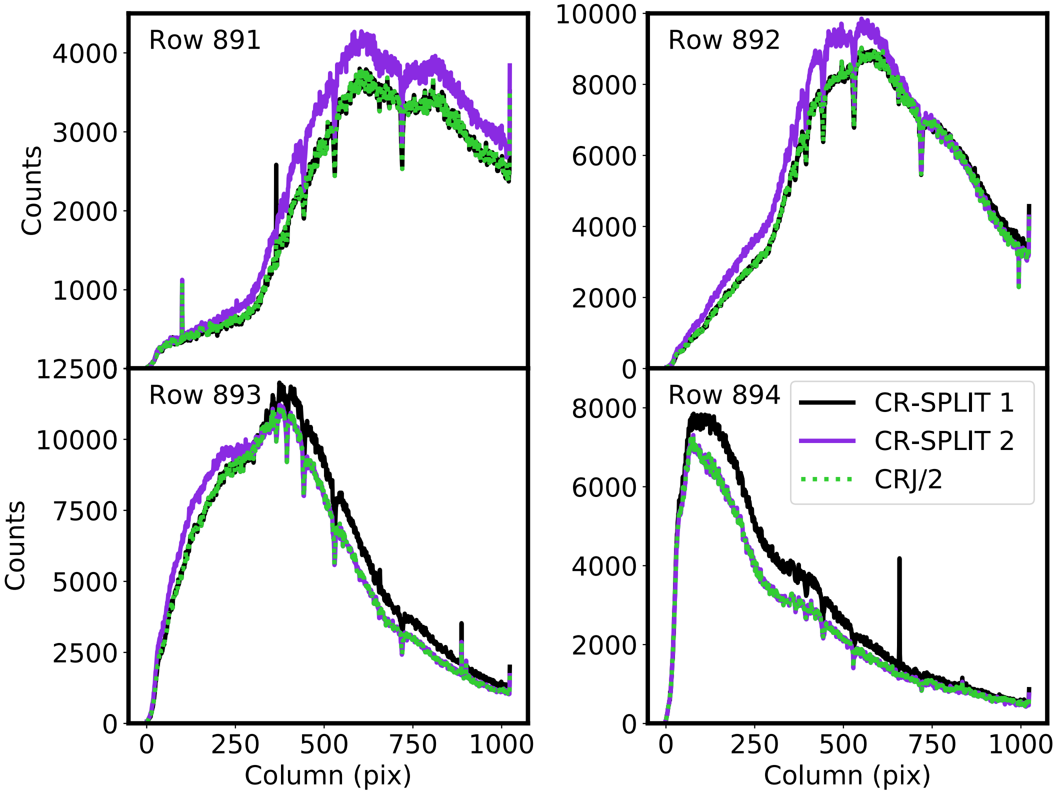

Figure 2. Slices through four rows of dataset ODVKL1040, sampling the peak of the line spread function. The black and purple lines slice through the individual CR-SPLIT exposures in the _flt files, while the green dotted line shows half the counts along the same slice in the corresponding _crj file.

Figure 3 illustrates the effect of the CR rejection on the extracted counts. The vertical slices through column 512 in this figure show the profiles that will be summed over during the spectral extraction. The individual CR-SPLIT exposures are again shown in black and purple, and the sum of these profiles is shown with the gray dashed line. The green line shows the same profile in the _crj file, which is the file **calstis** uses for extraction, and the peak of the counts profile is ~6% lower than that of the straight sum. In this example, the only obvious red flag is the unexpected drop in flux. There is no distortion in the line profile shape to arouse suspicion. (Contrast this to the case illustrated in Shaw and Hodge 1996.) When the absolute scale of the spectral energy distribution of the science target is unknown, the user must inspect the intermediate data products to determine whether a problem exists.

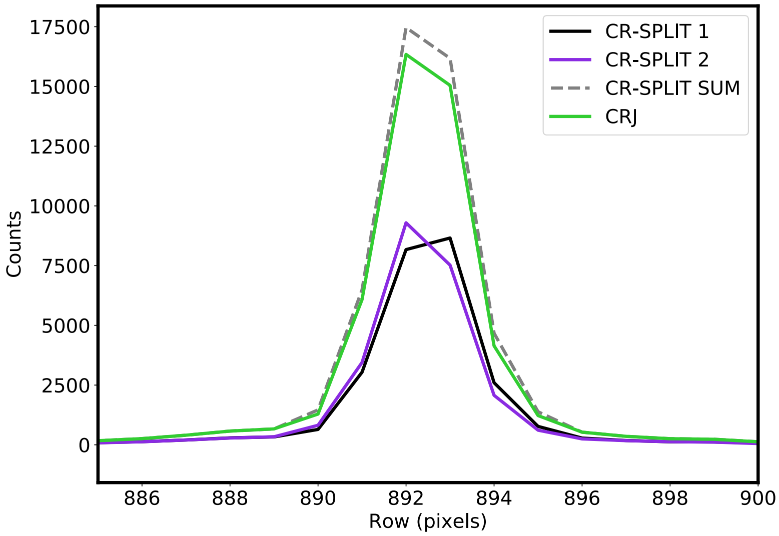

Figure 3. Vertical slices through column 512 of dataset ODVKL1040. These are the line profiles that will be summed over during spectral extraction. The black and purple lines slice through the individual CR-SPLIT exposures in the _flt files, while the green line shows the same slice in the corresponding _crj file. A straight sum of the CR-SPLIT profiles is shown by the gray dashed line.

3. Identifying Problematic Datasets

3.1 Using Cosmic Ray Statistics

An efficient way of identifying whether the cosmic ray algorithm is rejecting too much valid data is to compare the fraction of cosmic ray rejected pixels in the extraction region compared to that of the full CCD. The latter information can be trivially computed from the primary header keywords REJ_RATE (rejected pixels/sec) and TEXPTIME of the calibrated data product. The pixels of the _flt file included in the spectral extraction can be identified with the columns EXTRLOCY (the central y location of the extraction for each column in 1-indexed coordinates) and EXTRSIZE (extraction box size) in the STIS extracted spectrum files (e.g., _sx1 file). The data quality arrays of each CR-SPLIT exposure in the _flt files are updated by **ocrreject** to identify which pixels were rejected as cosmic rays by setting bit 13 (flag value of 8192).

The *stistools* package contains **crrej_exam**, a command line python script that calculates the rejected pixel fraction in a dataset, and the portion of the code needed to recreate the results for a single dataset is reproduced in Appendix A. The python script takes a list of dataset IDs and optionally a directory containing the data (default is the current directory). Note that the directory must contain both the _sx1 and _flt files from the dataset. A sample call and output of this script on dataset ODVKL1040 is:

```
$ crrej_exam odvkl1040
  Analyzing odvkl1040:
  Percentage of Pixels Rejected as CRs
     Extraction Region: 33.9%
     Full CCD Frame: 1.1%
```

In this example, one-third of the pixels in the extraction region were cosmic ray rejected compared to only 1% in the entire CCD. Similar results are found for the mis-registered dataset, O3TT40040, identified by Shaw & Hodge (1998) and an anomalous throughput dataset, OCEIL4060 with throughput drop of 12%, identified by Sana et al. (2015).

```
$ crrej_exam o3tt40040 oceil4060 -d failed_datasets/
  Analyzing o3tt40040:
  Percentage of Pixels Rejected as CRs
     Extraction Region: 31.5%
     Full CCD Frame: 2.8%

  Analyzing oceil4060:
  Percentage of Pixels Rejected as CRs
     Extraction Region: 33.8%
     Full CCD Frame: 2.4%
```

3.2 Inspecting the Jitter Files

Observers have access to the observation logs, or jitter files, that document how the telescope was pointing and tracking during the observation. These files are described in detail in Section 6.1 of the “Introduction to the HST Data Handbooks” (Shaw et al. 2011). Particularly useful is the telescope jitter along the V2 and V3 axes, which is recorded in 3-second averages as a function of time in the _jit files. The corresponding _jif files contain 2-D histogram images of the jitter distribution in STIS’s image plane (rotated ~45° from the V2 and V3 coordinate plane) and gives a quick visualization of where the telescope was pointing. The pointing information for each CR-SPLIT exposures has its own science extension in the jitter files (as in the _flt files) and can be inspected individually or summed.

Figure 4 shows the 2-D jitter histograms (summed over CR-SPLITs) for the series of six science exposures taken during the single orbit L1 visit in the CAL/STIS 15558 program. Dataset ODVKL1040 (lower left) clearly shows substantial jitter compared to most of the other exposures. The bimodal distribution highlights the average pointing offset between the two subexposures, as previously seen in Figure 1. Dataset ODVKL1060 (bottom right) also has a bimodal pointing distribution; however, in this case, the average spatial offset was not large enough to upset the cosmic ray rejection algorithm. Only 0.9% of pixels were rejected in the extraction region, compared to the full frame rate of 1.4%. The flux measured for this observation was nominal.

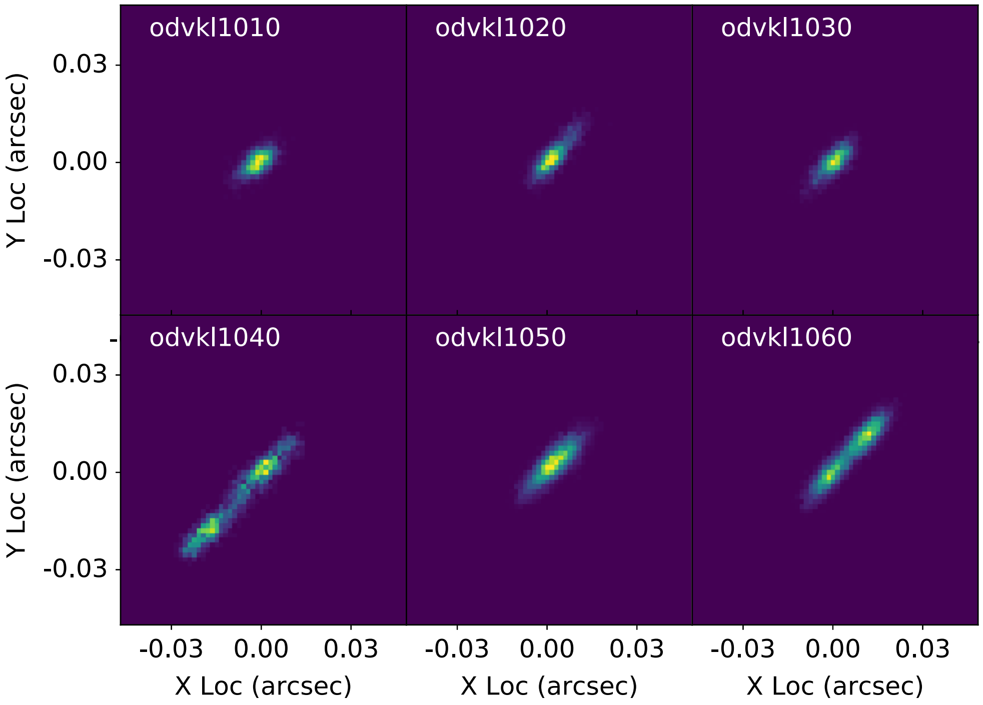

Figure 4. Images from the _jif.fits files, showing the 2-D histograms of the 3-sec avererage pointing location in the _jit.fits files. The images are in the STIS detector coordinate system.

4. Correcting the Extraction

To recover the improperly discarded flux, the default cosmic ray rejection parameters must be adjusted to increase the threshold for rejecting pixels. The two parameters adjusted for dataset ODVKL1040 are CRSIGMAS (statistical rejection criteria) and INITGUES (initial guess, either median or minimum flux). The default values used by the pipeline are '4.0' and 'minimum', respectively, and changing these to '10.0' and 'median' recovers the lost flux. The code snippets below illustrate how to use the standalone cosmic ray removal task **ocrreject**, which is available in *stistools,* to tweak the rejection parameters. Passing the new _crj file to **calstis** will complete the remaining calibration steps[2]. Note that the _flt file data quality arrays will be updated by **ocrreject** to flag the CR rejected pixels.

```
> from stistools import ocrreject, calstis
> ocrreject.ocrreject('odvkl1040_flt.fits','odvkl1040_crj.fits',
      initgues='median',crsigmas='10.0')
> calstis.calstis('odvkl1040_crj.fits', outroot='odvkl1040')
```

[2] Note that the output file may be a _x1d file instead of the expected_sx1 file. This is not a concern.

Acknowledgements

Special thanks to Sean Lockwood for valuable discussion on jitter characteristics and for reviewing this document.

Change History for STIS ISR 2019-02

Version 1: 17 June 2019 – Original Document

References

Sana, H. et al., 2015, STIS ISR 2015-09
Shaw, D. & Hodge, P., 1998, STIS ISR 1998-22
Smith, E., et al. 2011, "Introduction to the *HST* Data Handbooks", Version 8.0, (Baltimore: STScI)
Sohn, S. T., et al., 2019, "STIS Data Handbook", Version 7.0, (Baltimore: STScI)

Appendix A: The Python Code

```
   import numpy as np
   from astropy.io import fits

   sx1_file = 'odvkl1040_sx1.fits'
   flt_file = 'odvkl1040_flt.fits'

   with fits.open(sx1_file) as spec_hdu:
         spec = spec_hdu[1].data[0]
         shdr = spec_hdu[0].header

   split_num = shdr['CRSPLIT']

  #Create a mask that defines where the extraction region is
   extr_mask = np.zeros((1024,1024))
   del_pix = spec['EXTRSIZE']/2.
   for column in range(0,1024):
      row_mid = spec['EXTRLOCY'][column] – 1
      gd_row_low = int(np.ceil(row_mid - del_pix))
      gd_row_high = int(np.floor(row_mid + del_pix))
      extr_mask[gd_row_low:gd_row_high+1, column] = 1

   n_tot = np.count_nonzero(extr_mask) * split_num

   with fits.open(flt_file) as flt_hdu:
        n_rej = []

        for i in range(3,len(flt_hdu)+1 ,3):
           dat = flt_hdu[i].data

           #DQ flag 8192 used for CR rejected pixels
           rej = dat[ (extr_mask == 1) & (dat & 2**13 != 0)]
           n_rej.append(np.count_nonzero(rej))
```

```
t_exp = float(shdr['TEXPTIME'])
CR_rate = float(shdr['REJ_RATE'])

# Calculate the rejection fraction in extraction region
n_pix_rej = np.sum(np.array(n_rej))
rej_frac = n_pix_rej/float(n_tot)

# Compute the rejection fraction for the full image.
Expec_frac =  CR_rate * t_exp /(1024.*1024.*split_num)

print('Percentage of Pixels Rejected as CRs')
print('   Extraction Region: {:.1%}'.format(rej_frac))
print('   Full CCD Frame: {:.1%} \n'.format(expec_frac))
```