Instrument Science Report STIS 98-**10**

Calstis0: Pipeline Calibration of STIS Data–A Detailed View

Phil Hodge, Stefi Baum, Melissa McGrath, Dick Shaw

and the Spectrographs Group pipeline block: Ivo Busko (SSG), Jennifer Christensen, Paul Goudfrooij, J.C. Hsu (SSG), Steve Hulbert, and Rocio Katsanis

April 21, 1998

ABSTRACT

We describe calstis0, the STIS calibration program which combines all the individual calstis modules. The functionality of calstis0 is available as a host-level application and as a task in STSDAS to allow users to re-calibrate STIS data outside the pipeline with different reference files.

1. Introduction

This document describes *calstis0* in more detail than most users will need. For a more user friendly discussion, see the STIS chapters of the HST Data Handbook, volume 1.

The pipeline calibration software for STIS data, **calstis**, was written as a series of separate executables that could be run in sequence to perform the various steps needed for calibrating STIS data. The executables are called **csN.e** (or **csN.exe** in VMS), where N is an integer. Each of the csN.c source files is a main module that does little other than getting the command line arguments and calling the *calstisN* function that does the processing. The main module for **cs0.e** is in the source file cs0.c, which gets command line arguments and calls *calstis0*. *calstis0* is a "control" module, which calls each of the *calstisN* as needed. Note that the *calstisN* are called as functions; **cs0.e** is linked with these, and it does not use the independent executables **csN.e** at all. There are also IRAF tasks to execute these programs. An example of the notation used in this document is the following. **cs0.e** is the host-level executable; the IRAF task which invokes **cs0.e** is **calstis**; the high-level function called by this is *calstis0*. This document describes the *calstis0* function, what files it reads and writes, and what calibration switches and reference files it

uses. The reference coordinate system and the mapping from that to image pixel coordinates is described in the Appendix.

2. CL Scripts and Host Level Executables

The correspondence between the cl scripts and host level executables is shown in the following table.

Table 1: Names of cl scripts and executables

IRAF task	executable(s)	purpose
calstis	**cs0.e**	do all steps set to PERFORM
basic2d	**cs1.e**	bias, dark, flat
ocrreject	**cs2.e**	cosmic ray rejection
wavecal	**cs11.e, cs7.e, cs4.e cs12.e**	determine MSM shift from wavecal
x1d	**cs6.e**	extract 1-D spectrum
x2d	**cs7.e**	2-D rectify spectrum or image
none	**cs11.e**	subtract science from wavecal
none	**cs4.e**	find MSM shift, update SHIFTAi
none	**cs12.e**	copy SHIFTAi to `_flt` or `_crj` header

3. Description of Command Line Arguments

This section describes the main module cs0.c which calls *calstis0*. The command line arguments that are recognized are:

```
input
outroot
-w wavecal
-v
-s
-t
```

If **cs0.e** is run with no command line arguments, the following message will be printed listing the recognized arguments:

“`syntax: cs0.e [-t] [-s] [-v] input [outroot] [-w wavecal]`”

The only required command line argument is input, the name of the file or files to be calibrated. The flags `-v`, `-s`, `-t` may be combined, and they may be put either before or after the file name(s). Nothing except whitespace is allowed between `-w` and the name of the wavecal file, however. The `input`, `outroot`, and `wavecal` may be single file names or file name templates. A file name template is a name containing one or more wildcard characters, or an `@` sign followed by the name of a text file containing a single column of file names. If such a template is used, the string should be enclosed in single quotes to prevent expansion by the shell, unless only one file name matches the template. This is true for all the host-level programs (the **csN.e**). Here are some examples:

```
% cs0.e o47s01020_raw.fits
% cs0.e 'o47s010?0_raw.fits'
% cs0.e 'o47s010[123]0_raw.fits'
% cs0.e '*_raw.fits'
% cs0.e '@input'
% cs0.e '@input' calibrated/
% cs0.e '@input' '@outroot'
% cs0.e '*_raw.fits' -vts
% cs0.e -v '*_raw.fits' -s -w o47s01020_wav.fits
```

The reason for enclosing the name in quotes is that the program must be able to distinguish between `input` and `outroot`, which it does by argument number on the command line. When using the IRAF task **calstis** to run this executable, it is not necessary to enclose the string in quotes, because the cl script does that for you.

`outroot` is an optional root name for output files. Not specifying `outroot` is equivalent to specifying the input file name as `outroot`. `outroot` may be a directory name, in which case it must end in "`/`" to signify that it is a directory. When `outroot` is a directory name, the input file name will be appended to this directory name, and the resulting string will be modified in the usual way to construct output file names. If `outroot` is already a complete file name, such as "`abc_raw.fits`", possibly including a directory prefix, the output file names will be constructed in the usual way by replacing the suffix. If `outroot` is just a root name without a suffix, then the output names will be constructed by appending the output suffixes and the default extension "`.fits`". If `input` is a file name template, and if `outroot` is specified, `outroot` must either be a directory name or there must be the same number of names in `input` and `outroot`.

It is possible to specify a wavecal file to override the value of the WAVECAL keyword in the science file header. The notation "`-w wavecal`" is used, rather than assuming that it is the third argument, because `outroot` is optional. If `input` is a file name template

and `wavecal` is specified, the number of names in the wavecal list can either be one or the same as the number in the input list. If one wavecal is specified, it will be used for all input files.

The "`-v`" flag means "verbose." This is simply passed to the various *calstisN*, and they print some additional information if this flag is specified.

The "`-t`" flag means that strings will be printed giving the date and time at various points during processing. Some date and time information is printed anyway, but "-t" results in more, with a specific format that looks like:

1998042095311-I---------- Begin processing: O3SX01AKR ----------

The number at the beginning is the four digit year, three digit day of year, and the hour, minute, and second of local time. The "`I`" means "information." A short string is then printed to indicate where this occurred during processing, e.g. "`Begin processing`". The string "`O3SX01AKR`" is the value of ROOTNAME from the input header. The remainder of the 80-character line is then filled out with "`-`".

The "`-s`" flag means that temporary files should be saved. Several temporary files with names such as "`o40801090_w2d_tmp.fits`" are created during typical processing, and by default these files are deleted when they are no longer needed. Saving these temporary files is helpful when checking out problems.

4. Names of All Input and Output Files

Since *calstis0* typically writes multiple output files, or different output files depending on the input data and switch settings, the user doesn't explicitly specify the complete name of any particular output file. This section describes the way the output file name is constructed from the input name, and it lists the suffix used for each file that is written, including temporary files.

A typical input file name is something like "`o4e701050_raw.fits`". The "`_raw`" portion, called a suffix, identifies the type of file or level of processing that has been done. Some file name extensions other than "`.fits`" are allowed, such as "`.fit`".; this depends on the IRAF FITS kernel. All the output names are formed by replacing the "`_raw`" suffix by other suffixes. If the input name does not have a suffix, or if the suffix is not "`_raw`", then the output name will be the input name with the output suffix inserted just before the file name extension. For example, if the input name is "`o4e701050_flt.fits`", and the output suffix is "`_crj`", the output name will be "`o4e701050_flt_crj.fits`".

The STIS can take either imaging or spectroscopic data. These are distinguished by the OBSTYPE keyword in the primary header. Imaging data would be a direct image of the sky. Spectroscopic data can be first order, echelle, or prism, though prism is not sup-

ported by the pipeline calibration. The calibration steps that can be performed depend to some extent on whether the data are imaging type or spectroscopic.

Temporary files can be written for three reasons. The first is simply that one of the *calstisN* writes an output that is read by another *calstisN* but would not be considered a "final" output product. Before doing cosmic-ray rejection for CCD images, for example, the overscan subtracted image is written to a temporary file, and that file is read by the cosmic-ray rejection routine. The second reason for creating a temporary file is because of the principle that *calstis0* should not modify its input. Some of the *calstisN* that are called by *calstis0*, however, can modify their input. For example, *calstis2* can set data quality flags in its input file to mark the positions where cosmic rays were found. The calibration switches in a file could be set such that the first step to be performed would be CRCORR, and in this case *calstis0* would copy its input file, unchanged, to a temporary file before calling *calstis2*. Finally, the output from processing the wavecal (if that step is done), is written to temporary files.

The following is a list of suffixes used by *calstis0*, along with an explanation of the contents of each of the files with those suffixes. Suffixes that end in "`_tmp`" are for temporary files. Generally speaking, "`f`" in the suffix means "flat fielded," "`x`" means "extracted," "`s`" means "summed," and "`w`" means that the file is a wavecal.

Table 2: Suffixes for STIS Data Files

suffix	description
_raw	The input file to be calibrated. This is referred to as the science file to distinguish it from the wavecal.
_wav	The input wavecal file, if any, associated with the science file. The wavecal name need not use this suffix, and in fact, *calstis0* never actually assumes or makes use of this suffix.
_crj	The cosmic-ray rejected and flat fielded science data. This is output from *calstis1* after calling *calstis2*.
_flt	This is a flat fielded science file, output from *calstis1*. If cosmic-ray rejection is done and EXPSCORR is PERFORM, however, this is an intermediate output product. This is the file that has cosmic rays flagged but not rejected; it contains as many imsets as in the input raw file. (If EXPSCORR is not PERFORM, this file would not be created.) Note that the "final" flat fielded output will be either the _crj or _flt file, depending on whether cosmic-ray rejection was done or not.
_x1d	The 1-D extracted spectroscopic data, output from calling *calstis6* on the _flt file. The _x1d file contains a FITS BINTABLE extension for each input imset, with one row per input spectral order. This file is not created if cosmic-ray rejection was performed; see _sx1.
_x2d	The 2-D extracted data (either spectroscopic or imaging), output from calling *calstis7* on the _flt file. The _x2d file contains as many imsets as the input. This file is not created if cosmic-ray rejection was performed; see _sx2.

suffix	description
_sx1	The 1-D extracted, summed spectroscopic data, output from calling *calstis6* on the _crj file. Like the _x1d file, this is a FITS BINTABLE, but it contains only one imset. This file is only created from cosmic-ray rejected input.
_sx2	The 2-D extracted, summed data (either spectroscopic or imaging). If cosmic-ray rejection was done, this is the output of *calstis7* on the _crj file. If not, this is the result of calling *calstis8* on the _x2d file that was written by *calstis7*.
_sfl	Summed flat fielded data, output from calling *calstis8* on the _flt file. This is only created if 2-D rectification is not performed.
_blv_tmp	CCD science data after bias overscan subtraction. This is the output from *calstis1* when DQICORR and BLEVCORR are set to PERFORM, and it is the input to *calstis2*. If EXPSCORR is PERFORM, this file is also the input to *calstis1*, with output to the _flt file.
_crj_tmp	The cosmic-ray rejected science data. This is the output from *calstis2* before any further processing.
_fwv_tmp	The flat fielded wavecal, output from *calstis1*.
_cwv_tmp	For CCD wavecals taken with the HITM system, the science data is subtracted from the wavecal by *calstis11*. This is the output from *calstis11*, and it is input to *calstis7*.
_w2d_tmp	The 2-D rectified wavecal, output from *calstis7*.

5. What Steps are Called, and Under Which Conditions

Since **cs0.e** is the pipeline version of calstis, it should be no surprise that it determines what calibration steps to perform and what reference files to use by reading the primary header of the input file. Reference file names are discussed in section 7. The other **csN.e** use command line arguments to specify what steps to perform; although, in some cases (see below) they also read the input header to check whether the specified steps make sense, depending on the type of data and what has already been done to the data. Setting the calibration switches to "COMPLETE" and adding history records is handled by the *calstisN* modules. This is partly because the output files are written by the *calstisN*, and it makes more sense to change keywords and add history records before the output header is written for the first time.

Below is a list of the calibration switches that are used by *calstis0*. Some of these switches are secondary, in the sense that they are only used if another switch is PERFORM.

Table 3: Calibration Switch Names

keyword	calstis module	conditions and comments
ATODCORR	cs1	CCD only; currently not performed
BIASCORR	cs1	CCD only
BLEVCORR	cs1	CCD only
CRCORR	cs2	CCD only; more than one imset
EXPSCORR	cs0	
DARKCORR	cs1	
DOPPCORR	cs1	MAMA only; never reset to COMPLETE
DQICORR	cs1	
FLATCORR	cs1	
GEOCORR	cs7	imaging type only; currently not performed
GLINCORR	cs1	MAMA only
LFLGCORR	cs1	MAMA only
LORSCORR	cs1	MAMA only
PHOTCORR	cs1	imaging type only
RPTCORR	cs8	more than one imset; replaced by CRCORR if CCD
SHADCORR	cs1	CCD only; currently not performed
WAVECORR	cs4, cs12	spectroscopic type only
X1DCORR	cs6	spectroscopic type only
BACKCORR	cs6	(only if x1dcorr)
DISPCORR	cs6	(only if x1dcorr)
FLUXCORR	cs6, cs7	in cs6, must also perform BACKCORR and DISPCORR
HELCORR	cs6, cs7	must also perform DISPCORR
SGEOCORR	cs6	currently not performed
X2DCORR	cs7	spectroscopic type only
SGEOCORR	cs6, cs7	currently not performed
STATFLAG	cs1, cs7, cs8	boolean T or F; not reset after complete

The “conditions” in the above table are used as a sanity check to reset switches (during execution, not by changing the header) that were set to PERFORM but should not have been, i.e. performing the step wouldn’t make sense for the particular combination of detector and grating.

During wavecal processing, the X2DCORR switch is assumed; that is, 2-D rectification will be performed regardless of the value of the X2DCORR switch. This is simply because *calstis4* would give incorrect results if its input were not 2-D rectified. In fact, *calstis4* will refuse to run on data that does not have X2DCORR set to COMPLETE. Another characteristic of wavecal processing that is not obvious is that no calibration switch is used to determine whether *calstis11* should be called (to subtract the science file from the wavecal). *calstis11* is called if the science data were taken with the CCD using the HITM system; this depends on the DETECTOR and SCLAMP keywords. This will need to be changed in the future to make use of a new keyword giving the status of the external shutter.

The **cs2.e** and **cs6.e** programs allow the user to specify a number of parameters and/or a reference file on the command line. When called from *calstis0*, however, the default calling sequences are used for *calstis2* and *calstis6*, so the reference file names are read from the input header, and the values of parameters are read from the reference files.

Here is a simplified outline of the processing in *calstis0*.

If either CRCORR or RPTCORR is PERFORM for CCD data, cosmic-ray rejection will be performed. The first step is to call *calstis1* to initialize the error (ERR) array, to subtract and strip off the overscan, and to update the header keywords ATODGAIN and READNSE with values read from the CCD parameters table. *calstis2* is then called to reject cosmic rays and sum the input data.

calstis1 is then called to perform basic 2-D reduction. (Some of this may already have been done prior to cosmic-ray rejection.) If EXPSCORR is PERFORM, *calstis1* will also be called on the input file to *calstis2*, since the latter has flagged cosmic rays in that file.

For spectroscopic data with WAVECORR set to PERFORM, *calstis1* is first called on the wavecal. *calstis11* will then subtract the science data from the wavecal if the data were taken with the CCD using the HITM system (note that this step does not depend on a calibration switch keyword). Then the wavecal will be 2-D rectified by calling *calstis7*; this is done regardless of the X2DCORR switch. *calstis4* is then called to find the shift in each axis and to update the SHIFTA1 and SHIFTA2 keywords in the SCI extension headers. Finally, *calstis12* reads the SHIFTA1 and SHIFTA2 keywords from the wavecal and updates those keywords as well as CRPIX1 and CRPIX2 in the SCI extension headers of the science file.

If X1DCORR is PERFORM, *calstis6* is called to extract a 1-D spectrum. If either X2DCORR or GEOCORR is PERFORM, *calstis7* is called to perform 2-D rectification of the spectral data or imaging data respectively. If RPTCORR is PERFORM and the data were not already combined for cosmic-ray rejection, *calstis8* will be called to sum the repeated exposures.

Rectification of imaging data is not done in the pipeline. *calstis7* includes the code to do this, and it can be done off-line if a suitable reference table (IDCTAB) has been made.

If a serious error occurs (e.g. a missing reference file), a message beginning with "ERROR" will be printed, and processing will stop. For less serious cases (e.g. setting BLEVCORR to PERFORM for MAMA data), a warning message will be printed, and processing will continue. Other messages are also written for information to the user, such as the date and time of processing.

A warning message is printed each time a keyword is added to an output header. All relevant keywords are supposed to be present already in the input file (from which the output is copied). This warning is a reminder to the Spectrographs group to submit the documentation to have the keyword added to the input file and to the keyword database.

6. Rerunning *calstis* on its Own Output File

One of the design criteria for *calstis* was that it should be possible to perform a subset of the calibration steps, and then pass the resulting output file to *calstis* again to perform additional calibration steps if desired. This was implemented on two levels, (1) by having individual executables for major segments of the calibration, and (2) by allowing the output of some of those modules to be used as input to the same modules. In practice, there are limitations on this capability. *calstis1* is the only module for which the second option really makes sense. It can be done to some extent with *calstis0* by setting the header switches; however, it would be better to run the individual executables, because it is easier to control and easier to understand what is being done. One major limitation, though not of the software, is that if the steps are done out of order, the results will probably be incorrect.

It was mentioned above that some of the **csN.e** reset some calibration switches, depending on keywords in the input header, rather than relying entirely on the command line arguments. (Note that we're talking about **csN.e** here, not **cs0.e**.) This is done for several reasons. The first is simply that the default is to perform all steps, but some steps may be inappropriate, depending on detector or mode. Another is that a step may already have been done, and it wouldn't be sensible to repeat it. In **cs1.e** the steps that can be repeated are DQICORR and PHOTCORR, and the steps that cannot be repeated are ATODCORR, BIASCORR, BLEVCORR, DARKCORR, FLATCORR, GLINCORR, LFLGCORR, and SHADCORR. *calstis1* checks the latter switches in the input header and will reset a command line switch if the value in the header is COMPLETE. The **cs4.e** and **cs12.e** parts of wavecal processing can be repeated. The output of 1-D extraction is a table, not an imset, so **cs6.e** cannot be rerun with its output used as input. With the current reference files, the output of 2-D rectification is significantly different from the input, so rerunning **cs7.e** would give incorrect results, but there's nothing to prevent a user from doing so.

7. Reference File Names

For each calibration switch that is set to PERFORM, there may be calibration reference files required to perform that step. For CCD observations, a CCD parameters table is also required. If any required reference file is missing or can't be read, no processing will be done before returning with an error. This requirement was driven by the pipeline environment; a missing reference file is a common problem, and if checking were not done at the beginning, it could waste CPU time and leave partially processed files on disk.

calstis0 checks for the existence of reference files needed by all the *calstisN* that it will call, both for the science file and for the wavecal. It checks by actually opening the file read-only as an image, and if that fails, as a table, so the check is on accessibility, not just existence. *calstis0* does not check PEDIGREE, however; that is left up to the individual *calstisN*.

Not all reference file keywords are checked, because it is allowable that some be not specified. The term "not specified" in this context means either blank or "`N/A`" (case insensitive); N/A stands for not applicable. For every reference file keyword, however, if a value is specified then it is an error if the file doesn't exist. Keywords that may be left blank or N/A are WBIAFILE, BPIXTAB (if CCD), PFLTFILE, DFLTFILE, LFLTFILE, PCTAB and APERTAB. WBIAFILE may be blank or absent from the science header for backward compatibility. For CCD data, DQICORR could be done just to flag nonlinearity, rather than bad pixel initialization, in which case the BPIXTAB need not be specified. For the flat field correction, any one, two, or all three of PFLTFILE, DFLTFILE, LFLTFILE may be specified. If none is specified it is an error. PCTAB and APERTAB are allowed to be blank or N/A for backward compatibility. PCTAB tables have also not been created yet for all modes. PCTAB is used in the FLUXCORR step in *calstis6* and *calstis7*. APERTAB is used for FLUXCORR, but it may also be needed for PHOTCORR for imaging type data. While APERTAB may be optional for PHOTCORR, it is not optional for FLUXCORR, and it is a limitation of *calstis0* that it does not distinguish between these cases.

Some reference files will be needed for calibrating the wavecal but not for the science file; the names of those files will nevertheless be available as keywords in the science file as well as in the wavecal. There are a couple of reasons for this, one being that it should not be necessary for the user to set reference file names in both the science file and wavecal headers when an update is needed. The science file is the one that is more visible (the wavecal name is given in the science file header), so updating keywords in the science header alone should be sufficient. This means not only that the keywords must be present in the science header, but also that the value in the science header must override that in the wavecal in case of a discrepancy. So *calstis0* reads the names of all reference files that it needs from both the science file and wavecal headers. For each reference file needed for the wavecal, *calstis0* checks that the name gotten from the science file header is the same.

If they differ, a warning is printed, and the name from the science file header is used for processing the wavecal as well. For backward compatibility, LAMPTAB may be blank in the science file header. Also, for CCD data, the science file and wavecal may be binned differently or have different gain, in which case different bias files would be appropriate. Therefore, one can't simply compare BIASFILE values. The name of the bias file for the wavecal should be recorded in the science file header with the keyword WBIAFILE; this is the keyword that will be compared with BIASFILE in the wavecal. The rule for comparing these keywords is as follows. If the binning and gain of the science file and wavecal are the same, then BIASFILE in the science file header will be used for calibrating both the science file and wavecal; otherwise, if the WBIAFILE keyword is present and a value was specified, that value will be used for calibrating the wavecal.

Note that overriding wavecal keywords with science file keywords, while convenient for the user and for the archive, can result in subtle differences in processing when running **cs0.e** to calibrate a science file with a wavecal, as compared with running **cs1.e**, etc., on the science file and wavecal separately. The reason is simply that in the latter case, no comparison of reference file names can be made, so different reference files can be used for the wavecal.

8. Appendix

STIS data can be binned and/or they may cover just a subset of the full detector. Information about this is critical in several steps in *calstis*. For example, when applying a flat field, *calstis* must extract a subset of the reference image and bin it to match the raw image. There are keywords in the primary header to record what was specified in the proposal instructions, but these values remain fixed as the data are changed, for example by removing the CCD overscan or binning high-res MAMA data to low-res. Rather than inventing new keywords to convey the binning and subset information, we use the same keywords that the IRAF mini world coordinate system (MWCS) package uses for this purpose. These keywords are LTV1, LTV2, LTM1_1, and LTM2_2 (the cross terms LTM1_2 and LTM2_1 should always be zero for STIS data), and they are found in the extension headers. The names refer to "linear transformation vector" and "linear transformation matrix." These terms give the mapping from a reference coordinate system to the pixel coordinates in the image:

X = Xref * LTM1_1 + LTV1

Y = Yref * LTM2_2 + LTV2

For STIS, the reference coordinate system is detector pixels. For the CCD, our convention is that this means unbinned pixels starting at the first illuminated pixel (i.e. not overscan). For the MAMAs, this means 1024x1024 low-res pixels. In IRAF notation, the

first pixel is number one (i.e. not zero indexed). The pixel number is an integer at the center of the pixel, so the first pixel runs from 0.5 to 1.5.

From the expressions above, one can see that the reference pixel size is LTM1_1 by LTM2_2 in units of the image pixel size, and (LTV1,LTV2) is the location, in the image pixel coordinate system, of pixel zero of the reference coordinate system. "Pixel zero" in this context does not mean the first pixel using zero indexing; it means the location one pixel to the left of and below the first pixel. For the MAMAs, this is off the detector, but for the CCD using amp D readout, this is within the overscan region.

A number of IRAF tasks either update or make use of these keywords, tasks such as **blkavg, blkrep,** and **listpixels**. By setting **listpixels.wcs** to "`physical`", for example, the pixel coordinates that are printed will be in the STIS reference coordinate system.

Tables 4, 5, and 6 show the values of LTVi and LTMi_i for full-format data (i.e. not subarray) for the MAMA and CCD. The values for the CCD in tables 5 and 6 are for the case that the overscan regions (both serial and parallel) have been removed; see below for values including the overscan. These are values that may be seen in calibrated data and in reference images. For the CCD, values for the first and second axes are shown in separate tables to emphasize that they are independent, and there are also significant differences between the two axes.

Here is a table of LTV and LTM values for MAMA detectors. In the column headings, i is either 1 or 2 for the first or second axis respectively. Binning more than low-res is included just for example.

Table 4: LTV and LTM for MAMA detectors

binning	NAXISi	LTVi	LTMi_i
high-res	2048	-0.5	2.0
low-res	1024	0.0	1.0
2	512	0.25	0.5
4	256	0.325	0.25
8	128	0.4375	0.125

Here are two tables for the CCD detector. The first gives LTV1 and LTM1_1 (i.e. for the first image axis), and the second gives LTV2 and LTM2_2. Note that these values are for the case that the overscan regions have been removed; see below for values including the overscan.

Table 5: LTV1 and LTM1_1 for the CCD detector

amp	binning	NAXIS1	LTV1	LTM1_1
any	1	1024	0.0	1.0

amp	binning	NAXIS1	LTV1	LTM1_1
any	2	511	-0.25	0.5
A or C	4	255	0.125	0.25
B or D	4	255	-0.375	0.25

Table 6: LTV2 and LTM2_2 for the CCD detector

amp	binning	NAXIS2	LTV2	LTM2_2
any	1	1024	0.0	1.0
any	2	512	0.25	0.5
any	4	256	0.375	0.25

The complication for binned CCD observations is that the physical overscan region is 19 pixels, a prime number. For binned data, near each end of the line there is a pixel that consists of data from both the illuminated and overscan regions. These pixels are removed with the overscan when subtracting the bias level, so the first pixel in the output image does not begin at the beginning of the illuminated portion. These mixed pixels result in some peculiar values of LTV1 and LTV2, and these values can depend on which amplifier was used for readout.

Here are specific values of LTVi and LTMi_i for a full frame CCD image, including the overscan regions, for different binning and readout amplifiers.

For the case of no binning, the axis lengths are 1062 and 1044; LTM1_1 = LTM2_2 = 1.; the line contains 19 overscan, 1024 illuminated, 19 overscan pixels.

Table 7: No Binning

amp	LTV1	LTV2
A	19.0	0.0
B	19.0	0.0
C	19.0	20.0
D	19.0	20.0

For binning of 2 by 2, the axis lengths are 532 and 522; LTM1_1 = LTM2_2 = 0.5; the line contains 9 overscan, 1 mixed, 511 illuminated, 1 mixed, 10 overscan pixels.

Table 8: Binning 2 by 2

amp	LTV1	LTV2
A	9.75	0.25
B	10.75	0.25

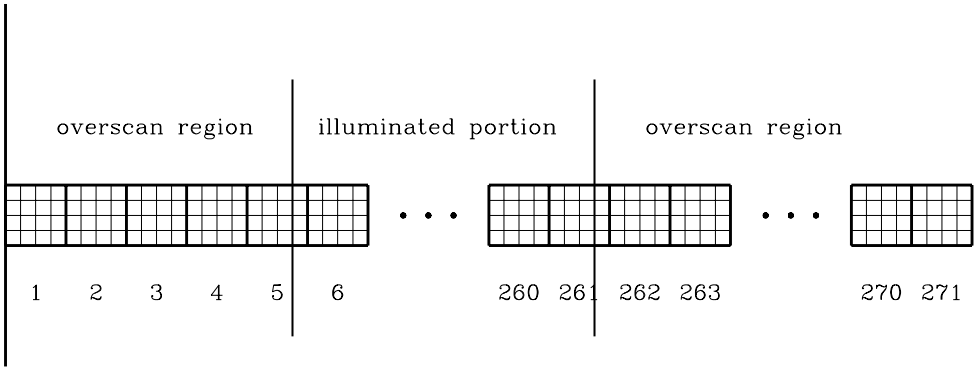
amp	LTV1	LTV2
C	9.75	10.25
D	10.75	10.25

For binning of 4 by 4, the axis lengths are 271 and 266; LTM1_1 = LTM2_2 = 0.25; the line contains 4 overscan, 1 mixed, 255 illuminated, 1 mixed, 10 overscan pixels.

Table 9: Binning 4 by 4

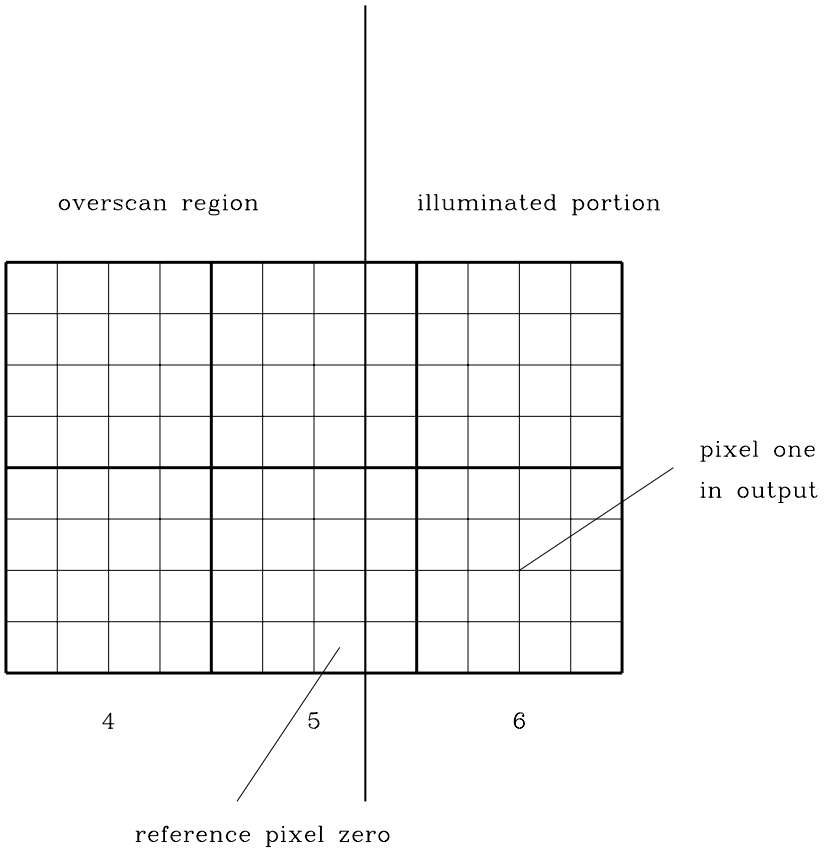
amp	LTV1	LTV2
A	5.125	0.375
B	10.625	0.375
C	5.125	10.375
D	10.625	10.375

As a specific example, consider CCD data binned by four in the first axis. The length of the first axis is 1044 / 4 + 10 = 271 pixels, which breaks down as follows for amplifier A: four overscan pixels, one mixed overscan/illuminated, 255 illuminated pixels, one mixed, and ten overscan (the readout is continued beyond the physical overscan). This is illustrated in figure 1, which shows one line of an image binned 4 by 4. Three portions of the line are shown, with the two missing regions indicated by ellipsis. The edges of the binned pixels are drawn with bold lines, and the edges of the unbinned pixels are drawn with light lines. The binned pixel numbers are shown below the pixels.



Here are details for understanding the value of LTV1 for amp A readout. Figure 2 shows a small region which is crossed by the boundary between the overscan and illuminated regions, with pixels binned by four in each axis. The region shown includes binned pixels 4, 5, and 6 along the first axis, and pixels 1 and 2 along the second axis; the binned

pixel numbers are shown below the pixels. When the overscan region is stripped off, the mixed illuminated/overscan column will also be removed, so binned pixel six from the raw data will become the first output pixel in the first axis. Pixel one in the reference coordinate system is the first unbinned illuminated pixel, which is the 20th pixel, the one just to the right of the vertical line in the figure separating the overscan from the illuminated regions. Thus, pixel zero in the reference coordinate system is the pixel just to the left of that line. In binned coordinates, that point is located 1/8 binned pixel to the right of pixel five; therefore, LTV1 is 5.125. When the overscan is removed, five pixels will be removed from the beginning of the line (four pure overscan and one mixed), so LTV1 for the output image will be 5.125 - 5.0 = 0.125.



9. References

Hodge, P., & Baum, S. 1995, STIS Instrument Science Report 95-06

HST Data Handbook, 1997, Volume 1