Automated Data Analysis in Astronomy, **xx-xx**

The Reduction of CCD Mosaic Data

Francisco G. Valdes

National Optical Astronomy Observatories, P.O. Box 26732, Tucson, AZ, USA

Abstract It appears that a technological limit on the size of astronomical CCDs has been reached. In order to get more pixels in the focal plane astronomers are developing cameras using mosaics of CCDs. There are a number of such mosaic cameras in use and bigger mosaics are under development or being planned. This paper describes the data reduction techniques required for mosaic CCD data. The techniques are presented in general terms applicable to any mosaic of CCDs. There are also sections describing the IRAF tasks and algorithms, based on these techniques, that are available for the reduction of mosaic CCD data.

Key words: Mosaic CCD Imagers, Data Reduction, IRAF, **MSCRED**

1. Introduction

One of the biggest advances in optical astronomy in the last thirty years has been the development of highly sensitive, digital detectors based on *charge coupled devices (CCDs)*. CCDs have grown in size from the first small devices of a few hundred pixels on a side to the current maximum of around 4000 pixels on a side. It now appears a technological fabrication limit has been reached and single CCDs are not expected to get much larger. However, at the typical resolution required for ground-based imaging these CCDs are still small compared to what is possible with existing and planned wide-field telescopes.

A solution to this problem is to use multiple CCDs to tile the field of view. We call cameras based on this approach *CCD mosaics*. In the simplest approach of butting CCDs together there will be gaps between the CCDs, though these can be fairly small. In the 1990s several observatories began building CCD mosaic cameras. Current functioning mosaics are 8K to 12K pixels on a side. Figure 1 shows an exposure with the early engineering NOAO Mosaic Imager [1] which illustrates the wide-field, the gaps between CCDs, and instrumental artifacts such as gain variations and bad columns. There are also several *mini-mosaics* of just a couple CCDs that are within the capabilities of smaller

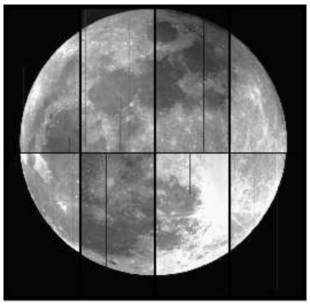

Figure 1: Raw mosaic exposure of the moon with the early engineering version of the NOAO Mosaic Imager at the KPNO Mayall 4meter telescope. The field of view is 36 arcmin square at 0.26 arcsecond/pixel. The mosaic consists of eight 2K by 4K CCDs. Note that figures 1, 2, 4, 5, 6, and 7 which show a mosaic as a single image in an image display are really a tiling of the display by the separate CCD images and do not actually form a single image. The creation of a single image is described in §8.

observatories. The next generation of mosaic cameras will have roughly 16K formats. Future concepts include a camera with more than 1000 CCDs to cover a three degree field of view. Such a camera and telescope would be able to image the entire night sky at a site in just a few days with short exposures!

The data reduction challenges with CCD mosaic cameras include handling the large volume of data and combining the output of the individual CCDs to create photometrically and astrometrically correct images. While it is possible to simply treat each CCD image in a mosaic independently, the combining of CCDs from a single exposure and then combining multiple exposures is an important aspect of mosaic cameras for surveys and for making deeper images of the sky.

This paper discusses the data reduction techniques for mosaic observations, from basic calibration of the individual CCDs to producing deep images from multiple dithered exposures. It does not discuss photometric zero point calibrations from standard stars. The methods described in this paper are applicable to any CCD mosaic, though in some steps they assume the mosaics do not overlap so that points on the sky are imaged by no more than one CCD in a single exposure. Sometimes there is more than one way to accomplish a particular calibration. In this paper we describe and recommend one way which has been found to be practical and sufficient based on experience with several mosaic camera systems.

The techniques were originally developed for the NOAO Mosaic Imagers and refined as part of the NOAO Deep Wide-Field Survey (NDWFS) [2] (a combined CCD mosaic and IR survey). For various papers on the software designs and implementations for the NOAO Mosaic Data Handling System [3], which include the general IRAF mosaic reduction tools presented in this paper, see the references in [4]. All the illustrations in this paper are based on data from either of the two NOAO Mosaic Imagers (Mosaic I at KPNO and Mosaic II at CTIO). Some of the images are is from the NDWFS.

The reduction methods described here will be used in an NOAO project to create a pipeline for mosaic camera data and to reduce and archive all NOAO Mosaic Imager data. The software tools are implemented within IRAF and details and examples of these are given in this paper. However the methods and tools are generally applicable to any CCD mosaic and have been used with a variety of instruments from different observatories around the world.

We begin this paper with an overview of a data format appropriate for CCD mosaic data and a summary of the main stages of mosaic data reduction. Each section in this paper starts with a general description of the concepts, methods, and steps which are independent of any particular software. There are then subsections which discuss specific IRAF software that implement the general concepts. These IRAF sections are intended to identify the available software tools and describe some of the more complex, interesting, and mosaic specific algorithms. They do not discuss parameters or usage details. This type of information may be found in the *Guide to the NOAO Mosaic Data Handling Software* [5].

2. Mosaic Data Format and Data Reduction Overview

An exposure from a CCD mosaic camera produces multiple images; one for each CCD amplifier read out. In this paper when we refer to a *CCD image* we mean data read out from a single CCD amplifier, though during the data reductions images from multiple amplifiers of a CCD may be merged into a single image for the entire CCD. Depending on the controller, amplifiers, and mosaic configuration, the images may be stored in a common orientation relative to the sky or have relative flips based on the way the amplifiers are read out. The display and reduction software takes care of these orientations automatically.

There are various formats that might be used to store the data. The one adopted by NOAO and many other groups is *multiextension FITS format (MEF)* [6]. All IRAF tasks can access this format [7] and the IRAF **MSCRED** package for the reduction of mosaic data is designed to operate on this format. The MSCRED User's Guide [5] includes information on how to use IRAF with this MEF mosaic format.

Each amplifier produces a FITS *image extension* with its own set of keywords. A useful convention adopted by NOAO and others, called *inheritance* [7], places many of the common keywords in a *global header* and just records keywords in the extension headers which are different for each image or which the data provider explicitly places in the individual extension headers. The software then merges the two whenever a complete set of keywords is required.

Each image in a mosaic exposure can have an associated *pixel mask*. In IRAF these are currently separate files in a special *pixel list* format. The pixel list format compactly identifies classes of pixels, such as bad pixels, associated with a data image by integer values. In the future these files will be stored in special FITS extensions, either with the observation data or as separate MEF files. The important concept is that each CCD image has an associated pixel mask in a format understood by the software. When there are multiple amplifiers from the same CCD there may be a single pixel mask for the CCD and the software can determine the region covered by the amplifier in the mask. In other words, it is possible for one mask to be associated with more than one mosaic image.

During the observations *object or science exposures* are acquired along with various *calibration exposures*. The calibrations and science programs are very similar to observations with single CCDs. One aspect where they may differ is that mosaic science exposures are often *dithered* (multiple exposures of the same field with the telescope pointing shifted) to allow filling in the gaps between the CCDs and to correct for defects as well.

The flow of mosaic data reductions has four general stages. These are basic CCD instrumental calibration, astrometric calibration, resampling to a geometrically correct and common coordinate grid on the sky (either for each mosaic piece separately or as a single reconstructed image), and combining multiple exposures. Multiple exposures are combined to produce deep images, fill in the mosaic gaps, and eliminate cosmic rays and other bad data. The reductions steps may sometimes be iterated or performed in multiple passes with incremental corrections. Depending on the science program some of these stages may not be required.

During the various stages as the data are calibrated and resampled, the pixel masks are updated to reflect the addition of new information about the pixels and the geometric changes and merging of the mosaic elements.

2.1 The IRAF Mosaic Reduction Package: MSCRED

Data reduction of mosaic camera observations in IRAF is based on both general IRAF image processing facilities and specialized tools for mosaic data. Within in the IRAF environment we make use of the command language (for parameter editing, scripting, etc.), general image manipulation tasks, and interface tools such as image display and graphics devices. One important aspect of the IRAF environment is the *FITS Image Kernel* [7] which allows any IRAF task that operates on simple images to be applied to image extensions of the mosaic MEF format.

The mosaic specific tasks and utilities are organized in an IRAF package called **MSCRED**. One difference between mosaic specific tasks and general IRAF image tasks is that mosaic exposures are specified as a unit (the MEF file) rather than requiring a particular extension or list of extensions to be given. Though the package was developed for mosaic data having multiple CCDs, conceptually one can consider a single CCD as a mosaic of one piece. Therefore some of the useful tasks in this package may be used with ordinary images. Also a single CCD read out with multiple amplifiers into multiple extensions may be considered a mosaic and reduced by the same software.

In the following subsections we describe or identify the important mosaic reduction tasks in the **MSCRED** package. In some cases only a short reference is made. In the text the names of IRAF tasks, including those in **MSCRED**, are shown in bold uppercase. There are many tasks in the **MSCRED** package that are not mentioned, some of which may be used during data reductions. These include tasks for tape access, display, data inspection, and observing tools. For details on all the **MSCRED** tasks consult the **MSCRED** guide [5] and the on-line help pages (though not all tasks currently have help pages).

Some tasks which are identified as useful in the reduction of CCD mosaic data do not have specialized versions to treat a mosaic as a unit. In those cases the tasks must be applied to each image in the MEF file. There are several ways this can be done conveniently. These include using the tasks **MSCCMD** and **MSCSPLIT/MSCJOIN**. This is described in more detail in [5].

3. CCD Instrumental Calibrations

The first stage in reducing CCD mosaic data is the removal of instrumental artifacts and photometric variations from each raw CCD image. This includes correcting for electronic bias, exposure bias, dark counts, pixel responses (including non-linearities), as well as other, less common, instrumental effects. The goal of this stage is to produce data where all the pixel values in all the CCD images have the same linear response to incoming photons. Those pixels which cannot be calibrated are identified in bad pixel masks. The techniques for determining and applying many of these calibrations in a single CCD image are standard and so we just note the differences related to mosaics of CCDs.

We divide the CCD instrumental calibration description into three sections. This section deals with basic calibrations which are applied to each science exposure independently of other science exposures. The following sections describe calibrations for scattered light, fringing, and sky flat fielding which are derived from sets of science exposures.

There are two steps that are applied to the raw amplifier values before they are modified by bias and response calibrations. One of these is identification of *saturated* pixels and the other is identification, and possibly correction, of amplifier *crosstalk signals*. Figure 2 shows an example with both a mildly saturated star and crosstalk artifacts.

In this paper we define saturated pixels as those above some threshold in the raw data values. The threshold can be set lower than the actual analog-to-digital or CCD full-well saturation. The threshold should be where the data values become uncorrectably non-linear. Saturated pixels are identified by finding the raw pixel values above the specified threshold values, which may vary for each amplifier, and adding them to the bad pixel masks.

The CCD amplifiers in mosaic cameras are typically read out in parallel. In some instruments the controller electronics have a small amount of crosstalk where pixel values for one amplifier are affected by the signals in other amplifiers. This may result in faint artifacts, either positive or negative, from bright sources in one image appearing in another image at the location where those pixels were read at the same time as the bright source. Note that the crosstalk signal may occur at all signal levels and it is only the crosstalk signals due to the bright sources which are easily visible. One way to treat this problem is to find pixels above some threshold and flag, in the bad pixel masks, all pixels in the other images that are affected.

If the crosstalk has a predictable effect one can try and correct the data. A simple crosstalk model, which has been found to correct crosstalk artifacts in the NOAO Mosaic Imagers and may be applicable to other mosaic instruments, is that some fraction of the signal from one image, called the *source*, has been added or subtracted from the signal in another image, called the *victim*. The calibration is then to determine the fraction or *crosstalk coefficient* between a source and victim, multiply the source image by this coefficient, and add or subtract it from the victim image.

Note that it is possible that a source may also be a victim and that a victim may be affected by multiple sources. In this simple model each pair of source and victim are treated independently and the source pixel values used to correct a victim are treated as unaffected by other amplifiers. For small crosstalk coefficients this is a useful approximation.

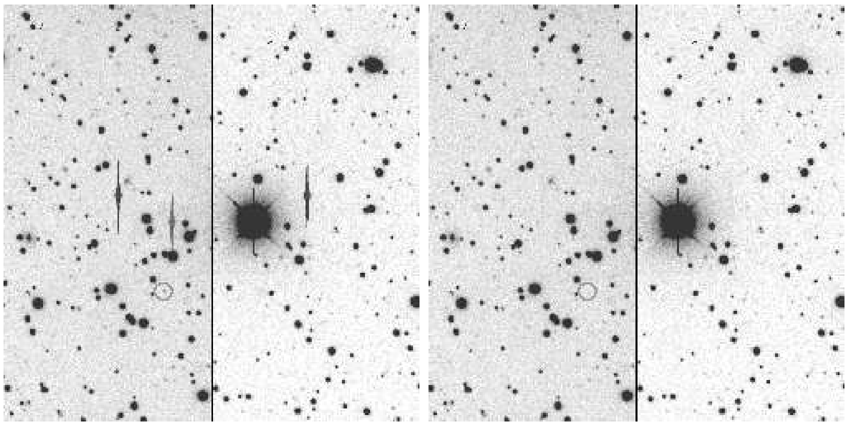

Figure 2: Raw (left) and crosstalk corrected (right) views of the same small region of a mosaic exposure with the NOAO Mosaic Imager (Mosaic II) at the CTIO Blanco 4meter telescope. In this data each of the eight mosaic CCDs is readout with two amplifiers. The region shown is between two amplifiers from the same CCD. The pixel readout order is from left-to-right for the left half and from right-to-left for the right half in each panel. In both halves the lines are read out from bottom to top. A mildly saturated star is evident by its short *bleed trail*. The *crosstalk signal* from the star is seen as a mirror image across the division between the amplifier readouts. The other strong signals are actually caused by saturated stars in amplifier images which are not shown. There are many fainter crosstalk signals in the left panel, from bright but unsaturated stars, which look just like stars. The circle shows one example identified by being present in the raw data and absent in the crosstalk corrected data.

The crosstalk coefficients are determined by taking the ratio of background subtracted pixel values in a victim amplifier to raw pixel values in a source amplifier. For regions where the source amplifier has bright objects and the victim has nothing but sky plus the crosstalk contribution, the ratios will average to the crosstalk coefficient between those two amplifiers. The sign of the coefficient determines whether the correction needs to be added or subtracted. Figure 3 illustrates this technique using the the program described in §3.2.

The identified saturated and uncorrectable crosstalk affected pixels are added to a static bad pixel masks to create exposure specific bad pixel masks. These static masks have bad pixels which have been identified as not useful or correctable in all exposures from the mosaic CCDs. These are generally due to defects in the CCD pixels which can be mapped in advance in the laboratory or during commissioning of the camera.

The static masks are generally derived from flat field exposures taken at different exposure times and processed as described in this section but without applying any correction for bad pixels. At a given exposure time, a sequence of exposures are averaged to reduce the noise. The idea is to find the pixels that do not respond linearly with exposure time. The simplest method is to take the ratio of two images at different exposure times and find pixels which deviate by some statistically significant amount from the average. This technique for the creation of bad pixel masks is the same as for single CCDs except for needing to make masks for all the CCDs in the mosaic.

In addition to adding new bad pixels to the static masks, the raw images may have bad pixels replaced by interpolation from nearby good pixels. This is done both for cosmetic reasons as well as to ensure that bad data values do not cause problems in programs which do not make use of the pixel masks for reasons of efficiency or design. It is important to realize that replacing the bad pixels in the data is a benign step since the pixel masks will continue to be associated with the data throughout the reduction process. Therefore, any scientific measurements can determine which pixels were not actually observed. This pixel interpolation step is a standard operation with single CCDs and there is no special treatment in mosaics other than the bookkeeping of matching masks with each CCD image.

For a CCD mosaic the calibration steps are repeated for each CCD image. Calibration images are now *calibration mosaics* in the same MEF format. The images in the calibration mosaic are matched by CCD and amplifier with the images in the mosaic exposure being calibrated. Calibration mosaics are obtained in the same fashion as with single CCDs by taking one or more calibration exposures. When a sequence of calibration exposures is taken they are averaged together, possibly with some statistical rejection for cosmic rays. Averaging of mosaic exposures means averaging all images from the same CCD amplifier to produce an average image for that CCD amplifier.

The standard CCD calibration steps are as follows. Electronic bias is subtracted using overscan/prescan data in each image. The overscan/prescan region(s) are removed after the electronic bias is calibrated. Exposure bias patterns are subtracted using a zero calibration mosaic. The zero calibration mosaic is an average of a sequence of zero exposure time readouts taken with the shutter closed. Dark count patterns are subtracted by scaling a dark count calibration mosaic to the exposure time of the mosaic observation being calibrated. The dark count calibration mosaic is an exposure (or average of exposures) taken with the shutter closed and of comparable exposure time to the primary science exposures.

Basic flat fielding, using an average of a sequence of dome screen or internal lamp exposures, is also standard except in one regard. Rather than normalizing each flat field image by its own mean, the mean of all the images in the flat field calibration mosaic is used. This ensures that the relative gains between the images is also removed to the extent the camera is exposed to a uniform illumination. Note that if there is a scattered light pattern in both the dome/internal flat field and the sky exposures then the pattern must be removed from the flat field calibration mosaic prior to applying it to the data. This is described in §4.

As with single CCDs, it is possible to readout each CCD in a mosaic using more than one amplifier. This is done to speed up the readout time. When there are multiple amplifiers per CCD the images may be *merged* into a single image for the CCD after the overscan/prescan data is applied and discarded. This step is simple since there are no registration or combining issues. The final MEF format will have fewer extensions than the raw data at this point.

This merging step is optional but it is recommended, both to make it easier to analyze the data as a set of CCD images and to avoid boundary effects between the amplifiers if the mosaic images are resampled to remove geometric distortions and produce a single image.

3.1 Identifying Bad CCD Pixels: CCDMASK

A method for identifying defective or non-linear CCD pixels using ratios of flat fields was described in §3. The flat field mosaics are processed and multiple amplifiers from the same CCD are merged. A ratio mosaic is produced using **MSCARITH**. The IRAF task **CCDMASK** may be used to identify the non-linear CCD pixels in each image. While this task tries to identify weak coherent column features which are obvious to the eye it is still not perfect. Other image processing tools, such as **IMEXPR**, **IMREPLACE**, and **IMEDIT**, may be applied. The tasks in the **CRUTIL** package may also prove useful. The development of additional IRAF tools for creating and manipulating bad pixels masks is currently an area of active work. The end result is the production of static bad pixel masks for each CCD.

Note that if multiple amplifier readouts are merged there will be one bad pixel mask associated with multiple raw amplifier images. The matching of the part of the CCD bad pixel mask corresponding to the amplifier readout is handled automatically by the IRAF software.

3.2 Crosstalk Corrections: XTCOEFF and XTALKCOR

XTALKCOR reads a file containing coefficients α_{vs} for a simple linear crosstalk model between amplifiers in a mosaic exposure and creates a corrected copy of the input exposure. In addition or alternatively, affected pixels are added to a bad pixel mask. The crosstalk model, proposed by James Rhoads and implemented in these tasks, takes the form

$$I'_v = I_v - \sum_s \alpha_{vs} I_s \tag{1}$$

where the arithmetic is done on each pixel in the victim image I_v using the matching pixels in the source image I_s. Note that in this model a source image may also be a victim image but there is no attempt to decouple the interactions. Bad pixel masks may be generated by flagging pixels which have corrections greater than a specified threshold, $|I'_v - I_v| > T_{cor}$, or which have source pixels greater than a threshold, $I_s > T_s$. Images which are not victims of any source image are simply copied to the output unchanged and nothing is added to the bad pixel mask.

The crosstalk is assumed to occur during the simultaneous readout of multiple amplifiers; from the same CCD and/or different CCDs. Thus the victim and source pixels must be matched in the order in which the pixels are read by the amplifiers. The task determines this based on keywords defining an *amplifier coordinates system*. This requires possibly flipping the image lines and/or columns.

When victims are affected by many sources and sources are also victims there will be many combinations. In order to apply equation 1 efficiently, the task uses an image line buffering algorithm that reads lines from all extensions which were read out at the same time, computes all the corrections based on those lines, and outputs all the corrected lines. In this way the input and output lines are read and written only once. The task also groups the images that affect each other and, when groups are disjoint, does each group sequentially. This minimizes the amount of memory required for the buffering.

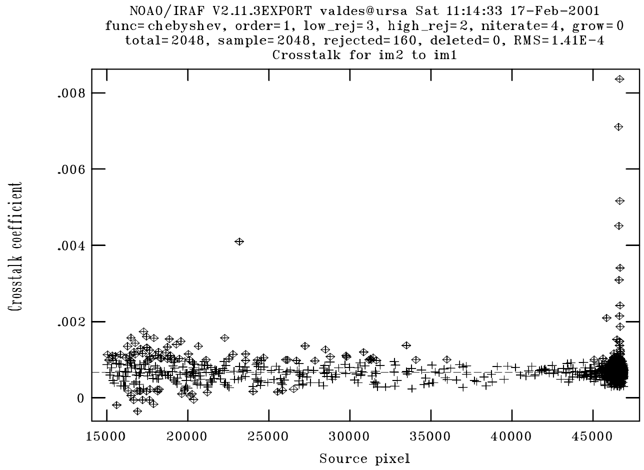

Figure 3: Graph from **XTCOEFF** showing the crosstalk coefficient estimates (see equation 2) from pixels above 15000 ADU for the data shown in figure 2. The source amplifier is the one on the right half and the victim is one on the left half. The crosstalk estimate is the fitted line. The diamond points were rejected automatically by iterative sigma clipping. The output of the task gave a crosstalk coefficient between the two amplifiers of 0.000654 ± 0.000014. This coefficient is what was used to remove the crosstalk signals in the right panel of figure 2.

While **XTALKCOR** can be executed separately, it is normally executed as part of all the corrections applied by **CCDPROC** (§3.3). Currently **CCDPROC** does not provide the bad pixel flagging option. The task uses keywords to insure the corrections are done only once and to document the crosstalk coefficients applied. Figure 2 shows an example of an exposure with crosstalk which is corrected by **XTALKCOR / CCDPROC**.

The crosstalk coefficients may be supplied by the observatory or derived by the user. In either case the coefficients are obtained by using the task **XTCOEFF** on a suitable calibration exposure(s). This is one that has many bright stars with data values sufficiently high to cause statistically significant crosstalk signals in each image. However the number of stars should not be so large that the probability of a crosstalk signal falling on top of the direct image of an object is high. Ideally the exposure should also have a low sky background.

XTCOEFF considers each pair of victim and source images. For each readout line having source pixels within some range the corresponding line in the victim is read and a background, B, is subtracted based on median of the line. An estimate for the crosstalk coefficient based on equation 1 is computed by

$$\alpha_{vs} = (I_v - B)/I_s \qquad (2)$$

where I_s is a source pixel in the specified range, I_v is the matching victim pixel, and B is the background estimate for the line. Values above some limit that would not be expected for the crosstalk are excluded as being affected by an object in the victim image at that point.

The set of coefficients from individual pairs of pixels are combined into a single coefficient estimate by fitting a constant function to the coefficients verses the source pixel value I_s. This is equivalent to computing the average. However, a fitting algorithm is used to allow examining the data graphically to check for trends away from the assumed linear crosstalk relation. The fitting approach also allows using a standard IRAF routine for examining the data interactively and applying region selections, point deletions, and sigma clipping. Error estimates for the coefficient are produced by the fitting routine.

The output of **XTCOEFF** is the file of coefficients and uncertainties that are used by **XTALKCOR**. Figure 3 shows an example graph and coefficient determination using the data from figure 2.

3.3 Applying CCD Calibrations: CCDPROC

CCDPROC is one of the primary tools for CCD data reductions in IRAF. It integrates the basic operations for calibrating CCD data in one efficient program. Most of the operations may be done at the same time in one pass through the data. This program was first developed around 1986 for single CCD data. The version in the **MSCRED** package was adapted from this early version to operate on mosaic data in MEF format. In addition to the operations and features provided in the single CCD version, the mosaic version adds crosstalk correction, detection and output of saturated pixels in a mask, application of a sky flat field as an incremental correct, and merging of multiple amplifiers from the same CCD into a single image.

3.4 Combining Calibration Exposures: COMBINE

COMBINE is another of the primary CCD reduction tools in IRAF. It is closely related to the general **IMCOMBINE** task. The primary difference is its use of the CCD reduction instrument file for keyword translations and recognition of CCD calibration types, subsets or filters, and amplifiers. The version in the **MSCRED** package was adapted from the earlier version in the **CCDRED** package for single CCD data. It operates on multiextension FITS files and distinguishes between extensions by amplifier. This task also provides output of various auxiliary data including exposure maps.

The key features of this task are scaling of data to a common photometric system, registration of offset images, and various algorithms to identify and exclude deviant data relative to the mean or median of the images being combined.

This task is used in a variety of applications. For basic CCD calibrations, as described in §3, it is used to combine sequences of calibration exposures into master calibration mosaic files. This is integrated with CCD calibrations of the individual exposures prior to combining in the tasks **ZEROCOMBINE**, **DARKCOMBINE**, and **FLATCOMBINE**. It is these IRAF commands that are commonly used to build the master mosaic calibration files. In addition **COMBINE** is used internally to **CCDPROC** to perform the merging of amplifiers into single images. Other places where **COMBINE** is used for mosaic reductions, as described in later sections, are in the tasks **SFLATCOMBINE**, **MSCIMAGE**, and **MSCSTACK**.

4. Scattered Light and Fringe Corrections

This section discusses the removal of scattered light patterns. Deriving the optimal calibration mosaic files for doing this requires combining object exposures, calibrated as described in §3, to create dark sky flat fields as explained in §5.

Scattered light having a spatially fixed pattern which varies only by a constant scaling factor is another type of instrumental artifact that must be removed. This requires creating a template of the pattern with no background, scaling it to match the pattern in each science exposure, and subtracting. Typically the amplitude of the scattered light varies depending on the amount of light in the field of view, which is why the pattern has to be scaled to each science exposure.

The determination of the amplitude scaling may be done automatically or interactively. When done interactively an exposure, or a portion of the exposure, is displayed with a trial scaling and correction. The uncorrected exposure may also be displayed for comparison. The user then sees if the pattern is over or under corrected and adjusts the scale factor followed by redisplaying the result. When the scattered light pattern is determined to be optimally removed based on a visual inspection, the final correction is performed on the complete exposure.

In this discussion we consider two categories of scattered light patterns; those which affect dome flat fields and those which do not. When the scattered light pattern affects the dome flat field, an iterative and approximate technique is required to disentangle the pixel responses from the scattered light pattern. The steps consist of first correcting the flat field so that it does not affect the photometry and then subtracting the scattered light from each flat fielded science exposure. The flat field has to be corrected first since otherwise the pattern will be suppressed from the science exposures through reducing the response of those pixels. This is incorrect because the pattern is due to extra light and not a higher response.

The first step is create a template for the scattered light pattern. In some cases this may require modeling and fitting. The pattern is generally determined from the flat field with the same filter as the data, though it could be done with some other exposure and filter if the pattern is relatively color independent. What makes this difficult and approximate is that one needs to determine the scattered light pattern in the presence of flat field variations. Ideally the source of the scattered light pattern for making a template should have the greatest contrast between the scattered light and the background.

To remove the scattered light from the flat field involves scaling the template, adding a background of one, and dividing the flat field by the pattern. The scaling may need to be done iteratively and visually until the pattern is no longer visible in the flat field. The pattern must be divided from the flat field rather than subtracted. This is because the scattered light is modulated by the actual flat field responses and so you need to *flat field the flat.*

After the flat field is corrected, the flat fielded science exposures will show the scattered light pattern but with most of the CCD pixel response pattern removed. Now the scattered light pattern must be subtracted since it is due to extra light. Again a template of the pattern must be created. The best way to do this is to make a first version of a sky flat field, as described in §5, from the set of science exposures and subtract the background. The template pattern is then scaled, possibly interactively, and subtracted from each science exposure.

So far we have talked in generalities about a scattered light pattern. A specific example occurs in data from the NOAO Mosaic Imager (Mosaic I) on the Mayall 4meter telescope. Despite anti-reflection coatings in the corrector there is a faint *pupil image* caused by reflections. This appears as a large ring pattern in the center of the field of view. A modeling program for this pattern is described in §4.1 as well as a tool for scaling the model to either a flat field or an object and removing it by division or subtraction. Figures 4 and 5 show the pupil image pattern in a dome flat field and in object data. They also show the results of removing this pattern as described above.

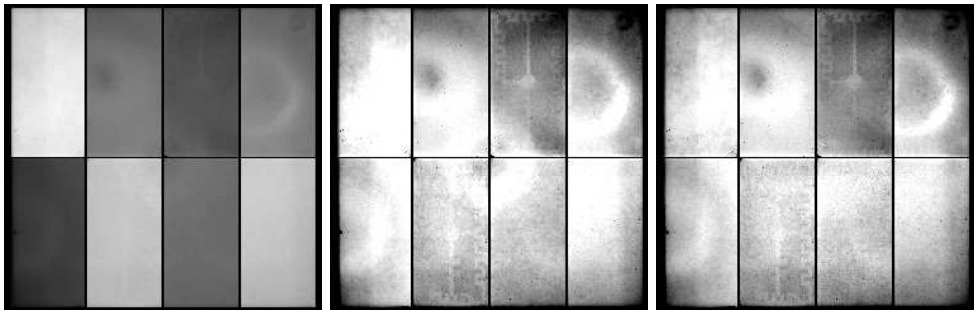

Figure 4: The three panels show the same NOAO (Mosaic I) processed mosaic flat field. The left panel is displayed with the same grayscale for all the CCDs to illustrate the variations in the mean gain between CCDs. The other panels are displayed with different grayscales for each CCD in order to show the flat field structure within the CCDs. Though the middle and left panels are displayed with independent scaling, when the flat field is applied to object data by **CCDPROC** a common mean value is used to normalize the flat field so that the relative gains seen in the left panel are removed. There are several ring-like, large scale flat field structures visible in the middle panel. The ring in the center, however, is not a flat field effect but a pupil image. The panel on the right shows the result of the pupil modeling and removal using **MSCPUPIL** and **RMPUPIL**. Note how only a quarter of the ring appears in each of the four central CCDs and the modeling must fit just these quarter rings.

Another scattered light pattern is fringing in the CCDs. It is a scattered light pattern in the sense that it is extra light in some places and a deficit of light in other places which is removed by a subtraction of a pattern. This type of pattern does not appear in the dome flat fields because caused by interference from the night sky lines. Therefore, the step of removing it from the flat field is not required. The pattern is extracted from the data by again creating a version of the sky flat field, now with any other pattern, such as the pupil image, removed. The scattered light and fringe patterns are removed in separate steps because they may scale differently. For instance a pupil image depends on the total light while the fringes depend on the night sky lines.

The fringe pattern is a modulation about some mean value. To obtain the fringe pattern template from the data a large scale background is determined and subtracted. This may be determined in various ways such as fitting a low order surface function or using a box average or median filter. The fringe template is scaled and subtracted from the science exposures in essentially the same way as described earlier. Figure 6 shows

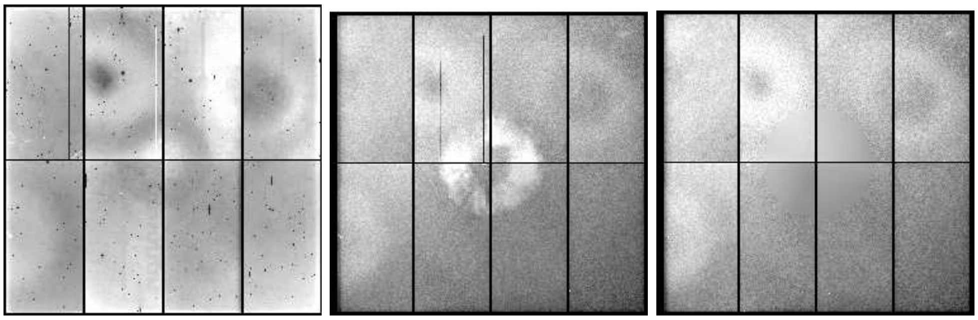

Figure 5: The left panel shows a raw NOAO (Mosaic I) standard star field exposure displayed with independent grayscales for each CCD. This shows the flat field structures and the pupil image. The middle panel shows a sky flat field produced from many science exposures that were processed with a pupil image removed dome flat field (see figure 4). The pupil image is more prominent because most, though not all, of the flat field structure has been removed and the display stretching (the same for all CCDs to show gain variations between CCDs) is larger. Note that despite using a dome flat field there are still signs of the flat field structures and gain variations due to color and illuminations differences between the dome flat field and the sky exposures. The right panel is a sky flat field made after modeling the pupil image in the middle panel and removing the model from all the science exposures. Though the region where the pupil image has been removed is visible, keep in mind that the pupil image in the middle panel is only a few percent in amplitude and so we are dealing with small corrections. Also the region is visible mostly due to a change in the noise pattern due to removal of most of the fringing in that region during the pupil image removal.

an example of fringing, a fringe template, and the result of correcting a science exposure for fringing as well as a final sky flat field.

4.1 Removing a Pupil Pattern: MSCPUPIL and RMPUPIL

The task **MSCPUPIL** is used to model a pupil pattern such as is seen in data from the NOAO Mayall telescope. The location of the pupil pattern is defined relative to the world coordinate system tangent point (§7) which is common to all the CCDs and is generally near the center of the mosaic field and the center of the pupil image. A radius for the pattern is also specified. Outside this radius the pattern is taken to be absent or negligible. The position and size of the pattern are specified for a full read out and the software corrects for binning and subregion readouts. The pupil pattern may be distributed across many images and the software determines what part of the pattern lies in each image. The program also ignores data identified in a bad pixel mask and pixel values outside specified limits.

The object of the process and program is to separate the pupil image from the flat field pattern as much as possible. This is hard to do precisely but one type of flat field feature which sometimes occurs in CCD responses, particularly near the edges, are line-by-line patterns. So an option in the program allows dividing the data by a median of each line prior to fitting the pupil image.

The pupil pattern is modeled as a ring which goes to zero where it merges into the background. Because the pattern is a ring it is modeled by fitting functions in polar coordinates (r, θ); in other words functions are fit radially and azimuthally. Similarly,

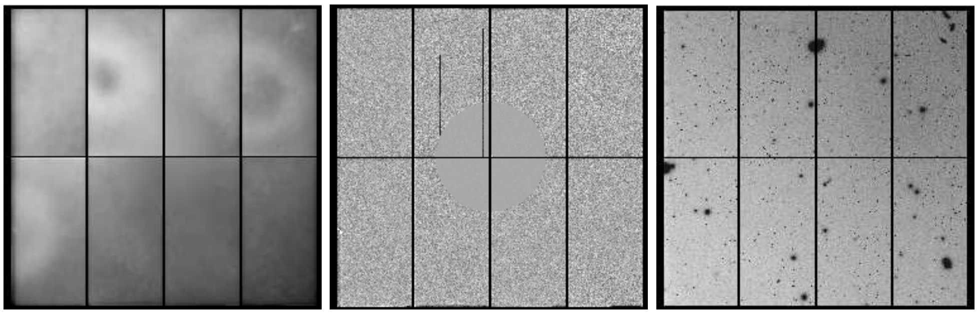

Figure 6: The left panel is a median filtered version of the sky flat field shown in the right panel of figure 5. The middle panel is the residual. All the structure outside of the central ring is caused by fringing (pixel noise is not significant at this resolution and grayscale stretch). Note how the region where the pupil image was removed has almost no fringing. This is because the process of subtracting the pupil image also removes the fringing when the scaling needed to subtract the pupil template is comparable to the scaling needed to subtract the fringing. There is no requirement that these scalings be the same but in practice they are similar. The fringe frame is scaled and subtracted from all the object exposures and a fringe corrected sky flat field is created. It is not shown in this figure but it looks similar to the left panel. This final sky flat field is applied to the science exposures to produce a final calibrated exposure as shown in the right panel. Note that the exposure is not flat across the field due to residual sky gradients. Compare the left panel of figure 5 with the right panel of figure 6 to see the transformation from a single raw exposure to a fully processed single exposure, though the two displays are not of the same field.

the background under the pattern is modeled by fitting circular regions of specified widths just inside and outside the ring. This is done independently for each image that contains part of the ring.

The data within the ring, including the inner and outer background rings, are extracted from the image being modeled. The standard deviation of the data is computed and data outside some specified range of standard deviations are excluded. The same clipping limits are also used in the function fits if a number of sigma clipping iterations is specified. The combination of excluding data using a mask, data thresholds, sigma clipping the data in the ring, and sigma clipping during function fitting are important when deriving pupil models from science exposures containing galaxies and stars, bad pixels, and cosmic rays.

The data in the two background regions are fit as a function of $theta$. The background under the pattern is obtained by interpolating in r between the inner and outer functions evaluated at the θ of each pixel in the ring.

A mean radial profile for the ring pattern is fit to all the background subtracted data as a function of r. The ring data are normalized by this mean radial profile. Where the profile is close to zero the data are ignored. Azimuthal structure in the ring is modeled by fitting residual radial profile functions in azimuthal bins. Because the large scale radial profile has been removed with a function of some higher order, the residual radial profiles are fit by lower order functions. Though the underlying fitting engine allows specifying this order, **MSCPUPIL** fixes this order to be a constant. Thus, the pupil ring model consists of a ring with a constant radial profile whose amplitude is modulated in azimuth. The azimuthal bins are allowed to overlap by specifying a bin step which is smaller than the bin size. This results in a smooth variation in the azimuthal modulation while allowing sufficient data in each azimuth bin for a measurement.

The last step is to output the pupil pattern. **MSCPUPIL** allows four types of output as well as printing statistics on the amplitude and flux of the pupil relative to the background. One output is the ratio of the data to the fit. This is what is commonly used to correct a dome flat by a pupil model derived from itself. Less common is to derive a pupil model from a flat field exposure in a different filter that has a higher pupil image amplitude. In that case the fit is output and **RMPUPIL**, as described below, is used to remove the pattern from the flat field for the filter being corrected.

Another type of output is simply the background subtracted data. When this output is selected the ring fitting described earlier is not performed. In effect this *skims off* the pupil image and so makes no assumptions about the details of the pupil image structure. This type of output is what is used when extracting the pupil image from a sky flat field for subtraction from the science exposures.

The last output type is the difference of the data to the fit. This would be used to subtract the pupil from exposures based just on the data itself. While this might be necessary in some cases and often works well, it can give poor results when large bright objects overlap the pupil image. Possibly in conjunction with an object detection mask this might be a viable approach. Using a pupil image template *skimmed off* a sky flat field is the recommended method for correcting individual object exposures and the difference output of **MSCPUPIL** would be used only when a good sky flat field template cannot be produced.

The task **RMPUPIL** is used to scale and remove a pupil template from other mosaic exposures. The removal may be either by division, as required to correct dome flat fields, or by subtraction. This task can make an automated estimate for the scale factor and apply the correction non-interactively. However, it typically does not do as good a job as having the user adjust the scale factor interactively using an iterative display mode.

In the interactive mode the exposure to be corrected is displayed in two frames. In one frame is the original data and in the other is the scaled and pupil removed version. An explicit default initial scale factor can be specified or the task can supply a first guess. The user then examines the two displays and enters a new trial scale factor or quits with or without applying the correction. To speed up the interactive correction and display there are options to display only a subset of the images and to use block averaged data. Since the pupil pattern may only appear in a subset of images only those images are usually displayed. When the final correction is applied, it is applied to all images in the mosaic at full resolution.

Figure 4 illustrates removal of the Mayall telescope pupil image from a dome flat field using **MSCPUPIL** with the ratio output. Figure 5 shows removal of the pupil from dome flat field corrected science exposures using a sky flat field as the template. The pupil pattern is obtained using the background subtracted data output of **MSCPUPIL** and then **RMPUPIL** to scale remove the pattern for the science exposures.

4.2 Removing the Fringe Pattern: MSCFMEDIAN and RMFRINGE

The task **MSCMEDIAN** is used to subtract a box median filtered version of a mosaic sky flat field from itself to produce a mosaic fringe pattern template. The task allows excluding data values outside a specified range. There is an option to use a fast median algorithm based on quantizing and histograming the data [8].

The task **RMFRINGE** is used to scale and subtract the fringe template from the mosaic exposures. It is a variant of **RMPUPIL** and operates as described for that task. **RMFRINGE** currently does not make an automated estimate for the scale factor. The user must specify the scale factor and run the task non-interactively or use the interactive display mode.

Figure 6 shows the result of applying **MSCMEDIAN** to produce a fringe template (middle panel) from a sky flat field. The right panel shows removal of the fringe pattern from a science exposure.

5. Sky Flat Fielding

Flat fielding of CCD data is often the limiting factor in precision photometry. The same technique of applying *sky flat fields* derived from twilight or disregistered science exposures which is used with single CCDs is also used with mosaics. However, mosaics have the extra requirement that all of the CCDs must be brought to the same gain to very stringent levels if they are to be combined into single images. Even small differences can be seen as discontinuities across the boundaries of the mosaic pieces. Complicating this is that the color response of the CCDs may not be the same and, for wide-field mosaics, the sky may not be uniform across the detectors.

The first pass at flat fielding using dome or internal flat fields (§3) is not generally adequate because the illumination pattern is different and the color of the lamp source is not the same as the sky. However, it is still useful to apply these calibration flat fields as a first calibration to allow better separation of scattered light patterns from the pixel response patterns. Therefore, a sky flat field is used as a differential calibration.

Taking exposures of the twilight sky, where objects are lost in the brightness of the sky, is one solution typically used with single CCDs. This might be adequate but experience shows that the color of the sky and brightness gradients across the generally larger field of view of a mosaic do not provide a completely satisfactory sky flat fields.

What does work well are dark sky flat fields produced by combining many exposures that do not have overlapping objects. One effect seen in using sky flat fields is that science exposures taken with the moon up should be corrected with sky flats created from data taken with the moon up and vice-versa for data taken with the moon down. This is a consequence of the color of the flat field source ideally matching the color of the sky.

Data to be used for making dark sky flats may occur naturally with the science program if the exposures are dithered (shifted a small amount comparable to the scales of the objects) or are of multiple independent fields. If the main program is of large galaxies then twilight sky flats or separate blank sky exposures need to be taken. The goal in making dark sky flat fields is to have a majority of the exposures contain blank sky at every point in the mosaic.

One step which is difficult to automate is selecting the exposures to be used. The first cut is to use all science exposures separated by whether the moon is up or down. However, during observing various conditions occur which may require throwing out bad exposures, exposures with stray light or ghosts from bright stars, or significant sky gradients across the field that are not common to all the exposures. Ultimately there is some amount of judgement required in selecting the set of science exposures to be used for a making a dark sky flat field.

There are various ways to reject the objects in the field when combining the images. One common method is to detect the objects at the same time the exposures are combined. This consists of scaling the exposures to a common level, computing a mean or median and a standard deviation at each pixel, and rejecting outliers. The standard deviation may be determined either from the known statistics of the CCD (using the readout noise and the mean photon counts for Poisson statistics) or empirically from the data. The faint wings of objects will often survive the clipping so a method to reject pixels near other clipped pixels is also required.

An alternative is to generate masks of the objects using an object detection algorithm. **ACE** [9] is an example of such software. The combining of the images may then exclude any pixels in these object detection masks.

Since the sky flat field is created from the dome flat fielded data, the sky flat field is normalized (using a single mean over all the CCDs in the mosaic as is done with the dome flat) and divided into the dome flat fielded science exposures. This is again done using matching CCD images.

As described in §4, several versions of the dark sky flat field may be created. If the sky flat field contains a pupil image pattern, the pattern is modeled and extracted from a version of sky flat field containing this pattern. The extracted pattern is scaled and subtracted from each science exposure before creating a new version of the sky flat field. If the sky flat field contains a fringe pattern, the mean background is subtracted from the sky flat to produce a fringe pattern template. The fringe pattern is scaled and subtracted from each object exposure followed by creating another new version of the sky flat. When the sky flat and all science exposures, including those not used in creating the sky flat, no longer contain additive light patterns, the sky flat field correction is applied to the science exposures. The middle and left panels of figure 5 show two versions of a sky flat field.

By design, flat fielding causes every pixel in an exposure to have the same counts for a given surface brightness. This means that a blank sky will produce the same pixel values regardless of how big an area on the sky is imaged by the pixel. When the pixels have different projected areas on the sky, due primarily to optical distortions, this leads to relative photometric errors. This flat fielding error can occur even with a single CCD but is more likely to be significant in a mosaic with a wide-field telescope.

When the photometric science measurements are made after the data are resampled to pixels of uniform size (§8) there is no need to make a special calibration for the different projected pixel sizes. However, if one does not resample the data prior to doing the photometry then a photometric calibration is required to correct for this effect. This is done using the *world coordinate system (WCS)* (§7). This coordinate system is used to compute the projected area on the sky of each pixel in a straightforward way by determining the celestial coordinate of each corner and then calculating the area. The areas are normalized to some value, usually the area of the pixel at the reference point. The relative areas are divided into the pixel values to give the correct number of photons falling in that pixel relative to all other pixels.

As mention before, this correction is not required if the data will be resampled to a output image with equal area pixels. However, if one both applies this correction and then resamples the data, the resampling operation must adjust the fluxes of the output pixels by the relative area change (called *flux conservation*) rather than simply interpolating the pixel values.

5.1 Making Sky Flat Fields: SFLATCOMBINE

SFLATCOMBINE is a specialized version of **COMBINE** (§3.4) that creates sky flat fields from multiple exposures taken at different points on the sky. It scales exposures to a common sky level and then excludes non-sky pixels before averaging or medianing the sky pixels. The non-sky pixels may be given in a mask of objects detected separately or by statistical clipping of deviant pixels. The deviations are primarily due to objects, though cosmic rays or bad pixels which are not in the pixel masks will also be excluded. The greatest problem in making sky flats is that any method which is purely based on statistical fluctuations at some level will not exclude the faint wings of objects below that level. Therefore techniques that allow rejection of neighbors of rejected pixels, sometimes called *growing*, tend to be needed. **SFLATCOMBINE** includes a grow radius option where pixels within this distance of a rejected pixel are also rejected.

The sky flat fields produced by **SFLATCOMBINE** include setting the keywords required by **CCDPROC** (§3.3) which is used to apply the correction to the science exposures as a secondary flat field calibration.

5.2 MSCPIXAREA

The task **MSCPIXAREA** is used either to make a calibration mosaic of the relative (or even absolute) pixel areas or to apply the relative area factors to calibrate mosaics data. This areas are computed using the world coordinate system in a mosaic file. Because the area correction should not vary from exposure to exposure, the area output can be used to correct a number of exposures. The relative pixel area calibration is applied by division to correct a flat fielded exposure. In addition it can be applied by multiplication to modify a flat field calibration so that when it is applied to non-flat fielded data both the flat field and the pixel area calibrations are done at the same time.

Computing the celestial coordinates of every corner of every pixel is a time consuming operation. The world coordinate system, and the underlying optical distortions that it describes, is a continuous function that generally varies slowly across many pixels. Therefore, the calculations in **MSCPIXAREA** are greatly speeded up by using a subgrid of pixels where the relative areas are computed and then interpolated, using fast linear interpolation, to every pixel.

6. Masking Cosmic Rays and Other Artifacts

Bad pixel masks are an important part of CCD mosaic reductions. In addition to the obvious importance of keeping track of data which is not scientifically valid, there is a less obvious aspect of tracking the effects of the bad pixels during resampling and reconstruction of geometrically corrected and stacked images.

In §3 the identification of non-linear CCD pixels, saturated pixels, and crosstalk artifacts was discussed. In this section we discuss the identification of CCD bleed trails and cosmic rays. In §8 the propagation of the bad pixel masks and masking of regions with no data during image resampling and reconstruction is discussed.

CCD bleed trails may be identified by looking for patterns in the vicinity of saturated pixels or bright stars. The simplest technique is to ignore the idea of a bleed trail pattern and simply extend any region with saturated pixels. This will account reasonably well for the edges of bleed trails that fall below the threshold used to define saturation. The next level of sophistication is to simulate the effects of saturated pixels that cause bleeding by shifting the locations of saturated pixels along the CCD columns. Even more sophisticated methods may be employed though the author has not investigated these.

The hardest features to identify are cosmic rays. When possible the cosmic rays should be identified before geometric resampling of the data. This is because the resampling unavoidably broadens cosmic rays, making them more difficult to discriminate from real astronomical objects. There are a variety of methods which may used to identify cosmic rays in single images or in multiple exposures. Whatever method is used the pixels corresponding to the identified cosmic rays need to be added to the bad pixel masks.

Bad pixels may be added to the bad pixel mask at various stages during the data reductions. However it is important to identify them all prior to resampling (§8), or to iterate back after finding additional bad pixels in the resampled data. This is because the bad pixels need to be replaced by smooth data values to minimize ringing artifacts during resampling. The bad pixel masks for the resampled data will identify regions where ringing artifacts occur provided the sources of the ringing are in the bad pixel masks. This is discussed further in §8.

Detection of cosmic rays from multiple exposures may be done prior to resampling only if the exposures have been made without moving the telescope. These are sometimes called *cosmic ray split* exposures. It might be possible to detect cosmic rays in shifted multiple exposures prior to resampling but this has not been studied.

When the exposures have been dithered in order to fill in the gaps and bad pixels, the detection of cosmic rays from the multiple exposures may be done after the images are resampled and registered. Though the cosmic rays will have been broadened by the resampling, making them more difficult to detect in individual exposures, the comparison with a clipped, deep stacked image or other single registered exposure is very powerful. The difficulty in this technique is dealing with mismatched point-spread-functions (PSFs) that cause automated algorithms to find the cores of the stars as potential cosmic ray candidates. See the discussion of difference detection with **ACE** [9] for one way to detect cosmic rays and deal with the PSF variations.

Cosmic rays identified in the resampled data should to be added to the unresampled bad pixels masks in order to replace them in the unresampled data prior to a new version of the resampled data being generated. The mapping of pixels from the resampled to unresampled data has also not yet been studied nor software developed. However, there is no reason this cannot be done using the WCS information.

6.1 IRAF Object/Cosmic Ray Masking Tools

For various reasons, including eliminating or minizing artifacts from saturated pixels, cosmic rays and other transient objects, it is necessary to identify pixels associated with these types of non-astronomical or transient sources in the bad pixel masks. This is an area that is open to various techniques of which only a few are noted here.

A general tool for identifying CCD bleed trails still needs to be developed. However, **MSCBLEED** is a simple prototype based on an algorithm developed by the NDWFS. This task requires the user to set thresholds for each CCD that define the core of the bleed trail. The pattern is expanded by shifting along the columns. Finally pixels within some some distance of the pattern are added.

The **CRUTIL** package has tasks for identifying cosmic rays, either in single images or in multiple images, and either interactively or non-interactively. Most of the tasks have been designed to produced or update bad pixel masks which is what is needed for the mosaic reductions described here. It includes the useful tool **CRGROW** for extending regions around pixels in a bad pixel mask.

For identifying cosmic rays in single exposures the task **CRAVERAGE** is recommended. It identifies cosmic rays as sharp features rising above a background. The most common problem with identifying cosmic rays as sharp features in single images is not identifying the cores of unresolved sources (that is stars) as cosmic rays. The **CRAVERAGE** task includes a crude measure of whether a sharp feature is associated with an astronomical object and avoids identifying cosmic rays within such objects.

More sophisticated methods for identifying cosmic rays based on the IRAF object detection and classification task **ACE** [9] will be developed in the future.

As discussed in §6, §8, and §9, it is recommended that pixels identified in the bad pixel mask be replaced by smooth data in order to avoid ringing artifacts during resampling. The pixels in the input data identified in the bad pixel masks are replaced by interpolation from nearby good pixels using the tasks **CCDPROC** or **FIXPIX**. This may be done in multiple passes where the pixels in the initial static bad pixel mask are replaced during the first pass of basic CCD calibration with **CCDPROC** and then saturated pixels, bleed trails, and cosmic rays are replaced prior to resampling.

The NDWFS has developed various IRAF scripts – **MKBLEED4**, **XZAP**, **SATZAP**, **BLKZAP**, to name a few – which may be of use. These are described and available in [15]. Note that **MSCBLEED** and **CRAVERAGE** were inspired by some of these scripts and are nearly equivalent though not identical.

7. Astrometric Calibration

The astrometric calibration of astronomical images, whether from a single CCD or a mosaic of CCDs, is useful in its own right. However, for a mosaic it is fundamental to creating single images where the pixels from the different pieces of the CCD are in the proper relation to each other, to geometrically correct the pixel positions and sizes, and register and stack multiple exposures. The astrometric solutions contain the information about the relative positions and orientations of the CCDs in the camera as well as the orientation and scales in the camera and the optical distortions in the telescope/camera system. While it might be of interest for engineering purposes to separate the various components, for data reductions only the total transformation between image pixels and the sky is required.

The astrometric calibration is used both to compute the celestial coordinate of a point in an image and a point in the image corresponding to a celestial coordinate. In the latter case the celestial coordinates are often for stars or other objects as given in some catalog. The ability to go from the celestial coordinates of catalog objects to points in the mosaic image data is important for automatically centroiding objects in the exposure in order to refine the astrometry or to do photometry for comparison between different exposures.

An astrometric calibration means determining or refining a *world coordinate system (WCS)* function. The function maps pixels on the CCD to arcseconds along axes aligned with the celestial coordinate system and having an origin, called the *tangent point*, tied to some point in the field of view, called the *reference point*. This origin remains fixed relative to the field of view as exposures are made at different points in the sky.

Each CCD image in the mosaic has its own mapping function tied to the same tangent point. The functions include terms describing the rotations, scales, and optical distortions specific to that CCD. The arcsecond offsets from the tangent point are converted to celestial coordinates by assigning a celestial coordinate, called the *reference coordinate*, to the reference point of the observation. Details of how this type of WCS is defined for images may be found in papers by Calabretta and Greisen ([10,11]).

For mosaic data reductions we generally assume that the WCS functions, apart from the reference coordinate and rotation angle, are static so that they may be determined once at some point in the sky and then translated to other parts of the sky and other rotation angles. We call this the *initial WCS*. If possible there should be a different initial WCS for each filter. However, for a large filter set this may not be possible and the WCS calibration will apply an approximate transformation (primarily a scale change) to the initial WCS from a similar filter.

Note that, provided an observation includes sufficient catalog sources in each CCD, new WCS functions can be computed starting from the initial WCS. However, an approach that does not require many catalog sources is to apply a simple correction to the initial WCS that preserves the distortion pattern carefully derived previously for a suitable field with many astrometric stars. The correction is a simple linear transformation which is common to all the CCDs. The correction accounts for an origin shift, a field rotation, a scale change in each axis, and a skewing of the axes due to telescope pointing error, camera alignment and flexure, atmospheric refraction, and filter differences (if the initial WCS is for a different filter). This transformation can be derived from just a few sources distributed across all the CCDs.

Creating the initial WCS begins by obtaining a calibration exposure of a field with many unsaturated stars. Ideally this would be a field which has an astrometric catalog. However, for the practical purpose of getting a reasonable starting point, any field with many stars will work in combination with the USNO-A2 catalog. The first step is to identify the celestial coordinate of the tangent point. If the camera can be rotated then the point in the image about which the field rotates should be used. When there are rotationally symmetric distortions, such as a pincushion distortion, the celestial coordinate corresponding to this point of symmetry should be used. Ideally the distortion and rotation axes will be the same. If there are no other considerations then a point near the center of the field of view of the mosaic camera may be used.

A WCS is derived for each mosaic image by associating object coordinates from a reference catalog with the pixel positions of the objects in the image. The image positions would typically be obtained by centroiding on the pixel values. From the set of pixel coordinates and celestial coordinates it is then possible to compute a WCS function tied to the selected tangent point.

In computing the initial WCS function one must decide the order of the function. This is done by looking at residuals between the WCS function value and the reference coordinates as a function of distance from the tangent point. Typically the residuals and distances from the tangent plane are shown in arcseconds. The function order should remove any systematics in the residuals without over fitting the data. The order which is necessary will be primarily determined by the optical distortions. When the optical distortions have a known form, such a pincushion distortion, this can guide the selection of the WCS function, coefficients, and orders. If the distortions are minimal a simple linear function described by a transformation matrix may be adequate. The general linear transformation matrix incorporates scales and rotations of the axes.

Once initial WCS functions for each mosaic image are derived for some calibration field they may be inserted into each exposure, possibly with a rotation applied, and the reference coordinate for the exposure set based on the telescope coordinate. A predetermined offset between the telescope coordinate and the reference coordinate might also be applied. These steps can be done either by the data acquisition system or during data reductions.

The WCS calibration of the individual exposures consists of adjusting the starting WCS to account for errors in the reference coordinate, the filter, and atmospheric refraction. With a reasonable starting WCS it is possible to do this in an automated fashion. By reasonable we assume the reference coordinate is correct to within a few hundred pixels. For example with a scale of a quarter arcsecond per pixel this means the reference coordinate should be correct to something like an arc minute.

The first step is to get a list of catalog coordinates for objects in the field. This can be automated by computing the region covered by an exposure based on the initial WCS including uncertainties in the reference coordinate. This region is passed to software that automatically retrieves the coordinates (and possibly magnitudes) of all objects from a catalog. This might be done with software and catalogs tied to the data reduction system or by internet queries to a web-based catalog. The retrieved coordinates might be limited to brighter magnitudes if the catalog has magnitude estimates and the number of objects in the region is large. The USNO-A2 catalog will typically provide sufficient object coordinates for any point on the sky for wide-field mosaics.

Note that for the purposes of simply combining dithered images using the initial WCS, one can adopt one exposure as the coordinate reference and use a cataloging tool that produces celestial coordinates for the objects in the exposure based on the WCS. The cataloging could be done with a sophisticated object detection and cataloging task that understands the WCS (e.g. **ACE** [9]) or the simple method of interactively centroiding a sample of objects on an image display to produce a list of coordinates (e.g. **MSCZERO** §7.2). With the list of celestial coordinates for objects in the reference exposure the astrometric calibration proceeds in the same way as when using an external catalog of objects. In essence this procedure registers all the overlapping exposures to the WCS of the reference exposure without producing absolute astrometry. Given the availability and density of objects in the USNO-A2 catalog this is method is not the optimal way to reduce mosaic exposures unless the field of view is small.

There are two ways for automated software to determine the reference coordinate offset. One is to catalog the objects in the image, convert the pixel coordinates to celestial coordinates using the initial WCS function, and correlate these with retrieved catalog coordinates. The challenges with this approach are that the objects cataloged from the image may contain many objects not in the catalog and, because of bandpass differences, they cannot be matched solely using relative magnitudes. However, there are methods which have been developed for this type of catalog matching [12].

The second technique works in the opposite direction by converting the catalog coordinates to pixel coordinates for each CCD image using the initial WCS and then locating the objects in the image. Note that the WCS functions continue outside the actual region of the image pixels so that coordinates outside the field of view of an image will produce pixel coordinates which are not in the image. For a mosaic with non-overlapping fields of view the pixel coordinate will fall either in none of the images or in just one. The challenge in this approach is to avoid identifying the wrong object in the image. As with the first method one cannot rely on the magnitudes. An algorithm for finding moderately large offsets provided the initial WCS has a small rotation error is described in §7.3.

Instead of an automatic determination, or if the automated method fails, the exposure can be displayed and the user can identify an object with a known coordinate to set the reference coordinate offset. This might also include overlying catalog coordinates converted to pixel positions using the initial WCS and reference coordinate. The display tool can then let the user indicate the offset in some way that registers the overlayed pattern with the image.

These techniques, automated or interactive, ultimately produce a corrected estimate for the reference coordinate from which a paired list of catalog celestial coordinates and centroided image pixel coordinates is derived. If there are enough objects, well-distributed over each CCD image, automated software can derive new independent WCS functions for each CCD. Typically the new functions would be of the same type and order as the starting WCS. Such solutions will naturally account for all effects of the optics and atmospheric refraction.

However, there is no guarantee that there will be sufficient objects in an exposure, particularly if the WCS function must include significant optical distortions over the scale of an image, to allow independent derivation of new functions. In this case one assumes that the solutions derived carefully from an astrometric calibration field are better determined than can be obtained from an arbitrary field. However, with just a few objects a low order, global correction over the mosaic field can be derived and used to adjust the individual WCS functions. This is based on the idea that the WCS contains all the information about the telescope optics and detector layout which is fixed from exposure to exposure to some practical level. What needs to be determined by the WCS calibration is the precise pointing of the telescope on the sky, any rotation of the camera, global scale changes, and skewing of the axes. These terms are sufficient to describe the effects of atmospheric refraction and instrumental flexure.

The residuals between the offsets from the centroids of the objects in the image and the catalog coordinates as a function of offset along the two coordinate axes are used to make a least square fit for a general linear transformation. The general transformation allows skewing of the axes and scale changes along both dimensions as well as an origin shift. The fitting can be done automatically or interactively to allow the user to evaluate the result and manually delete bad positions. In an automated fit misidentifications, errors in the catalog coordinates, and high proper motion effects not accounted for in the catalog are dealt with by iterative rejection of outliers. Limits on the final root-mean-square (RMS) and the magnitude of the changes can be set by the observer to cancel automatic updating of the WCS and flag the exposure as one to be examined more carefully.

If the global linear transformation yields a small RMS, the WCS functions are updated. If the WCS function lends itself to modification of selected coefficients for rotation, skew, etc. then this can be done for each CCD. For more empirical functions, such a general polynomials, the WCS is updated using a grid of points to refit a WCS function of the same type and order. In this approach the pixel coordinate of every grid point is converted to astrometric offsets. The linear transformation is applied to these, and then the pairs of pixel coordinates and corrected astrometric offsets are used to refit the WCS function. This will maintain all the higher order effects of distortion while empirically applying the global linear transformation.

7.1 Creating a Mosaic Coordinate System: MSCTPEAK and MSCSETWCS

There are a variety of IRAF tools for creating an initial WCS from an astrometric calibration mosaic exposure and then inserting the solution in individual exposures with the reference coordinate set appropriately for that observation. A more detailed description and guide for doing this may be found in *Creating a Mosaic Coordinate System* [13]. In this section we summarize the **MSCRED** tools which are customized to make this easier.

MSCTPEAK is an interactive task which interfaces a WCS fitting engine, **CCMAP**, with identifying and centroiding objects in an image display. It does this for each image in a mosaic. An image is displayed with circles overlayed for objects in an input coordinate list that fall within the image based on some crude WCS. Currently setting this crude WCS from scratch is not easy. Interactive cursor commands are used to identify one or a set of circles with objects in the image. A fit is made for a WCS of the desired form. The fit is examined and adjusted interactively with graphs of the residuals. The function parameters may be adjusted for the best fit and deviant coordinates not rejected by sigma clipping may be deleted manually. When the fitting is completed the WCS is written to a database file and the image WCS is updated. After returning to the display the coordinate overlays are updated allowing the user to iterate further if desired.

The end result of using **MSCTPEAK** is an IRAF database file with WCS solutions for each CCD image which may be inserted into different exposures using the task **MSCSETWCS**. This task sets the WCS keywords in each mosaic image with the appropriate solution for that image. It can account for different binning and trimming of the exposures. It also sets the reference coordinate using keywords in the headers giving the telescope or observation coordinates. An offset to the header coordinates may also be specified.

When there are multiple amplifiers per CCD the images should be merged first. This allows more stars to be used for the WCS solution and enforces continuity of the WCS across the whole CCD as must be the case. A WCS for each CCD is determined. The database entries need to be duplicated so that each amplifier from a CCD has an identical entry. The software will automatically deal with what region of the CCD is readout (provided all the appropriate header information is present describing the readouts).

The NOAO mosaic camera data acquisition systems [3] automatically insert previously derived WCS solutions when the exposures are acquired. The database files and **MSCSETWCS**, mentioned above, are not used directly. Instead the WCS keywords from a calibrated exposure are copied into the headers of the exposure with the reference point coordinate set using the telescope coordinate. There are appropriate adjustments made for precession, the overscan region, binning, subregion readouts, and multiple amplifiers.

7.2 Interactive Examination and Zeropoints: MSCZERO

MSCZERO is a very useful tool to both verify a WCS calibration by overlaying circles at the positions in the images corresponding to coordinates in a coordinate file and to apply a zero point correction to the reference coordinate. This is a combination of a display tool, which displays mosaic exposures in an approximate tiled representation of the field, a zero point calibration tool, and coordinate overlay, examination and editing tool. The task displays the exposure, if it is not already displayed, and then reads the image display cursor for commands to do various things related to coordinates. This include retrieving and overlaying catalog coordinates, reporting the coordinates of points in the images (either with or without centroiding), measuring and applying a zero point offset to the reference coordinate, and writing coordinates to a file based on the current WCS. The zero point offset may be set by selecting an object in the field and entering its coordinates or by overlaying catalog positions and identifying one of the positions with an object.

Applying a zero point correction with **MSCZERO** is not required to use **MSCCMATCH** to calibrate the WCS. But if the pattern matching algorithm for determining large offsets in the reference coordinate fails then **MSCZERO** can be used to adjust the zero point more closely before running **MSCCMATCH** again.

7.3 WCS Calibration: MSCCMATCH

The IRAF task **MSCCMATCH** computes and applies a common, global, linear correction to the initial world coordinates systems of each CCD image in order to match a set of astrometric celestial coordinates for objects in the field of the entire mosaic. Note that this does not mean each CCD will then have a common WCS but only that each is corrected by the same amount. The fact that this is done across all the CCDs with a small number of terms in the correction means that this task works with even just a few object coordinates which are not necessarily in all of the CCDs. To do a full WCS solution for each CCD requires many more objects. In the future this task will allow doing a full solution if there are a sufficient number of catalog objects, but currently such a detailed adjustment of the WCS must be done with the interactive task **MSCTPEAK**.

Many of the IRAF mosaic reduction tasks require a list of catalog coordinates for objects in the exposure to be calibrated. The list may include objects outside the field

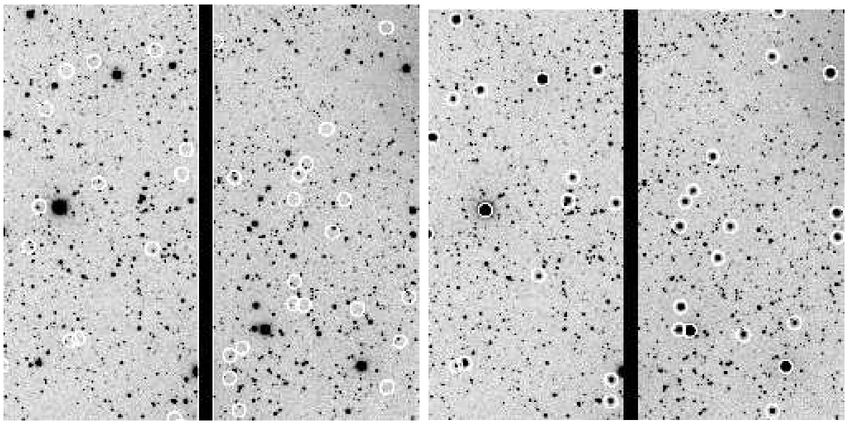

Figure 7: Two views of the same subregion of a NOAO mosaic exposure using the task **MSCZERO**. The overlays are of the brighter USNO-A2 object coordinates obtained automatically from a web-based catalog by the task. The panel on the left shows the initial WCS using the telescope pointing coordinate. One could use **MSCZERO** to set the initial zero point offset error but in the panel on the right the offset was found automatically using **MSCCMATCH** as described in §7.3.

which will be ignored. The catalog coordinates are a simple text file that the user can obtain in whatever way they want. The task **MSCGETCATALOG** is available to retrieve a text file of coordinates from the USNO-A2 catalog (though other catalogs may be added in the future) based on the current WCS with some margin for error. This task is used by **MSCZERO** to automatically get USNO coordinates and can be used by **MSCCMATCH** by specifying a command instead of a file as the coordinate input.

MSCCMATCH assumes that an initial WCS of the desired type exists but needs to be adjusted by a global linear correction. In outline the task does the following. The input object coordinates are transformed to positions in the images using the initial WCS. A pattern matching algorithm finds a shift in the image positions to match the object positions in the images. The positions of the objects in the images are refined by centroiding on the image profiles. The image positions are transformed back to celestial coordinates using the initial WCS. A linear transformation between these measured image coordinates and the input coordinates is derived. Finally, the transformation is applied to correct the initial WCS for each CCD image so that the measured image positions now correspond to the input coordinates. The transformation between the measured coordinates and the input coordinates includes terms for an origin offset, rotation, and scale change for each axis independently.

The hardest part of this to automate is finding the objects in the images when there is a moderately large error in the telescope coordinate used to initially set the reference coordinate. If the initial image position estimates are not close to the objects then blind centroiding will not work. In this case a pattern matching algorithm is required to find the objects. The most novel aspect of **MSCCMATCH** is its pattern matching algorithm.

The earlier discussion mentioned that different pattern matching methods might be used. **MSCCMATCH** uses a method that is not very general but is efficient and robust

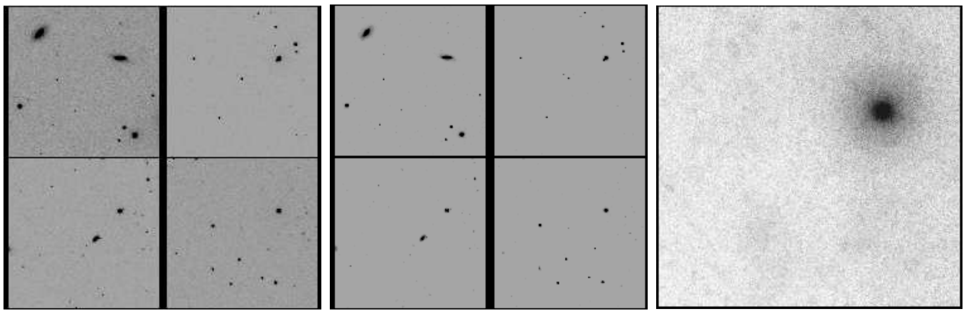

Figure 8: The left and middle panels shows four regions from an NOAO mosaic exposure centered on the initial guess for the position of objects from the USNO-A2 catalog. The size of the boxes is 200 arcseconds or 790 pixels. In the left panel the regions are shown as images and on the right the vote arrays where the background is zero and the brightest 10% of the pixels have values of 1. The right panel is the sum of the vote arrays over 60 regions. The regions were selected as the being the brightest objects in the catalog where the box is entirely within one image and where there are no bad pixels.

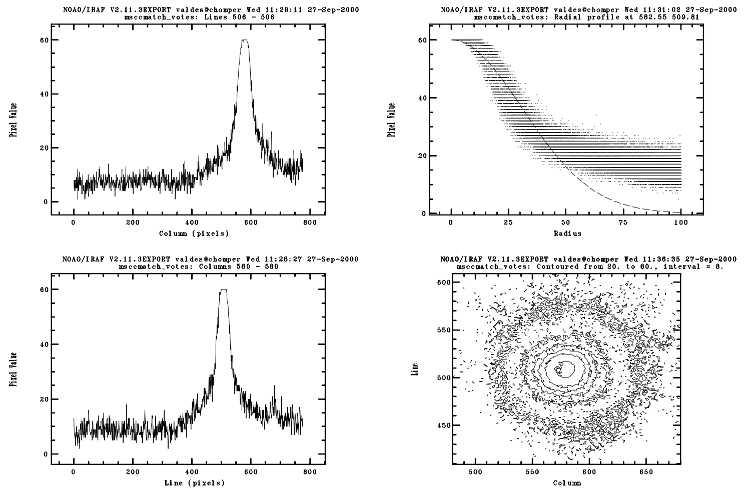

Figure 9: These plots are made from the final vote array shown in the right panel of figure 8. On the left are cuts along a line and a column near the center the region with the most votes. The right panels are a radial profile and contour of the region. The final position offset is the centroid of values above 30 votes (which eliminates almost everything away from the region of high votes). The centroid gives an offset of (193,121) pixels which is within a pixel of the final offset determined by the fine centering of 700 coordinates.

when the WCS predicts pixel positions such that a uniform offset along the image axes and a small rotation is sufficient to find the objects.

The inputs to the algorithm are a list of celestial coordinates for objects in the field, an reference point offset search radius, a rotation search radius, and the number of objects to use. If the list of coordinates includes a third column with magnitudes the list is sorted by magnitude in order to select the brightest objects, otherwise the objects are used in the order of the input list.

The celestial coordinate of each object in the list is converted to a pixel position in one of the mosaic pieces using the initial WCS. If the pixel position is not in one of the images it is skipped. A box around the image position is defined extending to plus or minus the search radius along the two image axes. If any part of the box falls off the edge of the image the coordinate is skipped. If the box contains any bad pixels, as identified in a bad pixel mask, the object is skipped. The selection of the boxes continues until the specified number of objects is reached. The left panel in figure 8 shows a selection of boxes a mosaic exposure.

The basis of the algorithm is that the objects in the coordinate list will fall somewhere within their box and they will be offset from the box center by the same amount. Note that spatial variations due to distortions have been taken into account to first order by the initial WCS which includes the distortions in the mapping between the sky and the pixels in the mosaic images. It then follows that by stacking the boxes the position where the objects in the list fall will accumulate coherently while other random objects in the boxes will be washed out.

Rather than directly stacking the data a *vote* array is used. The data in each box is converted to values of 0 (no vote) and 1 (a vote). The pixels with values above the 90th percentile of the data in that box cast a vote for their position in the box. The reason for using a vote of 1 instead of the pixel values is to avoid giving undue weight to bright random objects and to allow the objects in the list to be largely independent of magnitude. The middle panel in figure 8 shows the vote arrays for the four regions shown in the left panel. The right panel is the accumulated vote array for 60 regions. The final vote array is also displayed graphically in figure 9.

Once the vote array is accumulated, all positions where the number of votes is more than half of the maximum number of votes (which is the same as the number of boxes or object positions) are used in computing a centroid. If there are no points with votes above the threshold the algorithm tries again with a search radius which is twice as large. If it fails a second time then the algorithm reports a failure. If it does find offsets for the two axes and either is more than 80 percent of the search radius from the initial guess, the offsets are applied and new offsets are redetermined by running the algorithm again. This is to correct for possible edge effects in the centroiding of the vote array.

The extension to allow small rotations is to define a set of rotations spanning the rotation search radius. A vote cube is used where the third dimension is related to the rotation. For each candidate rotation the pixels in the box are de-rotated about the reference point before casting votes at that rotation. Note that the boxes remain the same so that the data is only read once. It is the positions in the vote array for each box that are adjusted. After accumulating votes at each rotation the vote cube will produce a centroid in three dimensions which define the x and y offsets and the rotation.

After the coarse determination for the reference point offset and rotation is completed, the entire input coordinate list is converted to pixels in the image based on that correction. Note that if no coarse search is specified, by using a search radius of zero, then the task begins at this point. A small centroiding box is used. Again boxes that don't fall fully within an image or have bad pixels are excluded. The centroid of the data in each box is computed using a standard IRAF centroiding algorithm; the centroid algorithm of

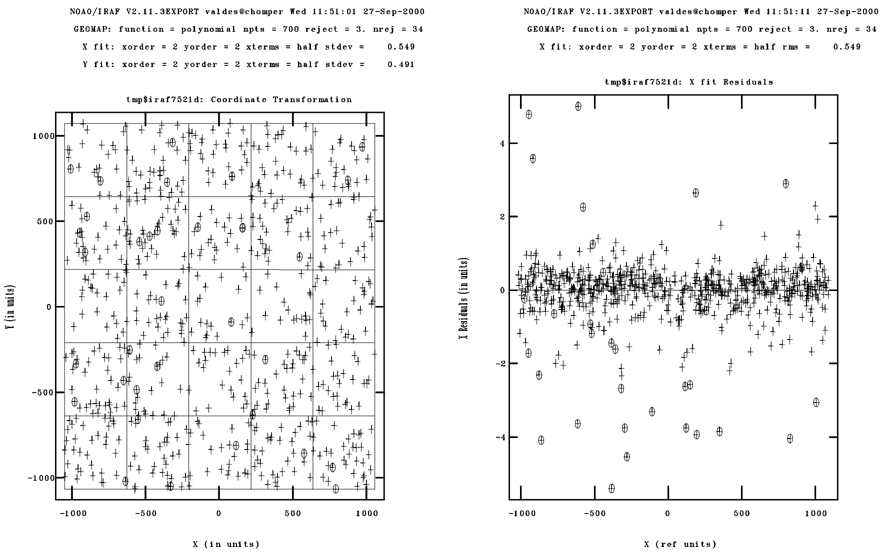

Figure 10: Two graphs from the interactive fitting of the global, linear transformation in **MSCCMATCH** using a large number of centroided objects across the field of view from all eight CCDs. The graph on the left shows the location of the centroided objects and the figure on the right shows one of the four possible residual graphs. The plus symbols are coordinates used in the fit and those with a circle are those rejected by sigma clipping. The coordinate units are arcseconds from the reference point.

the **APPHOT.CENTER** task. This algorithm takes the marginal distributions of the data in the box along each axis and computes the centroid using data above the mean of the distribution. The algorithm iterates using the new center until a consistent center is found. The centering may fail for various reasons. Those objects whose centering fails are excluded. Those which are successfully centered are converted back to celestial using the initial WCS.

If too few of the objects are found **MSCCMATCH** will report a failure. Otherwise the corrections to the WCS are determined from the original coordinates and those derived from the image centers expressed as distances from the initial WCS reference coordinate. A a linear transformation between these coordinates is determined using the iterative least squares fitting task **GEOMAP**. This may be done automatically or interactively. Figure 10 shows two views from the **GEOMAP** task when the interactive option is used.

MSCCMATCH reports the transformation in terms of offsets, rotations, and scale changes and also the RMS of the fit as shown in figure 11. The fit can be automatically accepted or rejected based on the RMS and the minimum number of objects found or the user can be queried.

The method for applying the correction to the WCS of each CCD was outlined previously. In order to preserve the distortions, and since a general polynomial WCS function does not have specific coefficients which can be identified with the rotations and scale changes, a new function is derived of the same polynomial type and order. This is done by computing the celestial coordinates of a grid of points spanning a CCD image using the initial WCS. The global adjustments of the celestial coordinates are applied to these values. The new WCS is derived using the grid of pixels coordinates matched with the corrected celestial coordinates. The new solution will incorporate the same distortions as in the initial WCS but with the linear transformation applied.

```
ms> msccmatch
List of input mosaic exposures: obj068pfsz
Coordinate file (ra/dec): usno
MSCCMATCH:
  obj068pfsz:
    1582 input coordinates
    829 coordinates out of bounds
    search using up to 60 objects:
    search found offsets of (193, 121) pixels and rotation 0.00 degrees
    1582 input coordinates
    831 coordinates out of bounds
    Fit coordinates:
      input number of coordinates = 700
      tangent point shift = (-31.03, -49.02) arcsec
      fractional scale change = (1.000, 1.000)
      axis rotation = (0.007, 0.001) degrees
      rms = (0.549, 0.491) arcsec
Accept solution?: yes
```

Figure 11: An example of running **MSCCMATCH**. The bold type is the user input and the other is the task output. The output provides information about the number of coordinates input and the number used as well as information about the coordinate offsets, fit results, and corrections computed.

8. Resampling and Single Image Reconstruction

A CCD mosaic exposure consists of a number of separate images, each with its own distortions. For some science programs it may not be necessary to make a single image out of the pieces or even to remove the distortions, such as a variable pixel sizes on the sky, in the individual images. Instead each image can be analyzed independently and the distortions taken into account using the calibrated WCS. This avoids resampling effects.

However, other programs, particularly those requiring the full field of view without gaps, require making single, geometrically correct images from the individual mosaic exposures or sets of mosaic exposures. Making an image from a single exposure is probably not very useful. Geometric correction of the individual images is also not usually a concern, though some software may not properly treat distortions and variable pixel sizes using the WCS. Being able to combine dithered exposures which have been resampled into single images with a uniform and registered pixel sampling *is* useful in order to fill in the mosaic gaps, eliminate bad pixels, create deeper images, and apply algorithms, such as automated cataloging, that require or produce better results from single contiguous images with uniform pixel sizes.

There may be more than one image per CCD if multiple amplifiers are used. Putting the amplifier images from the same CCD together is just a simple matter of pasting the pieces together into a larger array without interpolation or resampling. The result is a single CCD image that is identical to what would have been obtained if only one amplifier had been used. The reason each amplifier is initially read out into different images is to allow independent overscan and prescan data in the image and keywords in the headers. The merging of multiple amplifiers may be done during the basic CCD calibration after the overscan and prescan data are used and discarded (§3). While the resampling algorithm described in this section could also be used to put the amplifier pieces together, it is not recommended because boundary effects between the amplifier regions will be unnecessarily added.

The removal of geometric distortions in the elements of a mosaic and the reconstruction of a single image from the pieces is straightforward provided the data has a calibrated world coordinate system for each CCD image. A continuous grid of pixels on the sky (that is in celestial coordinates) is defined and filled with data from the input mosaic pieces. A single image may be made in one step by resampling all the input mosaic images to a single output image or in two steps by resampling to separate output images, all relative to the same larger grid of pixels on the sky, and then putting the pieces together. In the second case, the individual pieces will be registered after the resampling except for integer offsets along each image axis. Putting the pieces together then simply requires applying the offsets when adding the pieces into a larger single image. This last step does not require any further resampling.

A variant of this is to entirely skip making single images from the individual mosaic exposures. Instead all the overlapping mosaic exposures may be resampled to new geometrically corrected mosaic exposures where all the pieces are registered to the same grid on the sky apart from integer offsets. Then a final single image may be made by aligning all the pieces from all the exposures, using the integer offsets for each piece, and combining them. While this approach is possible, some of the steps described below for determining and removing sky gradients and photometric transparency variations still need to treat each exposure as a single entity. The software that does this may require each exposure to be a single image. This is currently the case for the IRAF tasks, though the generalization to working with the data in mosaic form would not be difficult. There is no advantage to keeping all the mosaic images separate as opposed to making single images of each mosaic exposure. There is a disadvantage for the case where the output sky grid is rotated away from 0 or 90 degrees relative to the original image axes because of wasted space in the corners.

Whether or not a single image is made from a mosaic exposure the discussion of resampling applies. The next section (§9) discusses putting together multiple images that have the same world coordinate system by offsetting and combining.

Defining a grid of pixels for the resampled images is done by specifying an output world coordinate system. The WCS is typically, though not necessarily, a simple linear function with a single scale and orthogonal axes. The orientation may be set to the common convention of north up (declination increasing along the image line axis) and east to the left (right ascension decreasing along the image column axis).

The output WCS may be defined by adopting an existing WCS from a *reference* image, such as a previously resample image, so that the new images will be registered with that image. Alternatively, one may specify parameters such as the scale in arcseconds per pixel, the position angle of the declination axes relative to the image line axes, and the reference coordinate. In either case, using the same output WCS for all related mosaic exposures will produce resampled images which are registered with each other. In order to save space, the size of each output image is set to be just large enough to include the data from the input image or images. This means that the output images using the same WCS will be offset from each other by integer amounts along both image axes. These offsets can easily determined from the WCS of the output images when registering and combining them.

The input and output pixels are related by mapping a point in one image to celestial coordinates using its WCS and then applying the inverse transformation using the second image's WCS to map the celestial coordinate to a point in the second image. This can be done in either direction. Software will most likely combine the two WCS transformations into a single transformation between pixels in the two images before resampling the pixels.

There are two common ways to resample the input pixel values into an output image. One is to model how light in an input pixel divides into the output pixels which it overlaps. The *drizzle* algorithm [14] is an example of this type of resampling. The other is to *interpolate* the input data to the point in the input image which corresponds to the center of the output pixel and use that for the output pixel value.

In either case the relative areas of pixels in the input and output images need to be taken into account. This is called *flux conservation.* Flux conservation conserves the amount of light in the area covered by the output pixel. The alternative is to conserve the surface brightness. For the most common situation of flat fielded data where, by construction, the sky brightness has been made (artificially) uniform regardless of the input pixel sizes, and when the output pixel sizes are defined to all be the same size on the sky, surface brightness conservation (no flux conservation) is the proper estimate. When the flat fielded input data has been corrected to true flux per pixel, as described in §5.2, flux conservation needs to be used.

There are a variety of interpolation schemes ranging from nearest neighbor, to linear, to polynomial, to sinc. There are three important characteristics of interpolation. Interpolation smooths the data, rings around sharp features, and correlates the noise. These tend to play off each other so that a method such as sinc interpolation, which minimizes the noise correlation, has the most drastic ringing.

To minimize the effects of ringing it is important to identify sharp features in the mosaic images before interpolation. These are identified in the bad pixel masks (§6). Ideally the interpolation scheme would avoid these pixels. But in cases where this is not done, such is currently the case in the IRAF software, the sharp features must be removed by replacing them with interpolated values from nearby pixels (§6.1).

One other place where discontinuities occur are at the edges of the images. However, this can be handled in the interpolation algorithms by using boundary reflection for the off-edge pixels required by the interpolation function. Alternatively the edge can be avoided entirely by resampling only the portion of the input image that does not require out-of-bounds pixels for the interpolated value. The latter case is recommended though in either case the bad pixel mask will still identify the affected pixels.

The best interpolation method is probably sinc though it is also the most time consuming. Because sharp features in the data being interpolated ring and adversely affect large regions (as large as the size of the sinc function used) it is vital to remove these in the input data. All the common interpolation schemes introduce unavoidable smoothing. Sinc interpolation best preserves the noise properties and, if care is taken to mask and remove sharp features, it is the one which is recommended.

The resampling operation and reconstruction of single images from a mosaic make the pixel masks invalid unless the software is clever enough to map back through the WCS to the original masks. But what constitutes a bad pixel in the resampled data? The resampled values will generally have contributions from a number of input pixels as described previously. When one or more of those pixels is identified as bad then the question is whether the resampled value should be considered bad. This can be rephrased as whether the resampled value is *significantly* affected by the bad pixel in the input data. Ultimately it is up to the user to decide what is significant.

If the resampling algorithm ignores bad pixels in an unbiased way and defines what constitutes a significant error in the resampled value then that would be the best definition for a bad pixel. However, if the algorithm does not support special treatment of bad pixels (as is currently the case in IRAF) then a different approach is required. One choice is to simply say that if any bad pixel or out of bounds pixel is used by the interpolator the output pixel is identified as bad.

A mask based on this principle can be produced by resampling the input mask(s) to the output mask using the same resampling as the data (except without flux conservation). Regions of the mask with only good values (zero values) will produce output mask values which are also good and regions that have bad pixels (non-zero values) will produce output mask values which are also non-zero. The degree to which they are non-zero is a measure of how much the bad pixels affect the resampled value. By setting criteria on the resampled output mask value one can determine how significant the effect of the bad pixels is on the resampled value.

In addition to identifying the resampled pixels which are affected by bad pixels in the input mask, the resampling can also assign special mask values to pixels which are have no contribution from input data. This is needed to identify the gaps in reconstructed images and the regions at the edge which have no data due to rotations.

Defining any non-zero resampled mask value obtained by resampling the input mask in the same way as the data is reasonable with low order polynomial interpolators or drizzling. However, with sinc interpolation a bad pixel in the input data affects a large number of output pixels even if only slightly. Since it is recommended that bad pixels be replaced in the input by suitable smooth values, the sinc interpolated pixel value where the bad pixel is in the low weight wings of the sinc function will have little effect on the output value. For this reason the recommended method for propagating bad pixels with sinc interpolation is to identify output bad pixels as only those where the input pixel is near the corresponding output pixel position. This can be done by using a different interpolation function on the mask which is based only on relatively nearby pixels. Linear interpolation is a reasonable choice though others might be used. With this method it is required that ringing be minimized by removing all cosmic rays and other sharp features to avoid affecting output pixels which will be treated as good data.

8.1 Resampling Mosaic Exposures: MSCIMAGE

The task **MSCIMAGE** resamples mosaic images into a single image or a new mosaic with a particular target world coordinate system. Figure 12 shows an example of a single image produced by this task. The target world coordinate system is specified by a reference image, by parameters for the scale, rotation, and reference coordinate, or by some combination. The reference image might be a previously produced single image from **MSCIMAGE**, a dummy image of zero dimension but with a WCS, or some target image with a WCS from another source. If no reference image is specified but is required to define all or part of the WCS, the mosaic image in the first input mosaic exposure nearest the reference point is selected. When a reference image is used the linear part of the WCS is extracted. This means that distortion terms in the output WCS are not allowed.

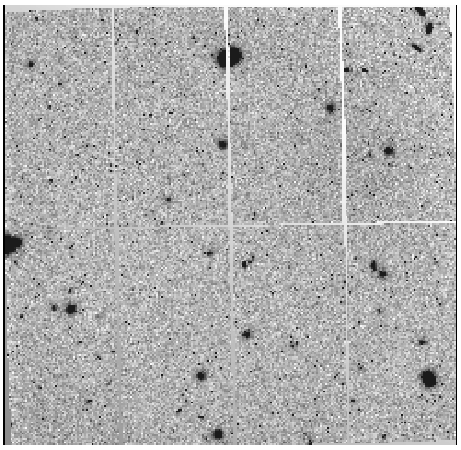

Figure 12: Example of a single resampled image produced by **MSCIMAGE** from a mosaic exposure. Note the rotation of the whole field caused by using a reference WCS that is exactly aligned with north (to the left) and east (down). It also shows the slight relative rotations between the pieces as well as some of the pincushion distortion. The gaps are given a constant value and by comparison with the image data it is clear there is a small gradient in the sky. A bad pixel mask is also produced that identifies the gap, bad pixels, and saturated pixels.

After the reference WCS is defined, the maximum size of an output image which just contains the input data, after being trimmed to avoid interpolation outside the image, is computed. Note that the output image will not be registered in pixel space but instead share the same WCS grid on the sky. This means that the output images can be registered by simple integer shifts along the two image axes. These shifts are generally determined automatically from the WCS by the software that need to register the images (§9.3).

A transformation between pixels in the input and output image is determined by combining the forward and reverse WCS transformation of the input and output image. The input image is then sampled to produce the output pixels. The sampling or interpolation function is specified by the user and there are a number of choices from simple to complex. This includes simple linear, drizzle, and sinc. Another option is whether to apply flux conservation to account for pixel area changes as discussed in earlier sections.

Boundary effects in the resampling are dealt with in two ways. First, if the interpolation requires data outside the input image a boundary extension parameter defines what values will be used. The recommended choice is "reflect" which means the out of bounds data will be a reflection into the in bounds data across the image edge. This is a good choice since it will be continuous across the boundary thus minimizing ringing in the interpolation function. Other options include a constant value and wraparound though these are not recommended.

The other option is to exclude output pixels which require out of bounds pixels in the interpolation function. This is done by specifying a trim region around the edge of the input images. It is the trimmed data that is resampled but the data outside the trim is available to the interpolator. Note that the trimming may also be used to exclude bad data at the edges. The amount of trimming required to avoid edge effects is interpolation function dependent. When sinc interpolation is used then it must be at least half of the size of the kernel function. It is recommended to use the trimming since it makes the least assumptions about the edge and avoids interpolator ringing.

The IRAF supported resampling interpolation functions do not consider bad pixels. For this reason the input data should have bad pixels identified by their bad pixel masks replaced by reasonable interpolated values. But it is also desirable to produce an output bad pixel mask indicating which pixels are affected by bad pixels in the input data. This is done by resampling the input bad pixel masks in exactly the same way as the data, though without flux conservation and possibly using a different interpolation function.

MSCIMAGE converts an input bad pixel mask into one with zero for good pixels and 10,000 for bad pixels. In regions where there are no bad pixels, the input masks will be uniformly zero and the interpolated value will be zero. However, if there is a non-zero value within the range of the interpolation kernel it will produce a non-zero value. The reason for the large value of 10,000 is that pixel masks are integer valued and any interpolated value between -1 and +1 will produce zero. In other words, interpolated values which have less than a millipercent effect will have a mask value of zero which identifies the pixel as good.

When the interpolation function produces values outside of the pixel mask being resampled it assigns values of 20000. This is done to allow separating mask values into those due to bad pixels in the input mask and those due to being out of bounds. After the output bad pixel mask is created then absolute values between 1 and 10100 (to allow for ring) are set to 1 and pixels above 10100 are set to 2 to produce a final mask with 0 for good data, 1 for bad pixels in the data, and 2 for no data. The separate output of bounds values are needed in order to pass through resampled values of the bad pixel while converting areas where there is no data to some specified output blank value when putting the separately resampled mosaic images together into a final single image.

As discussed previously, sinc interpolation will indicate that an output pixel is bad if there is a bad pixel within some large region around the nearest input pixel. To reduce this effect the user has the option to use linear interpolation for the mask even if the data is interpolated by a sinc function. This will result in output pixels being identified as bad only if there is a bad pixel within one pixel of the nearest point in the input data. One could also use a cubic or quintic interpolation function to set a somewhat larger region that is still small with respect to the typical sinc function which has a radius of around 10 pixels.

The above steps are done on each image of the input mosaic to produce resampled images in a new MEF mosaic file. The user may choose to stop there along with the bad pixel masks described earlier. However, it is recommended that the output "image" option be used. In this case the resampled mosaic file is pieced together into a single image and the resampled bad pixel files for each image are pieced together into a single bad pixel mask. The output bad pixel mask in this case will have zero for good data and one for both bad pixels and no data.

The version described here is an improvement over earlier versions where it was not possible to produce a resampled mosaic format or to have an output WCS with any rotation relative to the data. This latest version is completely general and will allow a common desire to resample directly to a standard orientation regardless of the orientation of the observations.

9. Combining Dithered and Survey Mosaic Exposures

Mosaic exposures are combined or stacked (the terms are interchangable) for a variety of reasons. These include increasing the depth, increasing the field of view, and filling in the gaps between the individual CCDs and bad pixels. Dithering (taking multiple exposures with shifts of a small fraction of the field) is used to remove the gaps and bad pixels while increasing the depth of the final image. Shifting by large fractions of the field of view is used to increase the field of view for programs such as surveys. An ideal program combining these two types of shifts is to take a number of observations, say 5, with a small dither pattern and then offset to a slightly overlapping field and repeat the dithering. This is how the NDWFS is designed. Note that if the fields are not of very large galaxies, the collection of exposures with different pointings can also be used to create dark sky flat fields §5.

The requirement for combining dithered and offset mosaic exposures is to accurately match them astrometrically and photometrically. The astrometric matching or registration is accomplished using the astrometric calibration and image reconstruction steps described in §7 and §8. The one special point for disregistered data is that the image resampling for all exposures must be done to a common WCS. The observer can define the common WCS or the software can use a WCS from the complete collection of mosaic pieces to define a center and average scale and orientation.

By a common WCS we not only mean the same orientation and scales but the same reference point. This logically creates a common grid of pixels on the sky into which each exposure is resampled. When the individual exposures are resampled they can be cropped to a size which just includes the field of the exposure. There is no need to have each resampled image fill the full final field until they are combined. The various single images are then registered except for integer offsets along the image axes. The offsets are automatically determined from the WCS when they are combined.

Photometric matching consists of adjusting the background sky to a common level, possibly requiring the removal of gradients, and adjusting the gains so that common objects have the same instrumental flux. The sky adjustment is an additive correction and the gain adjustment is a multiplicative correction. In principle the flat fielding will have adjusted the sky and gain of the mosaic pieces within a single exposure to common values. It is much more difficult to do the photometric matching if this is not the case. In the following discussion we will treat this as true.

The additive correction is required because the sky brightness varies with time. So even if the exposures are obtained with the same exposure time and even near the same observation time the sky will not be identical. The multiplicative correction is required because the transparency of the sky varies with time and zenith distance.

The combining operation can apply the scaling corrections while combining the data or one can modify the individual images by applying the corrections to produce corrected versions of the images. For example the sky can be determined and subtracted from each image to produce images with zero sky and the images can then be multiplied by the gain scaling to produce images with common photometric gains. One might choose to do this for other reasons but since the combining can apply these corrections at the same time as it is producing the final image there is no reason.

The exception would be if the combining software only includes a simple linear transformation of the pixels consisting of a single sky brightness and gain correction. In this case if there are sky gradients they have have to be removed first. One could either subtract the sky to a zero level or subtract only the spatial gradients by adding back a mean sky level. Mapping of spatial variations in the gain might be handled separately but generally one assumes there are no such variations (which would be difficult to measure).

While it is possible to use the photometry of common objects to simultaneously derive the additive and multiplicative corrections, the better method is to measure these separately. The sky brightness can be measured from a single exposure by fitting a low order function with some method for eliminating the effects of the objects, gaps, and bad pixels in the field. The gaps and bad pixels are taken care of with the bad pixel masks. The objects may be eliminated either by creating a mask using an object detection program such as **ACE** [9] or using absolute and iterative clipping. Note that object detection programs also produce measures of the sky which might be used. Absolute clipping means to ignore all pixels above some value and iterative clipping means to reject the residuals of the fit which deviate by some number of standard deviations (as determined from the residuals averaged over a larger region) and then repeat the fitting.

The end result of the sky brightness measurement is a constant or low order function. When a low order function is fit the gradient component can be removed by subtracting the fit and adding back the mean. The reason this might be desirable is to keep some knowledge of the photon counts for Poisson statistics. The mean sky is written to the image headers for future reference.

Determining the relative gains is done using photometry of matching regions which are typically centered on objects in the images. Note that it is not necessary for the same regions to be in every exposure but just that every exposure have some number of common regions to photometer and compare to some other exposures. The algorithm described in §9.2 takes the ensemble of measurements of all pairs of overlapping regions and finds a least squares solution that simultaneously minimizes the differences in the photometry with a single set of gain corrections.

While one could pick matching regions at random, obviously most of the sensitivity to gain variations lies in the brighter objects. One must also account for variations in the point spread function in the photometry. What this means is that common photometry apertures for objects such as stars must be large enough that PSF variations do not alter the measured flux within the aperture significantly. Therefore apertures need to be centered on objects rather than randomly placed.

To obtain the regions to be measured and to match the regions between images we again rely on the world coordinate systems and catalogs of object celestial coordinates. The catalogs may be the same ones used for WCS calibration and obtained from an on-line catalog. However, it is possible to also catalog the sources from the images using an object detection program or interactive selection from a display of the data. To speed the calculations one might limit the number of objects by using just the brighter ones.

Given a set of coordinates and aperture sizes, simple boxes are adequate, covering the entire ensemble of exposures the WCS is used to translate them to pixels in the images. If an aperture doesn't fall completely in a particular image it is not used. The aperture is checked for bad pixels based on the bad pixel mask for the data. If the aperture contains any bad pixels the object is excluded. This eliminates saturated objects if the mask includes saturated pixels identified during the basic CCD calibrations. It also eliminates objects where the aperture falls over a gap. The instrumental flux in the aperture is then measured. Each object coordinate can be used both with an aperture containing the object and an annulus around it to sample photometry near the sky.

For each pair of exposures all the common objects are used to determine a linear relation in the photometry. When the sky brightness has been previously determined the origin of the relation is fixed appropriately, otherwise it is a free parameter and both the gain and sky brightnesses are determined simultaneously. Photometry measurements from an image may be excluded by sigma clipping outliers from a linear fit. Once a measurement is excluded from one image based on comparison with another image then that measurement will also be excluded from all pairs. Once all the pairs of images have gain and possibly sky brightness estimates, the algorithm must combine all of them in some simultaneous least squares fashion to get a single gain and sky brightness value for each image. See §9.2 for how this may be done.

Once the additive and multiplicative scaling factors for sky brightness and photometric transparency are derived, the individual exposures can be combined. This involves determining integer offsets along both axes to register the images, creating an output image that contains all the input data, applying the scaling factors, collecting pixels from all the images that overlap an output pixel, rejecting pixels based on various criteria and algorithms, and computing the average or median of the remaining pixels. Rejection of pixels can be done using the bad pixel masks, data value thresholds, and statistical clipping. The clipping can be done based on a noise model, rejection of a set number of high and low values, or by deriving standard deviations from the data. Neighbors of rejected pixels may also be rejected.

In addition to the final image it is desirable to produce some auxiliary data. This includes a mask identifying where data was combined, an exposure map giving the total effective exposure time at each pixel, and some information about how many or which pixels were rejected.

Rejection of transient objects, such as cosmic rays, satellite trails, and asteroids, using statistical clipping is something to be done carefully if at all. This is because in the presence of PSF variations, mostly due to seeing, the cores of objects such as stars will vary more between images than than the clipping threshold based on the noise statistics. So valid data will be systematically clipped which adversely affects the shapes of the objects and biases the photometry. Therefore, if clipping is done it should use large clipping limits. What is better is to identify the transient objects separately and include them in the bad pixel masks before combining.

One method of doing this is to go ahead and make a combined image with fairly severe clipping. Then subtract each individual image from this combined image. This will produced images with most objects which are in every image subtracted but leaving things such as cosmic rays, transient sources, and moving objects. An automatic detection program can then find these sources. The variable PSF induced residuals in the cores of bright and small objects will be found along with real transient objects. They can be eliminated by comparing the integrated flux of the residual object to the flux in the combined image. When the ratio of these fluxes is less than some threshold the object is not considered to be a transient object. The regions occupied by the remaining sources are added to the bad pixel mask for that single image along with some increase in the area to eliminate wings in the light profile. An example of this technique is given in [9].

After all the images are processed in this way a new combined image with little or no clipping can be produced using the bad pixel masks to exclude the objects. Note that the detected objects in these difference images may be classified and cataloged. The catalog of non-cosmic ray transient and moving objects is an interesting result in itself.

The optimal extension of this procedure is to add the objects found by this difference method, particularly cosmic rays, to the bad pixel masks of the unresampled data. The object is then removed, by interpolation, from the data before resampling data again. This will reduce the ringing artifacts that are associated with undersampled cosmic rays.

9.1 Determining the Mean Sky and Removing Gradients: MSCSKYSUB

Images are combined by applying a single additive offset to match the sky levels. If there are sky gradients, such as caused by moonlight, the constant offsets will not match the skies everywhere in the image resulting in a poor combined image. To remove these gradients and, at the same time, make an accurate determination of the mean sky to use for setting the scaling offsets, the task **MSCSKYSUB** is used. This task is a variant of the general surface fitting task **IMSURFIT** that provides for excluding data in a pixel mask and for subtracting the fitted surface without changing the mean value.

The algorithm is to compute medians of a specified size in boxes across the image while ignoring bad pixels. A two dimensional function is fit to the median points, possibly with some clipping. The output is an image which may be the surface fit, the residual, the ratio, or the input with pixels with large residuals from the fit removed. When the residual or the ratio output are specified, the mean of the surface is subtracted or divided from the fit before applying the fit to the data. This effectively removes gradients without removing the sky level which is useful for noise estimation.

For mosaic reductions the output type should be the residual image. The mean of the sky fit is recorded in the image header under the keyword SKYMEAN which can then be used by later tasks to set the offsets between images for combining.

In this data reduction path the sky is subtracted in the single image formed by using **MSCIMAGE** to put the mosaic pieces into one image. This allows a continuous surface to be fit across the boundaries of the CCD. This is physically reasonable because the sky flat fielding removes the CCD-to-CCD gain differences and the remaining differences should be due to a sky gradient across all the CCDs. However **MSCSKYSUB** can also be used on the individual CCD images prior to **MSCIMAGE** if desired by specifying each extension explicitly. This will not necessarily produce a smooth across the boundaries.

9.2 Matching the Photometric Scales: MSCIMATCH

The task **MSCIMATCH** estimates additive and multiplicative scale factors between a set of overlapping images given a set of celestial coordinates. The coordinates are typically the positions of objects from the USNO-A2 catalog. The sizes of two concentric square boxes are specified. The WCS of the images are used to convert the celestial coordinates to positions for the box apertures in the image. If the inner box contains pixels which are off the image, either due to being off the edge or falling in a gap of a reconstructed mosaic image, or marked as bad or saturated in an associate bad pixel mask, both boxes are excluded in that image. If only the outer box has such data then it is excluded but the inner box is still used.

The reason two concentric apertures are used is that it is efficient to get the image data for the larger box and then compute two photometric values as the total counts in the smaller box and the difference between the counts in the smaller and larger box. When the boxes are centered on objects, each region then provides two photometric points, the flux of the object and the background around the object. If the outer box is excluded then there is only one measurement for that coordinate. In the end a set of fluxes, I_{in} where i is the image index and n is the aperture index, are obtained.

The scaling factors relate the photometry between an image i and and an image k such that the photometry obeys linear relations of the form

$$I_{kn} = a_{ik} I_{in} + b_{ik} \tag{3}$$

We can estimate the coefficients and their uncertainties for each pair of images using the regions, and hence measurements of the same source light, in common between the two. This is done using standard least squares. We determine both a_{ik} and a_{ki} and b_{ik} and b_{ki} as independent values. In **MSCIMATCH** we ignore uncertainties in the individual photometric measurements, that is give each measurement equal weight, and obtain an error for the coefficient based just on the scatter of measurements.

The fitting can be done using interactive or iterative sigma clipping to reject poor measurements, which might include bad data not in the bad pixel masks. When a measurement is rejected based on one pair then the region is eliminated from both images since which of the images has the bad data is indeterminate. However, the measurements from that region are not eliminated in any other image. When interactive fitting is done only a subset of the image pairs are displayed; each image is only shown twice, once as the independent variable and once as the dependent variable relative to the following

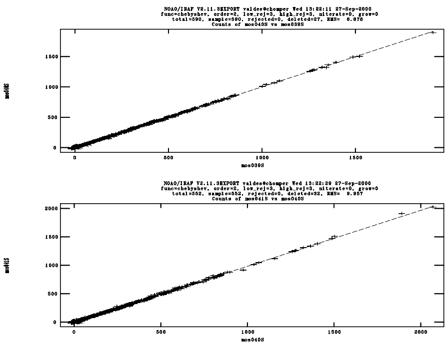

Figure 13: Example interactive graphs from **MSCIMATCH**. The marks are flux measurements inside square apertures or square annuli centered on coordinates from the USNO-A2 catalog. The graphs compare the fluxes for coordinates which are common to pairs of input mosaic images. The lines are least squares fits from which the a_{ik} and b_{ik} are estimated. In this example the mean sky has been separately determined with **MSCSKYSUB** and subtracted from the measurements so the fits are constrained to go through the origin and only the slope is determined. In interactive mode only graphs between successive images are shown rather than all combinations and the interactive step allows rejection of deviant points.

and preceding image respectively. Figure 13 show two examples of the data plots from pairs of image used to estimate individual relative scales and offsets.

As described earlier, if the additive offsets caused by sky brightness variations are obtained separately by fitting the background, then the coefficients b in equation 3 can be fixed and the least squares solution obtained just for a. This solution is more constrained since there are fewer degrees of freedom. This is what is recommended where the sky level is given by the keyword SKYMEAN as produced by the task **MSCSKYSUB** (§9.1). In the discussion here, however, we illustrate the algorithm with both coefficients determined simultaneously. Note that it is also possible to input values of a, the gains, determined independently and use this algorithm to find b alone. This something which is not likely to be done.

What we want to determine are the scaling coefficients relative to a reference image. Since the reference image is arbitrary we choose $i = 1$ and estimate a_{1k} and b_{1k}. We denote these estimates by A_k and B_k. A common way to estimate the coefficients is to look only at the transformations between each image and the reference image without regard to the relationships between other pairs of images. This is sensitive to the choice of reference image and any errors in the photometric calibration of that image will dominate. A more sophisticated method is to consider all the pairs and find the best overall estimate.

Equation 3 implies relationships between the coefficients. If we consider transferring the photometry from image k to an intermediate image j to get to image i we can derive the relationships

$$a_{ik} = a_{ij} a_{jk} \tag{4a}$$

$$b_{ik} = a_{jk} b_{ij} + b_{jk} \tag{4b}$$

One could consider looking at more intermediate steps but the number of combinations increase quite rapidly. In **MSCIMATCH** only the relationships given by equations 3 and 4 are used.

There are four ways to estimate the multiplicative scales A_k between the reference image and image k using at most one intermediate image j. These are a_{1k}, $1/a_{k1}$, $a_{1j}a_{jk}$, and a_{jk}/a_{j1}. To obtain an estimate for A_k we average the above quantities over all images j. Using the least squares error estimates for each of the measured quantities a_{ij} with standard error propagation, we compute variances for each of the four estimates of a_{1k}. The average of the variances divided by the number of values is then the variance of the average estimate of A_k. We also compute the standard deviations of the quantities that enter into the average. We then define a formal error as the sum of these errors added in quadrature.

When the zero point offsets are also determined at the same time the same method is used to estimate B_k and its error. The only difference is that now there are 15 ways to obtain an estimate for B_k using zero or one intermediate image. These estimates are b_{1k}, $-b_{k1}/a_{k1}$, $-b_{k1}a_{1k}$, $b_{1j}a_{jk}+b_{jk}$, $(b_{1j}-b_{kj})/a_{kj}$, $-b_{j1}/a_{j1}a_{jk}+b_{jk}$, $(-b_{j1}/a_{j1}-b_{kj})/a_{kj}$, $b_{1j}/a_{kj}+b_{jk}$, $(b_{1j}-b_{kj})a_{jk}$, $-b_{j1}/a_{j1}/a_{kj}+b_{jk}$, $(-b_{j1}/a_{j1}-b_{kj})a_{jk}$, $-b_{j1}a_{1j}a_{jk}+b_{jk}$, $(-b_{j1}a_{1j}-b_{kj})/a_{kj}$, $-b_{j1}a_{1j}/a_{kj}+b_{jk}$, and $(-b_{j1}a_{1j}-b_{kj})a_{jk}$.

The final result of these calculations are the scale factors A_k and B_k and their errors. Figure 14 shows the output form **MSCIMATCH** with the scale factors derived. The advantage of this algorithm is that it provides a more robust estimate of the scalings along with reasonable errors including variations between all the pairs of images. The disadvantages are a more complex calculation and, due to combinatorics, may become prohibitive beyond some number of input images.

```
mos039S: 1.0000 (0.0000)     0.00 (0.00)
mos056S: 0.8621 (0.0049)  5061.53 (0.00)
mos057S: 0.8636 (0.0063)  4780.30 (0.00)
mos058S: 0.8581 (0.0046)  4727.07 (0.00)
mos059S: 0.8615 (0.0055)  4818.74 (0.00)
mos060S: 0.8523 (0.0055)  5263.10 (0.00)
mos061S: 0.8613 (0.0053)  6308.23 (0.00)
mos062S: 0.8619 (0.0057)  6882.94 (0.00)
mos063S: 0.8482 (0.0052)  7224.76 (0.00)
mos064S: 0.8496 (0.0052)  7099.57 (0.00)
mos065S: 0.8497 (0.0068)  6668.69 (0.00)
mos066S: 0.8536 (0.0049)  6294.45 (0.00)
mos067S: 0.8562 (0.0069)  5665.67 (0.00)
mos068S: 0.8495 (0.0067)  3998.97 (0.00)
```

Figure 14: Example of the output from **MSCIMATCH**. The columns are image name, A_k, estimated one sigma error, B_k, and estimated one sigma error. In this example the B_k are fixed on input by the previously measured mean sky and so the errors are zero. The factors are normalized to the first input image. In addition to the terminal output the A_k and B_k are written to the image headers under the keywords MSCSCALE and MSCZERO respectively. There is a sharp drop in the transparency and increase in sky brightness because the first image chosen as the reference was taken significantly earlier in the night.

The relative scale factors are written to each image in the keywords MSCZERO and MSCSCALE in the way used by the combining task **MSCSTACK** (§9.3). Note that MSCZERO will be derived from the SKYMEAN keyword when the intensity matching is constrained to determining only the intensity scaling.

9.3 Putting Images Together: MSCSTACK

The task **MSCSTACK** combines images by registering them spatially and photometrically, rejecting pixels based on masks, data thresholds, and rejection algorithms, and then averaging or combining the remaining overlapping pixels. In addition to the combined final image, the task can produce pixel masks giving for each output pixel whether there is any unrejected data, which input pixels were excluded, the number of pixels rejected, the total effective exposure time, and the standard deviation of the combined pixels. The task is a script based on the general **COMBINE** task (§3.4).

MSCSTACK computes integer offsets between the set of input images based on the WCS. When the images are produced by **MSCIMAGE** using a common reference WCS this will register the images as precisely as the astrometric calibration allows. The size of the output image is determined by the minimum size that includes all the input data.

The input images are adjusted to a common photometric scale by a combination of additive offsets and multiplicative scalings. These may be determined by the program but the recommended scaling method is to use the scale factors determined by **MSCSKYSUB** (§9.1) and **MSCIMATCH** (§9.2) and recorded in the keywords MSCZERO and MSCSCALE. Once the images are scaled, overlapping pixels are combined by averaging or medianing. Pixels may be excluded from the combining in two ways. Pixels are explicitly excluded using bad pixel masks and data thresholds. Pixels may also be excluded using one of the rejection algorithms provided by the **COMBINE** task.

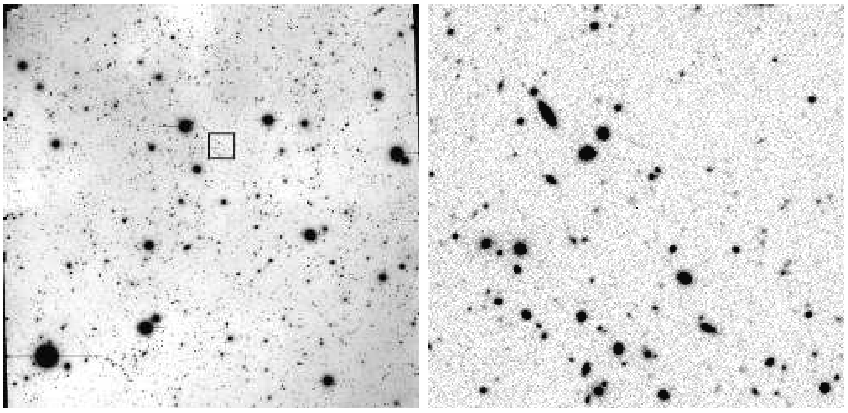

Figure 15: Final image from a stack of dithered exposures. The panel on the left is a full field (8711 x 8968 pixels = 37.5 x 38.5 arcmin) and the panel on the right is a small (512 x 512 = 2.18 x 2.18 arcmin) region at full pixel resolution. The box in the left panel indicates the region covered by the right panel. The data is the NDWFS J1426+3456 Bootes I-band field from the first survey release data (see web reference [2]).

References

[1] See http://www.noao.edu/kpno/mosaic and http://www.ctio.noao/mosaic.

[2] Jannuzi, B. T., and Dey, A., 1999, in *Photometric Redshifts and the Detection of High Redshift Galaxies*, ASP Conference Series, V191, ed. Weymann, Storrie-Lombardi, Sawicki, and Brunner, p111. (See also http://archive.noao.edu/ndwfs/)

[3] Valdes, F. and Tody, D., 1998, *The NOAO Mosaic Data Handling System*, Optical Astronomical Instrumentations, SPIE Vol. 3355, p497 (also http://iraf.noao.edu/projects/ccdmosaic/spie98.ps).

[4] See http://iraf.noao.edu/projects/ccdmosaic.

[5] Valdes, F., *Guide to the NOAO Mosaic Data Handling Software*, http://iraf.noao.edu/projects/ccdmosaic/mscguide.html.

[6] Ponz, J. D., Thompson, R. W., and Munoz, J. R., *The FITS Image Extension*, Astronomy and Astrophysics Supplement Series, V105, p.53.

[7] Zarate, N., 1996, *A FITS Image Extension Kernel for IRAF*, Astronomical Data Analysis Software and Systems V, ASP V101, ed. Jacoby and Barnes, p331. (See also http://iraf.noao.edu/iraf/docs/fits_userguide.ps.Z.)

[8] **Topics in Applied Physics: Two-Dimensional Digital Signal Processing: Transforms and Median Filters**, V43, 1981, Springer-Verlag, ed. T. S. Huang, p209.

[9] Valdes, F., 2001, *ACE: Astronomical Cataloging Environment*, Automated Data Analysis in Astronomy, ed. R. A. Gupta and H. P. Singh. (also http://iraf.noao.edu/projects/ace/ace_adaa.ps)

[10] Calabretta, M. and Greisen, E., *Representations of celestial coordinates in FITS*, in preparation, ftp://ftp.cv.nrao.edu/NRAO-Staff/egreisen/scs.ps.gz.

[11] Greisen, E. and Calabretta, M., *Representations of world coordinates in FITS*, in preparation, ftp://ftp.cv.nrao.edu/NRAO-Staff/egreisen/wcs.ps.gz).

[12] Valdes, F. G., Campusano, L. E., Velasquez, J. D., and Stetson, P. B., 1995, PASP, V107, 1119.

[13] Valdes, F., *Creating a Mosaic Coordinate System*, http://iraf.noao.edu/projects/ccdmosaic/astrometry/astrom.html.

[14] Fruchter, A. S. and Hook, R. N., 2001, submitted PASP, astro-ph/9808087.

[15] http://www.noao.edu/noao/noaodeep/ReductionOpt/frames.html.

Acknowledgements

The author would like to thank the organizers and editors, R. A. Gupta and H. P. Singh, of the *Workshop on Automated Data Analysis in Astronomy* for their encouragement and financial support to attend the workshop. I am also grateful for the incentive provided by the meeting and the publication to prepare this paper as an reference for mosaic users. I also acknowledge the help provided by Carolyn Valdes in completing this paper.

The **MSCRED** package was developed by the author. It is built on various components, libraries, and tools developed by various members of the NOAO IRAF Group. The software developed by Lindsey Davis is of special importance in the data reduction tasks described in this paper.