The 2005 HST Calibration Workshop
Space Telescope Science Institute, 2005
A. M. Koekemoer, P. Goudfrooij, and L. L. Dressel, eds.

Multidrizzle and Tweakshifts: Overview and Future Plans

A. M. Koekemoer[1], A. S. Fruchter[1], R. N. Hook[2], W. Hack[1], C. Hanley[1]

Abstract. The MultiDrizzle software provides a flexible, unified interface to the various steps involved in registering, cleaning and combining dithered *HST* data. MultiDrizzle can be run on ACS, WFPC2, NICMOS and STIS imaging data in a single step, with default parameters chosen to provide good results for the majority of datasets if standard recommended dithering patterns were followed. Alternatively, the various parameters can be adjusted and specific steps can be run one at a time for additional flexibility, if desired for certain scientific applications. MultiDrizzle has also been incorporated into the *HST* OTFR pipeline for ACS data, thereby allowing automatic delivery of cleaned, combined data that can be used directly for science in many cases, or alternatively used as a baseline for re-running MultiDrizzle off-line using customized parameters, if required for certain scientific applications. Here we review the basic fundamentals of dithering, together with the current status of MultiDrizzle and describe plans for future related pipeline enhancements, including the Tweakshifts shift-refinement script.

1. Introduction

A widespread technique in obtaining Hubble Space Telescope *(HST)* observations involves the use of "dithering", or spatially offsetting the telescope in order to move targets to a number of different locations on the detector. There are a number of different scientific drivers for employing dithering, with different benefits/trade-offs depending on the size and layout of the dither steps and the scientific goals of the observations, as described in detail in the *HST Dither Handbook* (Koekemoer et al. 2002a). Here we review these considerations, and describe current progress with the MultiDrizzle software (Koekemoer et al. 2002b) which has been developed to provide a flexible, unified interface to the various steps involved in registering, cleaning and combining dithered *HST* data.

One of the most common reasons for dithering is to mitigate the relatively severe undersampling of the *HST* point-spread function (PSF) by the pixels in most of the detectors onboard *HST*. The undersampling imposes strong limitations on the accuracy with which morphological and photometric measurements can be obtained from exposures at any given single location. By obtaining a series of exposures shifted by non-integer pixel offsets, the information can be sampled on scales that are finer than the detector pixel size, thereby permitting the construction of a more finely sampled image when the individual exposures are combined. This is the motivation behind the "drizzle" software (Fruchter & Hook 2002), which is called by MultiDrizzle and makes use of linear reconstruction algorithms to allow input images to be resampled onto a finer output pixel grid.

In addition, the sensitivity of each pixel varies as a function of position across the pixel, hence the undersampling can lead to problems when pursuing very accurate stellar photometry since the measured flux of a star may vary significantly depending on how it is

---

[1] Space Telescope Science Institute, 3700 San Martin Dr., Baltimore, MD 21218, USA

[2] Space Telescope – European Coordinating Facility, Karl-Schwarzschild-Str. 2, D-85748 Garching, Germany

centered on a pixel. Sub-pixel dithering helps mitigate this intra-pixel sensitivity variation by allowing the pixel response function to be sampled more completely, thereby resulting in more robust photometric measurements.

Furthermore, dithering on larger scales to move targets to different locations on the detector can help improve the signal-to-noise ratio (S/N) by averaging out variations in flat field sensitivity, including variations in sensitivity from one pixel to the next as well as large-scale background variations, that may lie below the level of accuracy of the flat fields and other reference files. If no dithering is performed, then these uncertainties can act as systematic errors in any determination of photometric properties, hence they can be mitigated by averaging them out when the targets are moved around to different locations on the detector.

Another reason for dithering is to move bad pixels or detector blemishes around to different parts of the sky, so that any given object will not be too severely affected by them. This also includes moving the gaps that may be present between multiple detectors in the same instrument, for example the two chips of the Advanced Camera for Surveys / Wide Field Channel (ACS/WFC), or the four separate cameras of the Wide Field Planetary Camera 2 (WFPC2). This is often done when it is desirable to obtain reasonably uniform coverage across the field. Although the resulting depth in the gaps may be slightly shallower than the surrounding coverage, this is often preferable to having no coverage at all.

Finally, very large-scale offsets on the scale of the detector itself can be used if it is necessary to cover a contiguous region that is significantly larger than the detector size, thereby creating a "mosaic" of the target field. Strictly speaking, such offsets address a different class of issues than the smaller-scale dither offsets discussed above, but they are included in the discussion since many of the same technical considerations apply to them as well.

2. Dither Strategies in Practice

When planning to combine dithered data with MultiDrizzle, it is important to first plan the observations in a way that will maximize the scientific return. An important issue that should be taken into account when considering dither patterns is the degree of geometric distortion across the detectors, which is particularly severe for ACS/WFC and ACS/HRC that have about a $\sim 7\%$ plate scale change across the detector, in addition to substantial skew. The plate scale change means that a relatively small offset at the center, for example 7 pixels, gradually changes to a different offset across the chip until it contains an additional non-integer component of ~ 0.5 pixels at the corner. Similarly, a larger offset of 14 pixels at the center would gradually change to an additional 0.5 pixels halfway to the corner, then continue changing back to integer sampling by the time the corner is reached. Even larger dithers introduce additional cycling between sub-pixel and integer offsets between the center and corner of the chip. In reality it is somewhat more complicated since the ACS distortion contains a significant skew term in addition to the scale change, which means that the degree of non-integer pixel sampling varies across the detector in an even more intricate pattern for any given dither offset.

This has implications when considering the design of a dither pattern to choose for a given science program. If retaining absolutely uniform sub-pixel sampling across the entire detector is the most important consideration, for example in very critical photometric studies, and if some loss of coverage can be tolerated due to bad pixels and chip gaps, then it is plausible that dither offsets may be chosen that are entirely less than 1 pixel in extent. In this case the largest differential change in sampling introduced by the $\sim 7\%$ geometric distortion across the detector would correspond to ~ 3.5 milliarcseconds for the ACS/WFC and ~ 1.7 milliarcseconds for the ACS/HRC, which becomes comparable to the pointing uncertainty of *HST*. Such a strategy would not move bad columns or chip gaps around to sufficiently different locations, hence resulting in a loss of coverage at those locations, but this may be tolerated if uniform sub-pixel sampling is critically important.

A more common strategy involves combining sub-pixel offsets with larger integer pixel offsets, to ameliorate the effects of hot pixels and bad columns as well as the ACS/WFC chip gap. In this case, the $\sim 2.5''$ offset along the y-axis that is needed to cover the chip gap corresponds to ~ 50 pixels, and thus will most certainly introduce variable amounts of sub-pixel sampling along the y-direction, particularly if multiple exposures are used that cover the gap in increments of 50 pixels. This is generally mitigated by using a dither pattern that fully samples 4 different pieces of sub-pixel phase space, such as the pre-defined 4-point BOX dither, or a 2-point primary LINE dither with an additional 2-point secondary LINE dither at each of the two pointings. By obtaining a sufficient number of dither pointings which cycle through sub-pixel and integer pixel sampling in different ways across the detector, it is generally possible to ensure reasonably uniform sub-pixel sampling across the entire field. This strategy is commonly used for extragalactic field studies where the targets are generally resolved, where it is important to obtain some degree of sub-pixel sampling, but equally important to move hot pixels and bad columns to different locations, as well as to cover the ACS/WFC chip gap. We note that these issues are not as critical for WFPC2 which only has $\sim 1-2\%$ distortion; however, one of its cameras (the PC) has about half the pixel scale of the other three cameras, thereby necessitating a special dither pattern optimized to provide sub-pixel sampling in all the cameras simultaneously.

Generally, the 4-point dither patterns which include half-pixel offsets provide sufficient sub-pixel sampling for almost all the observing programs on *HST*. This is because the degree of undersampling by the pixels is typically no more than about a factor of 2 at worst for ACS/WFC, while the ACS/HRC provides sampling that is essentially Nyquist-limited toward the red part of the spectrum. Furthermore, the ACS/WFC requires a few minutes to read out one exposure, which places a natural limit on the number of separate dither positions that can be obtained in one orbit for small to medium-sized programs. In addition, the typical pointing uncertainty of *HST* is in the range $\sim 2-3$ milliarcseconds (Gilliland et al. 2005; Koekemoer et al. 2005a, and these proceedings), or $\sim 25\%$ of a half-pixel dither with ACS/HRC, thus it becomes infeasible to expect dither patterns to execute perfectly if they attempt to sample scales much finer than 1/2 or 1/3 of an HRC pixel. Some of the deeper observing programs, such as the Hubble Ultra Deep Field, used 12-point patterns (effectively a 4-point box pattern replicated 3 times, with slight additional shifts, randomized to some extent by the *HST* pointing). The primary effect of this is to ensure more uniform sub-pixel sampling across the entire field, particularly when adding in the small random offsets of a few milliarcseconds that are a consequence of the guide star re-acquisitions over large numbers of orbits.

3. MultiDrizzle - Combining Dithered Images

Once a series of dithered images have been obtained, several steps need to be carried out in order to combine them. First, their relative shifts need to be determined, for which the *HST* header astrometry is generally accurate enough to be used, if the exposures were all obtained in the same visit using the same guide stars. The typical offset and pointing accuracy of *HST* is $\sim 2-3$ milliarcseconds when there are no guide star problems, which is thus at the level of 0.1 pixel for ACS/HRC and 0.05 pixel for ACS/WFC (and 0.02 pixel for the WFPC2/WF chips), and generally below the level of other effects such as PSF-related changes. Occasional guide star problems can increase these errors, and certain science programs demand more accurate alignment, as well as cases where data need to be combined that were taken in different visits using different guide stars (in which case their astrometry could be offset by up to a few arcseconds). For such applications we describe later the Tweakshifts script, which can directly use the contents of the images themselves to improve their alignment.

After the image shifts have been determined, they need to be geometrically transformed to a set of common, registered output images where all the sources are coincident in pixel space. This permits the creation of a clean approximation to the final image, by evaluating quantities such as the median value of all pixels at a given location, which mitigates the contribution from outliers such as cosmic rays or hot pixels. Once this clean image has been created, it is then transformed back to the frame of each of the input exposures. Comparing the clean image with each input exposure then allows the identification of cosmic rays and bad pixels, which can be incorporated into a pixel flag mask. The pixel masks are then used in the final "drizzle" step, which maps all the input exposures onto the output frame, and combines them by performing a weighted sum of their pixel values, using the pixel masks to eliminate the contribution from rejected pixels.

The MultiDrizzle script was developed to provide a flexible, unified interface to all the above steps. Prior to the availability of MultiDrizzle, all these steps needed to be carried out manually using a variety of different IRAF scripts, with a significant amount of detailed book-keeping required to ensure that all the correct intermediate files were present. This was burdensome even for small and intermediate programs, and became prohibitive for large programs. The goal of MultiDrizzle is to remove the overhead of keeping track of all the intermediate information, while still retaining the ability to run each of these steps separately. It provides enough parameters to permit detailed control of the behavior of each step if required, while also providing default values for these parameters that allow it to be run in a "one-touch" mode, which can start with a set of calibrated exposures and automatically produce a cosmic-ray cleaned, distortion corrected, drizzled combined output image, by invoking a single command.

Data that were obtained using standard recommended dither patterns can be provided to MultiDrizzle to carry out all the steps of sky subtraction, registration, cosmic ray rejection and final drizzle combination all in a single command, simply by specifying the list of input files, for example:

```
--> multidrizzle input=*_flt.fits output=outputfilename
```

The other parameters can be specified on the PyRAF command line or alternatively can be edited using the standard IRAF 'epar' mechanism before running the task. MultiDrizzle is designed to carry out the following steps, either in a single pass or alternatively by selecting various steps individually:

1. Staticmask - Identify negative bad pixels, based on examining all the images, and include them in the dq file
2. Skysub - Sky-subtract each frame
3. Driz_separate - Drizzle the input images onto separate, registered outputs (using shifts computed from the headers)
4. Median - Create a median image from the separate drizzled images
5. Blot - Blot the median image back to the original input frames
6. Driz_cr - Use each blotted image to create a derivative image, and compute CR masks
7. Driz_combine - Do the final drizzle combination

The parameters to each of these steps, as well as the specifics of their behavior, are described in more detail in the on-line documentation available within PyRAF for MultiDrizzle. There are also "startup" parameters which include whether or not to specify an output reference frame, or whether to use a shiftfile that may have been generated by Tweakshifts. Here we summarize the parameters for each of the seven steps mentioned above, pointing out some of the relevant issues that may need to be considered for various scientific applications.

3.1. Create the “Static” Mask of Negative Pixels

Parameters:

`static_sig`	=	4.0	Sigma value to use in flagging negative pixels

This step goes through each of the input images, calculates the r.m.s value for each chip, and identifies pixels that are below the median value by more than some number times the r.m.s. This is aimed at identifying pixels that may have high values in the dark frame that is subtracted during calibration, but may not necessarily have high values in the images, thus the subtraction gives them strongly negative values. Such pixels are not always flagged in the data quality array, hence this step allows them to be identified. Sometimes such pixels fall on bright objects so they would not be negative, but instead would be positive although lower than surrounding pixels. However, if the images are dithered then they should land on blank sky at least some of the time, in which case they will appear negative and will be flagged. The reason for identifying such pixels is to avoid problems later on when creating the median image.

3.2. Perform Sky Subtraction

Parameters:

`skywidth`	=	0.1	Bin width for sampling sky statistics (sigma)
`skystat`	=	’median|mode|mean’	Sky correction statistics parameter
`skylower`	=	INDEF	Lower limit of usable data for sky (in DN)
`skyupper`	=	INDEF	Upper limit of usable data for sky (in DN)
`skyclip`	=	5	Number of clipping iterations
`skylsigma`	=	4.0	Lower side clipping factor (in sigma)
`skyusigma`	=	4.0	Upper side clipping factor (in sigma)
`skyuser`	=	”	Header keyword containing sky value

This step calculates the sky value, using iterative sigma-clipping if specified. For instruments with multiple detectors, such as ACS/WFC and WFPC2, it calculates the sky separately for all the detectors, then chooses the lowest value and subtracts that from all the detectors. The reason for this is that the presence of bright sources on one detector can create a biased value of the sky measurement, thus the more accurate sky measurement is always taken to be the one that is lowest, since that is the least affected by any potential bright sources. If the ’workinplace’ parameter is set to ’no’ (the default), then MultiDrizzle will create copies of the input files and subtract the sky from those, leaving the original files unsubtracted. Setting this parameter to ’yes’ will cause sky to be subtracted from the original input exposures.

3.3. Create Separate Drizzled Images

Parameters:

`driz_sep_outnx`	=		Output image x-size
`driz_sep_outny`	=		Output image y-size
`driz_sep_kernel`	=	’square|point|gaussian|turbo|...’	Drizzle kernel
`driz_sep_wt_scl`	=	’exptime|expsq’	Weighting factor
`driz_sep_scale`	=	’INDEF’	Output pixel size (arcsec)
`driz_sep_pixfrac`	=	1.0	Drop size, in input pixels
`driz_sep_rot`	=	INDEF	Output y-axis position angle
`driz_sep_fillval`	=	INDEF	Value for undefined pixels
`driz_sep_bits`	=	0.	Flag values considered good

This step drizzles the input images onto separate output images. By default it uses the *drizzle* ‘turbo’ kernel, and *drizzle* parameters of `pixfrac` = 1 and `scale` = 1. Specifying ’INDEF’ for the scale means that it will set the output pixel size to the native pixel scale of the camera, whatever that may be. These values can be changed depending on the scientific goals; for example, masks can be substantially improved by specifying a smaller value of `scale`, with the trade-off being larger images (their size increases as the inverse square of the value of `scale`), and increased computation time. The 'bits' parameter can be set to non-zero values if there are specific classes of pixel flag values that can be considered good. For example, the ACS data quality arrays contain large numbers of pixels with bit values of 32 and 64, which are often good in the images, thus setting 'driz_sep_bits' = 96 will allow both of these types of pixels to be treated as good, which means they will contribute to creating the median image in the next step. This may often be desirable if only a few exposures are being combined, where as many good pixels as possible are needed.

3.4. Create the Median Image

Parameters:

`median_newmasks`	=	yes	Create new masks?
`combine_type`	=	'minmed|average|median'	Type of combine operation
`combine_nsigma`	=	4 3	Significance for min. vs median
`combine_nlow`	=	0	Number of low pixels to reject
`combine_nhigh`	=	1	Number of high pixels to reject
`combine_lthresh`	=	INDEF	Low threshold for clipping
`combine_hthresh`	=	INDEF	High threshold for clipping
`combine_grow`	=	1.0	Radius for neighbor rejection

This creates a median image from the separate drizzled input images, allowing a variety of combination and rejection schemes. If `combine_type` is set to 'median' or 'average', then the routine behaves similarly to the IRAF task *imcombine*, using the values of `combine_nlow` and `combine_nhigh` (the number of low and high pixels to reject) and `combine_grow`, the amount by which flagged pixels can grow. If `median_newmasks` = 'yes', then pixels are flagged using the static bad pixel masks. If this parameter is 'no' then this step will simply use whatever masks are specified in the 'BPM' header keyword of each image (which could be created by the user). In general, however, it is recommended to use the static bad pixel masks that are generated by default.

If `combine_type` is set to 'minmed', then this step will use a slightly more sophisticated algorithm to create a cleaner combined image. The basic concept in this case is that each pixel in the output combined image will be either the median or the minimum of the input pixel values, depending on whether the median is above the minimum by more than a certain number of sigma. An estimate of the 'true' counts is obtained from the median image (after rejecting the highest-valued pixel), while the minimum is actually the minimum unmasked ('good') pixel. This algorithm is designed to perform optimally in the case of combining only a few images (3 or 4), where triple-incidence cosmic rays often pose a serious problem for more simplified median combination strategies. It performs the following steps:

1. Create median image, rejecting the highest pixel and applying masks
2. Use this median to estimate the true counts, and thus derive an r.m.s.
3. If the median is above the lowest pixel value by less than the first value mentioned in `combine_nsigma`, then use the median value, otherwise use the lowest value.

If `combine_grow` > 0, repeat the above 3 steps for all pixels around those that have already been chosen as the minimum, this time using a lower significance threshold specified as the second value in `combine_nsigma`. This is very successful at flagging the lower-S/N 'halos' around bright cosmic rays that were flagged in the first pass.

3.5. Blot Back the Median to the Frame of the Original Images

Parameters:

`blot_interp`	=	'poly5|poly3|nearest|linear|sinc'	Interpolant
`blot_sinscl`	=	1.0	Scale for sinc interpolation kernel

This takes the median image and uses *blot* to apply the geometric distortion and transform it back to the reference frame of each of the original individual input images, in preparation for the subsequent step of cosmic-ray rejection. Since *blot* uses interpolation to map the pixels back to the input detector frame, it is possible to choose the type of interpolant, as well as varying the scale if the sinc interpolant is used. Generally the default interpolant of 'poly5' is sufficient, and access to the other interpolants is provided purely on the basis of flexibility.

3.6. Create Cosmic Ray Masks

Parameters:

`driz_cr_corr`	=	no	Create CR-cleaned _cor file and _crmask file?
`driz_cr_snr`	=	'3.5 3.0'	driz_cr.SNR parameter
`driz_cr_grow`	=	1	Radius to grow around flagged pixels
`driz_cte_grow`	=	0	Length to grow along CTE direction
`driz_cr_scale`	=	'1.2 0.7'	driz_cr.scale parameter

This uses the original input images, the blotted images, and the derivative of the blotted images (created using the *deriv* task) to create cosmic ray masks (using the *driz_cr* task), stored as separate files, which can later be combined with other masks. This step can also create a '_cor' image, where bad pixels are replaced with pixels from the blotted median image. These relatively clean '_cor' images can also be used to determine shifts. The cosmic ray mask can be 'grown' by a certain pixel width, which can help eliminate faint halos around cosmic rays. In addition, it can be grown specifically along the CTE direction (typically $\sim 10 - 20$ pixels) which is useful in datasets where cosmic rays have CTE trails.

3.7. Perform Final Drizzle Combination

Parameters:

`final_wht_type`	=	'EXP|ERR|IVM'	Type of weight mask
`final_outnx`	=		Output image x-size
`final_outny`	=		Output image y-size
`final_kernel`	=	'square|point|gaussian|turbo|...'	Drizzle kernel
`final_wt_scale`	=	'exptime|expsq'	Weighting factor
`final_scale`	=	INDEF	Size of output pixels
`final_pixfrac`	=	1.0	Size of 'drop'
`final_rot`	=	0.0	Rotation (anticlockwise)
`final_fillval`	=	INDEF	Value for undefined pixels
`final_bits`	=	0.	Flag values considered good

This takes the original input images, together with the final cosmic ray masks, and drizzles them all onto a single output image. The standard *drizzle* parameters of `kernel`, `scale`, `pixfrac` and `rot` can be specified for this step. By default the pixel scale of the output image is 1, but feel free to experiment with other options (e.g. when combining at least 4 sub-pixel dithered images, `scale` = 0.5 and `pixfrac` = 0.7 can yield a sharper output PSF). By default, it creates an output weight image that is simply the effective exposure time ('EXP'); it is also possible to request it to create an inverse variance weight mask which takes all the information in the '[ERR]' extension in ACS FLT files. Users can also create their own inverse variance mask, in which case 'IVM' is selected and the mask filenames are provided with the input exposures in an ASCII file that is given to the 'input' parameter.

4. Tweakshifts - Real-Time Shift Refinement

Although the relative astrometric information in the image headers is generally reliable to better than 0.1 pixel for data obtained during the same visit and with the same guide stars, there are often cases where the shifts need to be improved. These include the following situations:

- Visits that had guide star problems, which can produce drifts from one exposure to the next depending on how the telescope was tracking. Sometimes these drifts are on the order of less than a few pixels between exposures, which generally means that it may still be useful to combine them. However, if the drift rate is much larger then this will also be noticeable on the exposures themselves and they may need to be excluded from the combination.
- Science programs that require much better relative positioning between exposures than the nominal $\sim 0.05 - 0.1$ pixel accuracy achievable from the header astrometry. This can be required for very precise stellar photometric or astrometric measurements, for example. Such data can be amenable to much more accurate relative registration if they contain more than a few thousand stars, since the aggregate uncertainty in the cumulative shift measured from all the stars can be reduced from the nominal ~ 0.1 pixel centroiding accuracy for a single star down to $\sim 0.001 - 0.003$ pixels for the net shift measurements if several thousand stars are used.
- Combination or comparison of data from different visits, where different guide stars may have been used, in which case the header astrometry between the visits may differ by as much as a few arcseconds. In these cases, if there are enough objects in common between the overlapping exposures then they can be used to directly align the images. On the other hand, if the exposures are part of a mosaic pattern with minimal overlap, then it may be better to register the images separately to a high-quality external catalog of the entire field, which will then provide sufficiently accurate relative registration between them. This is the strategy employed for several of the large-scale imaging programs on *HST*.
- Exposures of moving targets within our solar system, for which the alignment may need to be improved if MultiDrizzle does not recognize the ephemeris information, or if the ephemeris information is not completely accurate, in which case the target may be at an unexpected location on the detector.

The Tweakshifts script has therefore been developed in order to provide a means to solve for these shifts, in a relatively automatic way. It can be run in several different modes, depending on the scientific requirements:

- Refine the relative shifts of a set of exposures, either from the same visit or from different visits (assuming sufficient overlap).
- Register an image to a pre-existing output reference image.
- Register an image to a pre-existing catalog.

When using Tweakshifts to align images to one another, the observer can select either cataloging or cross-correlation as the technique to use in solving for shifts. Cataloging can be carried out using either SExtractor (Bertin & Arnouts 1996) or DAOPHOT (Stetson 1987), and is generally appropriate when the images contain a large number of discrete sources, such as star clusters or sparse extragalactic fields. When the images contain large, extended emission, such as galactic nebulae, large external galaxies or even solar system targets, then it is often more appropriate to use cross-correlation, since a lot of the signal in the images may be contained in large-scale diffuse structure that may prevent effective cataloging but can provide a robust cross-correlation measurement.

Tweakshifts provides a significant amount of access to the parameters for the various techniques available to solve for shifts. All the SExtractor parameters are available, as are the relevant parameters for DAOFIND. When an input catalog is provided, Tweakshifts will use the R.A. and Dec. columns as input (as specified by parameters), and will use transformations to the distorted frame of an individual exposure in order to compute the required shifts.

The resulting shifts are written out to an ASCII text file for easy manipulation, and can also be stored in an association table if that was provided as input to Tweakshifts. Thus, when MultiDrizzle is run, it can be given the shiftfile as input (or the modified association table), and will correctly apply the shifts to each exposure in order to improve their registration.

The first public release of Tweakshifts in PyRAF STSDAS V3.4 (1 November 2005) to the community is considered a prototype, but has sufficient flexibility and robustness that it should be directly useable for a significant majority of scientific applications that require it. Future versions will include more functionality and other improvements in response to feedback from the community.

5. The Future - Virtual Observatory and Hubble Legacy Archive

Since the fundamental design of both MultiDrizzle and Tweakshifts is aimed at allowing them to be run autonomously on a large variety of datasets, this makes them amenable to incorporation into the *HST* Archive Pipelines that automatically process *HST* data. MultiDrizzle was incorporated into the *HST* on-the-fly-reprocessing (OTFR) pipeline for ACS in September 2004, and since then it has been delivering automatically cleaned, drizzled, combined images for all ACS associations that contain multiple exposures of a given target.

As a result, it is conceivable that the use of MultiDrizzle can be extended to provide products that are suitable for the Virtual Observatory and Hubble Legacy Archive (VO/HLA), which will be aimed at delivering clean, combined, geometrically rectified images for a variety of different purposes. For example, enabling scientific work to be carried out on multiple visits of the same target, either by combining them or by enabling time-variable phenomena to be searched for in different epochs, will require automatically cleaned images of each epoch, registered onto a common grid.

Since Tweakshifts is amenable to being run autonomously to determine the relative registration between datasets, this means that it could in principle be used in the VO/HLA context to refine the shifts between any set of images specified by the observer. Another use for Tweakshifts would be to register images to an existing, pre-defined catalog, for example the Guide Star Catalog II (GSC-II; McLean et al. 2004; McLean 2006), which will be used in *HST* operations from Cycle 15 onward and has much improved astrometry over the previous system. A demonstration of the potential of this technique has recently been carried out for a significant number of images from the first year of ACS operations (Koekemoer et al. 2005b, and these proceedings). Current work is aimed at investigating ways in which Tweakshifts can be made more robust, as well as incorporating new algorithms that may make it applicable to an even wider range of datasets.

Acknowledgments. We are pleased to acknowledge very valuable contributions from a large number of people who have contributed ideas or feedback, including Eddie Bergeron, Tom Brown, Ivo Busko, Stefano Casertano, Colin Cox, James Davies, Linda Dressel, Harry Ferguson, Shireen Gonzaga, Perry Greenfield, Dave Grumm, Robert Jedrzejewski, Diane Karakla, Ray Lucas, Jennifer Mack, Jesus Maiz-Appelaniz, Max Mutchler, Vera Platais-Kozhurina, Adam Riess, Marin Richardson, Ed Smith, Megan Sosey, as well as many others.

References

Bertin, E. & Arnouts, S. 1996, A&AS, 117, 393

Fruchter, A. S. & Hook, R. N. 2002, PASP, 114, 144

Gilliland, R. 2005, "Guiding Errors in 3-Gyro: Experience from WF/PC, WFPC2, STIS, NICMOS, and ACS", *Instrument Science Report* TEL 2005-02, (Baltimore: STScI), available through http://www.stsci.edu/hst/observatory/documents/isrs

Koekemoer, A. M. et al. 2002a, "The *HST* Dither Handbook V2.0" (Baltimore: STScI)

Koekemoer, A. M., Fruchter, A. S., Hook, R. N., & Hack, W. 2002b,2003, in *Proc. 2002 HST Calibration Workshop*, ed. S. Arribas, A. Koekemoer, & B. Whitmore (Baltimore: STScI), p. 337

Koekemoer, A. M., Kozhurina-Platais, V., Riess, A., Sirianni, M., Biretta, J., & Pavlovsky, C. 2005a, "Two-Gyro Pointing Stability of *HST* Measured with ACS", *Instrument Science Report* ACS 2005-07, (Baltimore: STScI), available through http://www.stsci.edu/hst/acs

Koekemoer, A. M., McLean, B., McMaster, M., & Jenkner, H. 2005b, "Demonstration of a Significant Improvement in the Astrometric Accuracy of *HST* Data", *Instrument Science Report* ACS 2005-06, (Baltimore: STScI)

McLean, B., 2006, The 2005 *HST Calibration Workshop.* Eds. A. M. Koekemoer, P. Goudfrooij, & L. L. Dressel, this volume, 413

McLean, B., Lattanzi, M., Greene, G., Loomis, C. & Meakes, M. 2004, SpaceOps 2004 conference proceedings

Stetson, P. B. 1987, PASP, 99, 191