ANNEX B

Papers on filled array observing strategies and calibration

Draft: September 27, 1999

Dithering Strategies for Efficient Self-Calibration of Imaging Arrays

Richard G. Arendt[1], D. J. Fixsen[1], & S. Harvey Moseley
Laboratory for Astronomy and Solar Physics,
Code 685, NASA GSFC, Greenbelt, MD 20771;
arendt, fixsen, moseley@stars.gsfc.nasa.gov

ABSTRACT

With high sensitivity imaging arrays, accurate calibration is essential to achieve the limits of detection of space observatories. One can simultaneously extract information about the scene being observed and the calibration properties of the detector and imaging system from redundant dithered images of a scene. There are large differences in the efficiencies of scanning and dithering strategies for allowing the separation of detector properties from sky brightness measurements. In this paper, we quantify these differences by developing a figure of merit (FOM) for the scan and dither procedure. Patterns based on the antenna arrangements of radio interferometers with good $u - v$ plane coverage, are found to have good figures of merit. We present patterns for both deep surveys of limited sky areas and for shallow surveys. We show that the effectiveness of any analysis procedure is determined by the observing strategy. By choosing a strategy which encodes calibration in the observations in an easily extractable way, we enhance our ability to calibrate our detector systems and to reach the ultimate limits of sensitivity which are required to achieve the promise of many missions.

1. Introduction

In order to achieve their required performance, many observing systems must observe with sensitivities very near their background limits. In general, the total noise of their measurements will be determined by both the statistical fluctuations of the photon flux and uncertainties in detector gain and offset. Any successful calibration procedure must determine these detector parameters sufficiently accurately so that their uncertainties make small contributions to the measurements errors compared to those of the background fluctuations. If the science done with the instrument requires substantial spatial or temporal modeling, calibration requirements become more demanding, ultimately requiring similar integration time for observation and calibration as in the case of the *COBE* FIRAS instrument (Mather, et al. 1994; Fixsen, et al. 1994). Additionally, in such cases robust error estimators are often needed. A common method to determine the instrument calibration is to look at known calibration scenes of different brightnesses to deduce gain and offset of each detector pixel. This requires a well characterized calibration source and often a change in instrument mode to carry out the measurement. This procedure may introduce systematic errors relating to the extrapolations from the time of the calibration observations to the time of the sky observations and from the intensity (and assumed flatness) of the calibration source to the intensity of the observed sky. A different approach is to use the measurements of the sky alone to extract the

[1]Raytheon ITSS

calibration data for the system. A technique for analyzing sky data to obtain the detector calibration, the sky intensities, and the full covariance matrix has been described by Fixsen, Moseley, & Arendt (1999). By using the sky observations for calibration, the systematic errors introduced by applying a calibration derived from a distinctly different data set are eliminated. The method requires a set of dithered images, where a single sky location is imaged on many different detector pixels. In this paper, we examine different observing strategies (i.e. dither patterns) that allow this self-calibration technique to be applied successfully. With a good observing strategy, this calibration process can enable us to reach background limits in surveys using calibration data from the survey itself, avoiding the operational overhead and systematic errors associated with a separate calibration process.

The remainder of this paper is concerned with characterizing what makes a dither pattern good for self-calibration purposes. We present a "figure of merit" (FOM) which can be used as a quantitative means of ranking the suitability of different dither patterns (§2). We then present several examples of good, fair and poor dither patterns (§3), and investigate how changes to the patterns affect their FOM. In §4, we show how dithered data can be collected in the context of both deep and shallow surveys. We also investigate the combined effects of dithering and the survey grid geometry on the completeness of coverage provided by the survey. Section 5 discusses miscellaneous details of the application and implementation of dithering. Section 6 summarizes the results.

2. Evaluation of Dithering Strategies

2.1. Dithering

We define the process of "dithering" as obtaining multiple mostly overlapping images of a single field. Normally, each of the dithered images has a different spatial offset from the center of the field, and none of the offsets of the dither pattern is larger than about half of the size of the detector array. Generally, the set of dithered images is averaged in some manner into a single high-quality image for scientific analysis. This is distinct from the processes of "surveying" or "mapping", in which a field much larger than the size of the array is observed, using images that are only partially overlapping. If survey data is combined into a single image for analysis, then the process required is one of mosaicking more than averaging. A region may be surveyed or mapped using dithered images at each of the survey grid points.

There are several reasons why an observer might wish to collect dithered data. The first is simply to make sure that no point in the field remains unobserved because it happened to be targeted by a defective pixel in the detector array. To meet this objective, two dither images would suffice, provided their offsets are selected to prevent two different bad pixels from targeting the same sky location. A second reason to dither is so that point sources sample many different subpixel locations or phases. Such a data set allows recovery of higher resolution in the event that the detector pixel scale undersamples the instrumental point spread function. Several procedures have been developed for this type of analysis, which is commonly applied to HST imaging data and 2MASS data (e.g. Fruchter & Hook 1999; Williams, et al. 1996; Lauer 1999; Cutri, et al. 1999). The third reason to dither is to obtain a data set which contains sufficient information to derive the detector calibration and the sky intensities from the dithered data alone. A procedure to carry out a self-calibration on dithered data has been described by Fixsen, et al. (1999). The means for obtaining data to facilitate this process is the main focus of the present paper.

Dithering involves repointing the telescope or instrument, and thus may require additional time compared to simply taking multiple exposures of the same field. Multiple exposures of the same field

without dithering would allow rejection of data affected by transient effects (e.g. cosmic rays), and improved sensitivity through averaging exposures, but of course lack the benefits described above. Whether the time gained by not dithering outweighs the benefits lost, will depend on the instrument and the observer's scientific goals.

2.2. A Figure of Merit

The goal of this figure of merit is to compare different dither patterns rather than investigating all of the details of a full observing system. The details matter of course and they may be important in choosing the dither pattern. Here we make several simplifying assumptions to ease the calculations and comparisons. First we assume that all of the detector elements (pixels) have the same noise and gain. Next we assume that the noise is independent of sky position, either because the Poisson counting statistics are not important or the observed field is so uniform that the photon counting statistics do not vary appreciably across the field.

With these assumptions we can simultaneously solve for both the gain and/or offset for each detector element and the sky brightness of each sky pixel (Fixsen, et al. 1999). The solution necessarily introduces correlations into the uncertainties. In general these correlations degrade the signal quality although they can improve the results of some types of measurements depending on whether the correlations are positive or negative and whether the two data elements are used with the same or opposite sign in the measurement.

For the IRAC, there are 256×256 detector elements in the detector arrays we model, hence there are $256^4 = 4$ billion correlations to look at. For the figure of merit we choose only a single pixel (128,128) and look at its correlations. This is done to reduce the calculational burden. Since all of the pixels are locked to the same dither pattern the correlations are similar for the other pixels (discussed below).

Because the correlations generally have negative effects and those effects are larger for larger correlations, we sum the absolute value of the correlation over all of the other pixels. This is compared with the covariance of the central pixel in the idealized case where a completely known field is observed, as this is the irreducible uncertainty due to detector noise alone. Thus, we define the figure of merit (FOM) as:

$$FOM = \frac{\mathcal{V}_{p_0 p_0}}{\sum_{i \in \text{ all pixels}} |V_{i p_0}|} \tag{1}$$

where V is the covariance matrix of the detector, $\mathcal{V}$ is the covariance matrix of the detector in the idealized case ($\mathcal{V}_{ij} = 0$ for $i \neq j$), and p_0 is the selected (central) pixel of the detector. The FOM is bounded on the range [0,1], and can be thought of as an efficiency of encoding correlations in the dither pattern. In this form it may also be used as a weighting factor.

Equation 1 is not the only possible way of defining a FOM. For example, an alternate definition of the FOM could use $(\mathcal{V}^2_{p_0 p_0} / \sum V^2_{i p_0})^{1/2}$ instead of $\mathcal{V}_{p_0 p_0} / \sum |V_{i p_0}|$ in Equation 1. However this alternate form is found to result in a more compressed range of numerical values which makes it less convenient for distinguishing the quality of various dither patterns.

Throughout this paper, we calculate the FOM based on calibration which only seeks to determine the detector gains or offsets, but not both. When both gains and offsets are sought, the solution for the covariance matrix will contain degeneracies that are only broken by the presence of a non-uniform sky brightness (Fixsen, et al. 1999). The FOM when solving for one detector parameter should generally be similar to that which would apply when solving for both gains and offsets.

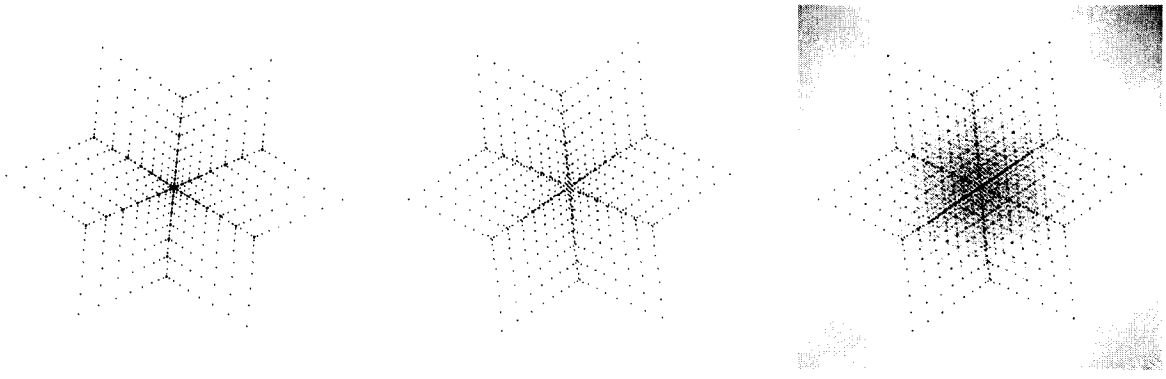


Fig. 1.— On the left is the $u-v$ baseline coverage of the VLA for a snapshot of a source at the zenith. In the center and on the right is the map of $|V_{ip_0}|$ for a "VLA" dither pattern, stretched to emphasize the similarity to the VLA $u-v$ plane coverage, and the weaker correlations respectively.

In order to compute relative gain and/or offset, two detector elements must observe the same sky pixel or have a connection through other detector elements that mutually observe one or more sky pixels. A shorter path of intermediate detectors implies a tighter connection and lower uncertainties. One goal of dithering is to tighten the connections between detectors and thus lower the uncertainties. This is closely related to another problem that has been dealt with previously, namely covering the $u-v$ plane with a limited number of antennas in a radio interferometer. The important difference is that with radio telescopes only pairs of antennas (although all pairs) can be used to generate interference patterns, whereas here a path involving several intermediate detector elements can be used to generate an indirect correlation. However, the greater the number of intermediate steps that must be used to establish a correlation, the noisier it will be. In general therefore, we should expect patterns that work well for interferometers will also work well as dither patterns.

Figure 1 shows the $u-v$ coverage of the VLA for a snapshot of a source at the zenith. Each antenna pair leads to a single sample marked with a dot in the $u-v$ plane. Also shown is the map of $|V_{ip_0}|$ generated by using a 27-position dither pattern with the same geometry as the VLA array (§3.2). The strongest correlations are found at locations corresponding to the VLA baselines. However, the non-zero correlations (and anti-correlations) found elsewhere in the map make a significant contribution to the total FOM.

Figure 2 shows maps of $|V_{ip_0}|$ generated using different choices of p_0. These maps illustrate that the correlations of all pixels are similar in structure to those of the central pixel, but the finite size of the detector limits the correlations available to pixels near the detector edges. The dither pattern used in this demonstration is the VLA pattern described in §3.2.

Despite the similar geometries of radio interferometer $u-v$ coverage and dither pattern maps of $|V_{ip_0}|$, several important differences should be noted. First, the $u-v$ coverage is derived instantly. Observing over a period of time fills in more of the $u-v$ plane as the earth's rotation changes the interferometer baselines relative to the target source. In contrast, the $|V_{ip_0}|$ coverage shown in Figs. 1 and 2 is only achieved after collecting dozens of dithered images. To fill in additional coverage, the dither pattern must be altered directly because there is no equivalent of the earth rotation that alters the geometry of the instrument with respect to the sky. The second important difference is that the short interferometer baselines (found near the center of the $u-v$ plane) will be sensitive to the large-scale emission. For dither patterns the inverse relation holds. Direct correlations between nearby detector pixels are sensitive to small-scale structure in

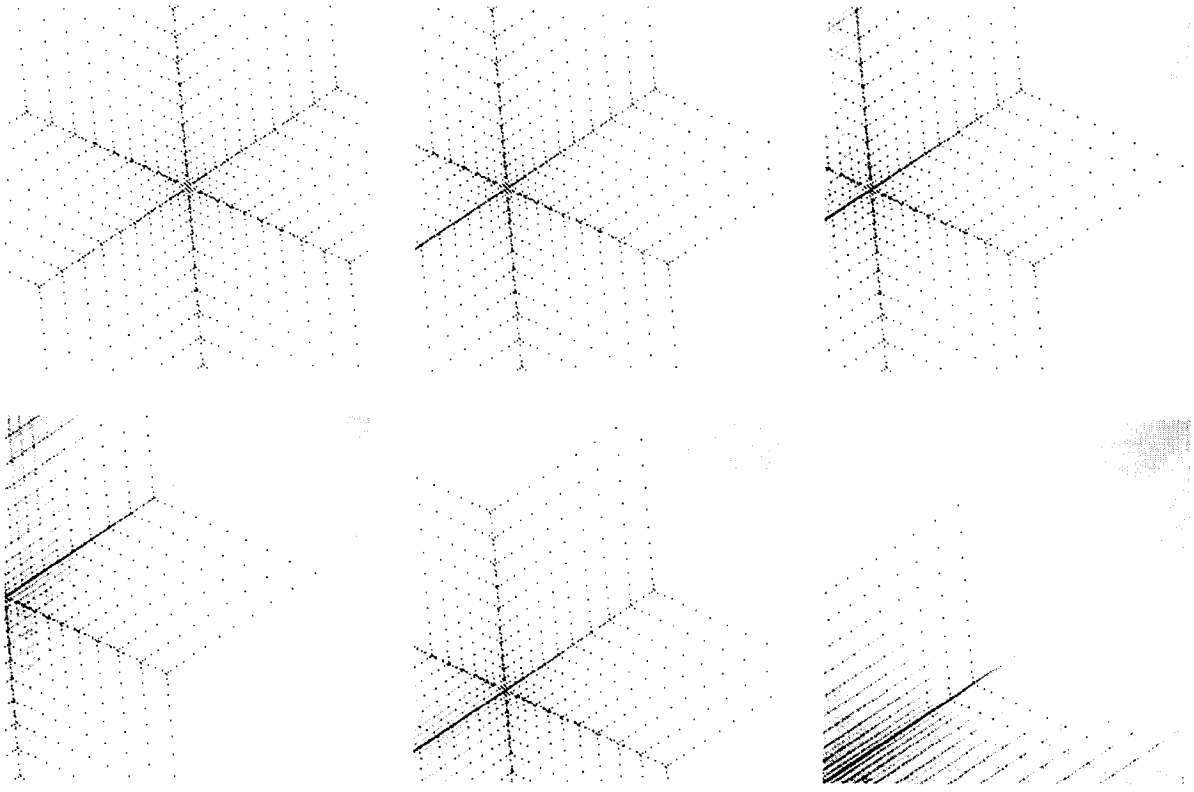


Fig. 2.— The panels show the $|V_{ip_0}|$ correlations for detectors at the locations (128,128), (64,128), (32,128), (0,128), (64,64), and (0,0) in a 256×256 array (left to right and top to bottom). Dark spots represent strong correlations. The dither pattern used to calculate these correlations is a 27-point VLA pattern.

the detector properties and sky intensities. Thus the outer edge of the interferometer's $u - v$ coverage represents a limit on the smallest-scale structure that can be resolved, while the outer edge of strong $|V_{ip_0}|$ correlations represents a limit on the largest-scale variations that can be reliably distinguished.

3. Various Dither Patterns

Several general algorithms for generating dither patterns have been examined. In many cases, we have also explored variants of the basic algorithms by changing functional forms, adding random perturbations, or applying overall scale factors. We have also tested several specific examples of dither patterns from various sources. Examples of the patterns described below are shown in Figure 3. All tests reported here assumed detector dimensions of 256×256 pixels unless otherwise noted.

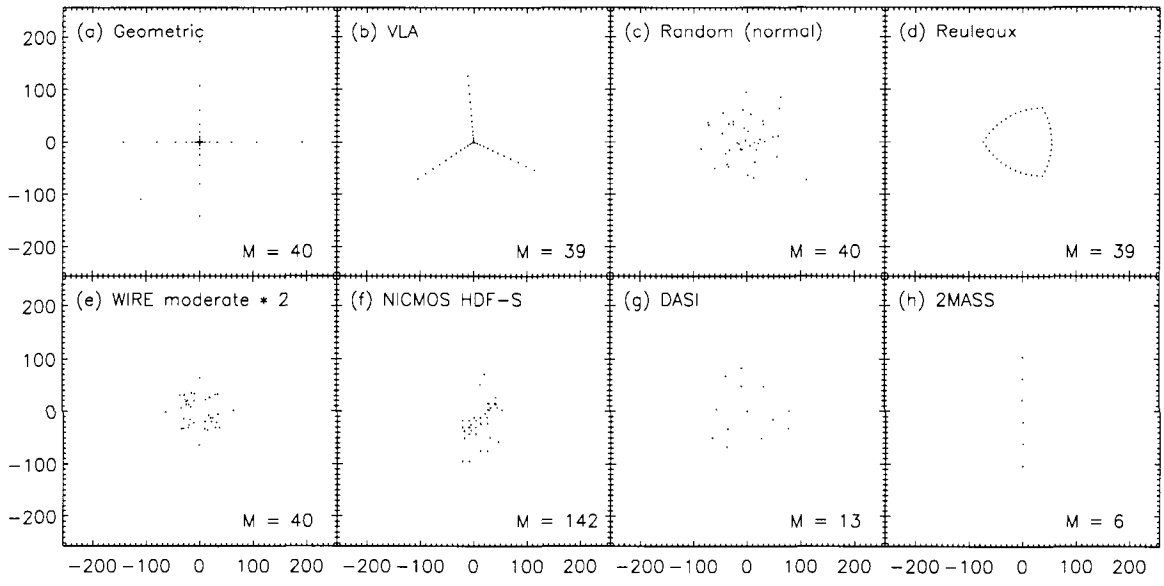


Fig. 3.— Examples of some of the tested dither patterns. The dots mark the center of the array for each of the M positions for each pattern.

3.1. Reuleaux Triangle

Take an equilateral triangle and draw three 60° arcs connecting each pair of vertexes, while centered on the opposite vertex. The resulting fat triangle is a Reuleaux triangle. This basic shape has been used to set the geometry of the Sub-Millimeter Array (SMA) on Mauna Kea (Keto 1997).

This shape can be used as a dither pattern by taking equally spaced steps along each side of the Reuleaux triangle. The length of the steps is set by the overall size of the triangle (a free parameter) and the number of frames to be used in the pattern. For an interferometer, Keto (1997) shows that the $u - v$ coverage can be improved by displacing the antennas from their equally spaced positions around the triangle.

3.2. VLA

The "Y"-shaped array configurations of the Very Large Array (VLA) radio interferometer are designed such that the antenna positions from the center of the array are proportional to $i^{1.716}$ (Thompson, et al. 1980). The three arms of the array are separated from each other by $\sim 120°$. We have adopted this geometry to provide a dither pattern with positions chosen along each of the three arms at

$$dr = \sqrt{dx^2 + dy^2} = i^p \quad \text{where} \quad i = 1, 2, 3, ..., M/3. \tag{2}$$

and p is an arbitrary power which can be used to scale the overall size of the pattern. The first step along each of the 3 arms is always at $dr = 1.0$. The azimuths of the arms were chosen to match those of the VLA, at 355°, 115°, and 236°.

3.3. Random

Random dither patterns were tested using dx and dy steps generated independently from normal (Gaussian) or from uniform (flat) distributions. The widths of the normal distribution or the symmetric minimum and maximum of the uniform distribution are free parameters.

3.4. Geometric Progression

We have generated a geometric progression pattern, stepping in x in steps of $(-f)^n$, where $n = 0, 1, ...N-1$ and $f^N = 256$. The same steps are also used in the y direction. This pattern separates the x and y dimensions. In each dimension the pattern is quite economical in generating correlations up to the point where $f = 2$. Beyond this there is little to be gained in adding more dither steps in the x or y direction. However, there is some benefit expected in adding steps combining x and y offsets. Hence, for a 256×256 array, we should expect the geometric pattern to be good for $M \leq 2\ log_2(256) = 16$ positions and not show much improvement by adding more positions.

The geometric progression patterns used here contain two additional steps chosen at $(dx, dy) = (0, 0)$ and at a position such that $\sum dx = \sum dy = 0.0$. This is a cross-shaped pattern, with one diagonal pointing, from which any desired pixel-to-pixel correlation can be made with a small number of intermediate steps. The alternating sign of the steps builds up longer separations quickly.

3.5. Other Patterns

Several other patterns were also tested with little or no modifications. The patterns that were planned for the WIRE moderate and deep surveys were examined with both the nominal dither steps, and with steps scaled by a factor of 2 to account for the difference between the 128×128 pixel WIRE detectors and a larger 256×256 pixel detector. The pattern used for NICMOS observations of the HDF-S was tested. The configuration of the 13 antennas of the Degree Angular Scale Interferometry (DASI; Halverson, et al. 1998) was used as a scalable pattern. The declination scanning employed by 2MASS yields a linear dither pattern.

3.6. Figures of Merit for the Patterns

In the simplest form, a specific pattern, M images deep, would be used to collect data at a single target. The FOM for all patterns tested, with various M and other modifications, are listed in Table 1. For all patterns, the FOM increases (improves) as M increases. For $M < 20$ the change is quite rapid. The variations of FOM as a function of M for the tabulated versions of each of the patterns are shown in Figure 4.

Table 1 also lists results for a Reuleaux triangle pattern applied to a 32×32 detector, and for two large grid dither patterns applied to the same array. The grid dither patterns are square grids with 1 pixel spacings between dithers, such that for the $M = 1024$ pattern a single sky pixel is observed with each detector, and for the $M = 4096$ pattern a 32×32 pixel region of sky is observed with each detector pixel. These results demonstrate that in the extreme limit where all correlations are directly measured, the $FOM \rightarrow 1.0$. The FOM does not reach 1.0 because of the finite detector and dither pattern sizes.

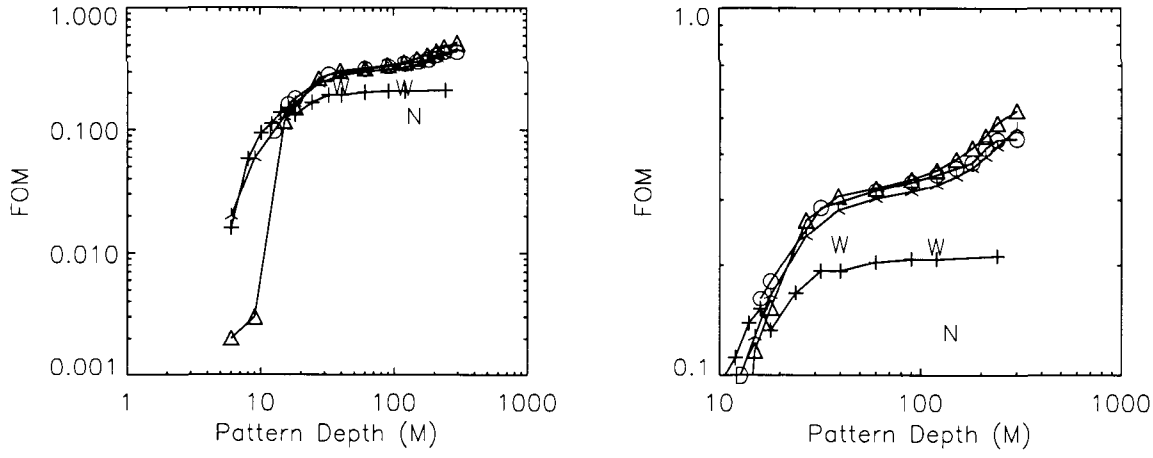


Fig. 4.— The FOM as a function M the number of positions in each pattern. "+" = geometric progression, inverted "Y" = VLA, "○" = random (normal), "△" = Reuleaux triangle, "W" = WIRE moderate and deep surveys, "D" = DASI, and "N" = NICMOS coverage of the HDF-S. The right panel shows the same data on an enlarged scale.

For the 256×256 arrays, the Reuleaux and random (normal) patterns have the best FOM for $M > 20$. The VLA pattern is only a little worse, but other patterns have distinctly smaller FOM than these patterns. For the scalable VLA, random, Reuleaux, and DASI patterns, the best FOM for a fixed M usually occurs when the maximum $|dx|$ or $|dy| \approx 128$ pixels. For patterns with small M the optimum scale factor is usually smaller, to avoid too many large spacings between widely scattered dither positions. For values of $M < 20$ no pattern seems to produce a good FOM, however, the geometric pattern usually does best in this regime. Rotating the patterns with respect to the detector array generally produces only modest changes in the FOM. For $M \lesssim 30$, the FOM of a Reuleaux pattern is improved by adding small random perturbations to the dither positions. No optimization of the perturbations was performed (as Keto 1997), but apparently any perturbation is better than none for small M patterns. Deep Reuleaux triangle patterns are neither improved nor worsened by small perturbations.

The results presented in Fig. 4 and Table 1 indicate that a good FOM is dependent on patterns that sample a large number and wide range of spatial scales. A variety of patterns with different geometries can yield satisfactory results, as demonstrated by the rather different Reuleaux triangle and random patterns. Therefore, attempts to find the single "optimum" pattern may not be very useful, and selection of a dither pattern needs to carefully avoid patterns that contain obvious or hidden redundancies that lead to a poor FOM. An example of this sort of pitfall is the $M = 18$ geometric pattern, for which all dither steps are integer powers of 2, leading to a FOM that is worse than geometric patterns with depths of $M = 14$ or 16.

The coverage of the VLA, random, and Reuleaux triangle dither patterns when used for observation of a single target is shown as maps in Figure 5, and histograms in Figure 6. The Reuleaux triangle dither pattern provides the largest region covered at maximum depth, but if a depth less than the maximum is still useful then the VLA dither pattern may provide the largest area covered.

The importance of the largest dither steps in a pattern is demonstrated through analysis of simulated

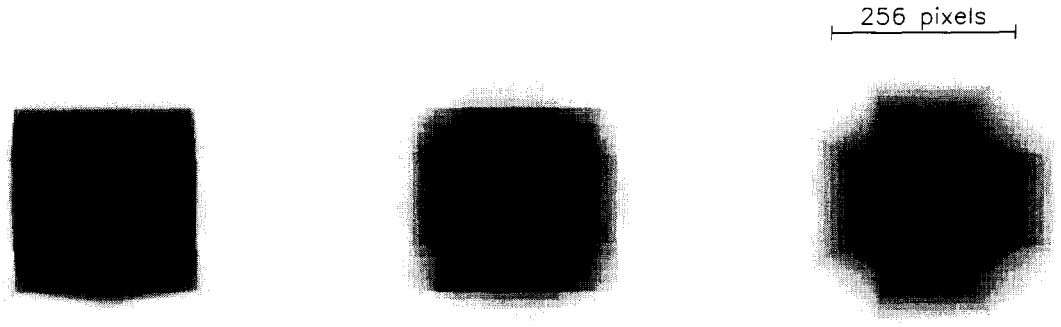


Fig. 5.— Coverage maps for $M = 39$ single target dither patterns (left) VLA: $FOM = 0.282$, (center) Random Gaussian: $FOM = 0.302$, (right) Reuleaux triangle: $FOM = 0.307$.

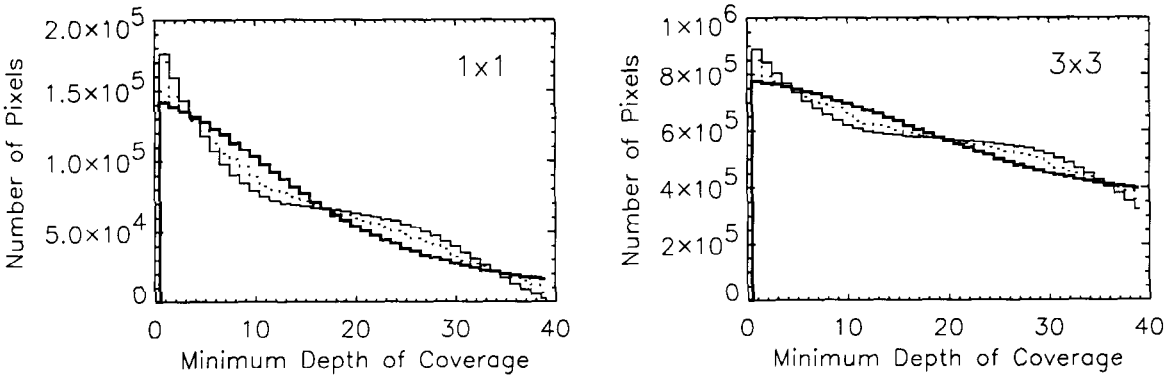


Fig. 6.— Cumulative histograms of the coverage as a function of minimum depth for $M = 39$ VLA (thin), random Gaussian (dotted), and Reuleaux triangle (thick) dither patterns. Coverage for a single target is shown at left; coverage for a deep 3×3 survey with a 256×256 pixel grid spacing is shown at right.

WIRE data. A synthetic sky was sampled using both geometric progression and random dither patterns. The maximum dither offset was 38 pixels for the geometric progression pattern and 17 pixels for the random pattern. The FOM for this geometric pattern is 0.127, and for this random pattern it is 0.099. WIRE's detectors were 128 arrays. The gain response map used in the simulations contained large scale gradients with amplitudes of $\sim 10\%$. Figure 7 shows comparisons between the actual gains and the gains derived when the self-calibration procedure described by Fixsen, et al. (1999) is employed. The random dither pattern without the larger dither offsets was less effective at identifying the large scale gain gradient. The undetected structure in the gain winds up appearing as a sky gradient that affects the photometry of both the point sources and the background in the images.

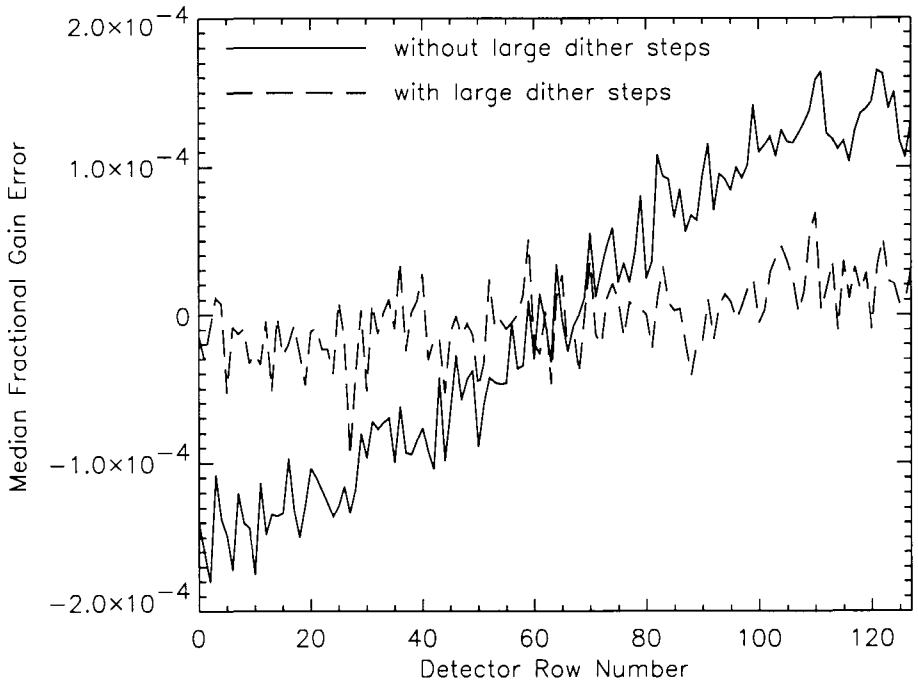


Fig. 7.— The median fractional gain errors are plotted as a function of detector row for detector gains derived from two simulated WIRE data sets. Each simulation contains 10 dithered images. Only one simulation includes relatively large dither steps. When applying a self-calibration algorithm, a lack of large dither steps leads to large-scale gain errors.

4. Surveys

4.1. Deep Surveys

For obtaining a standard deep survey, we have assumed that the same dither pattern is repeated at each location of a grid. The survey grid is assumed to be aligned with the detector array and square, with a spacing no larger than the size of the array. The FOM for surveys using several different dither patterns and grid spacings are listed in Table 2. The FOM derived for the entire survey as a single data set is basically determined by the FOM of the dither pattern used. The overlap between dithers from adjacent points in the survey grid, effectively adds additional steps to the dither pattern, which slightly improves the FOM over that of the pattern when used for a single target. Smaller survey grid spacings lead to increased overlap and increased FOM, but also lead to a smaller area of sky covered in a fixed number of frames. The improvement in the FOM when used in surveys rather than singly is most significant for relatively shallow dither patterns, however, even in a survey, the FOM of a shallow pattern is still not very good. The FOM improves only slightly as the survey grid grows larger than the basic 2×2 unit cell.

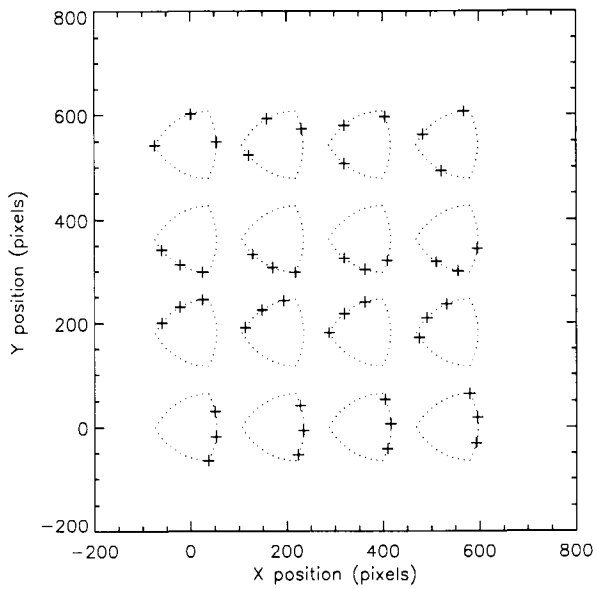


Fig. 8.— An example of a 4×4 $M = 3$ shallow survey on a 181×181 pixel grid using an $M = 33$ Reuleaux triangle dither pattern. The dots show the repetition of the full dither pattern, while the crosses mark the dither points that were actually used at each survey grid point.

4.2. Shallow Surveys

For shallow surveys in which as few as 2 images per grid location are desired, using the same small M dither pattern at each location yields a very poor FOM. An alternate method of performing a shallow survey is to choose a larger M dither pattern and apply successive steps of the dither pattern at successive locations in the survey grid (Figure 8). If the survey is large enough, it can contain all the direct correlations of the large M dither pattern, though spread out among many survey grid points rather than at a single location. The FOM of the shallow survey can thus approach the FOM of the single deeper dither pattern. The advantage of altering the dither pattern at each survey grid point is still present, though less significant, as the survey depth increases. The FOM derived from various surveys using this shallow survey strategy are shown in Table 3.

A random dither pattern is a natural choice for use in this shallow survey strategy. One can proceed by simply generating a new random set of dithers at each survey grid point. If a more structured dither pattern is used as the basis for the shallow survey (e.g. the Reuleaux triangle in Fig. 8), then one must address the combinatorial problem of selecting the appropriate subsets of the larger dither pattern at each survey grid point. The example shown in Fig. 8 is *not* an optimized solution to the combinatorial problem.

4.3. Survey Coverage & Grids

When a large area is to be observed, the most efficient way to cover the region is to use a square survey grid aligned with the detector array and with a grid spacing equal to the size of the array, or slightly less to

guard against bad edges or pointing errors. In this mode a deep survey using the same M position dither pattern at each survey grid point will cover the desired region at a depth of M or greater. There will be no holes in the coverage, though the edges of the surveyed region will fade from coverage of M to 0 with a profile determined by the dither pattern used (Fig. 6). A shallow survey, using a different dither pattern at each grid point, may or may not have coverage holes depending on the maximum size of the dither steps and the grid spacing of the survey. The constraint for avoiding coverage holes is that the overlap of the survey grid must be more than the maximum range of dither step offsets (independently in the x and y coordinates), e.g.

$$X - \Delta X > \max(dx_i) - \min(dx_i) \tag{3}$$

where X is the size of the array, ΔX is the survey grid spacing, and dx_i are the dither steps ($i = 1...M$). This constraint places the survey grid points close enough together that coverage holes are avoided even if dithers at adjacent grid point are offset in the maximum possible opposite directions. If the shallow survey observing program can be arranged to avoid this worst case, then the grid spacing may be increased without developing coverage holes. Coverage holes may be undesirable when mapping an extended object, but may be irrelevant if one is simply seeking a random selection of point sources to count. Note that some minor coverage holes are inevitable, where data are lost to bad pixels or cosmic rays. Additionally, a coverage hole where a depth of $M = 1$ is achieved instead of $M = 3$ might be more serious than one where $M = 18$ is achieved instead of $M = 20$.

For this shallow survey strategy there is an inherent tradeoff between the area covered (without holes) and the FOM. Using a dither pattern containing large dither steps as the basis for the survey will lead to a good FOM, but require a relatively large overlap in the survey grid spacing and a consequent loss of area covered by the survey. Decreasing the scale of the dither pattern leads to a lower FOM, but permits an increase in the survey grid spacing and total area covered. The ideal balance between these will depend on the instrumental characteristics and the scientific objectives.

In many instances, an observer may want to survey or map a region of fixed celestial coordinates. In some cases, instrumental constraints (i.e. the ability to rotate the telescope or detector array relative to the optical boresight) may not allow alignment between the detector array and the desired survey grid. This will result in coverage holes in the surveyed region, unless the grid spacing is reduced enough to prevent holes regardless of the array orientation. If a square grid with a spacing of $\Delta X = X/\sqrt{2}$ is used then coverage holes are prevented for any possible orientation of the arrays. This is illustrated by plots in the first two rows of Figure 9, which shows the array positions for 4×4 $M = 1$ survey (without dithering). With a deep survey strategy, avoidance of holes in the $M = 1$ case will prevent holes at any depth M, but for the shallow survey strategy additional overlap may need to be built into the survey grid to prevent holes as discussed above. Decreasing the survey grid by a factor of $\sqrt{2}$ in each dimension results in a grid that covers only half the area that could be covered if the detectors and grid are aligned. This efficiency can be increased if the survey is set up on a triangular grid rather than a square grid. If alternate rows of the survey grid are staggered by $X/2$ (middle row of Fig. 9) and the vertical spacing of the grid is reduced by a factor of $\sqrt{3}/2$, then holes are prevented as long as the array orientation remains fixed throughout the survey (4th row of Fig. 9). The area covered by this triangular grid will be $\sim$ 87% of the maximum possible area, rather than 50% for the square grid required to prevent holes. If the array orientation is not fixed throughout the survey (last column of Fig. 9) then the triangular grid must be reduced by an additional factor of $\sqrt{3}/2$ in both dimensions. This results in a $\sim$ 65% efficiency for the triangular grid versus 50% for the square grid, which requires no further reduction. The FOM of a survey on a triangular grid is similar to that of a survey on a square grid with an equivalent amount of overlap.

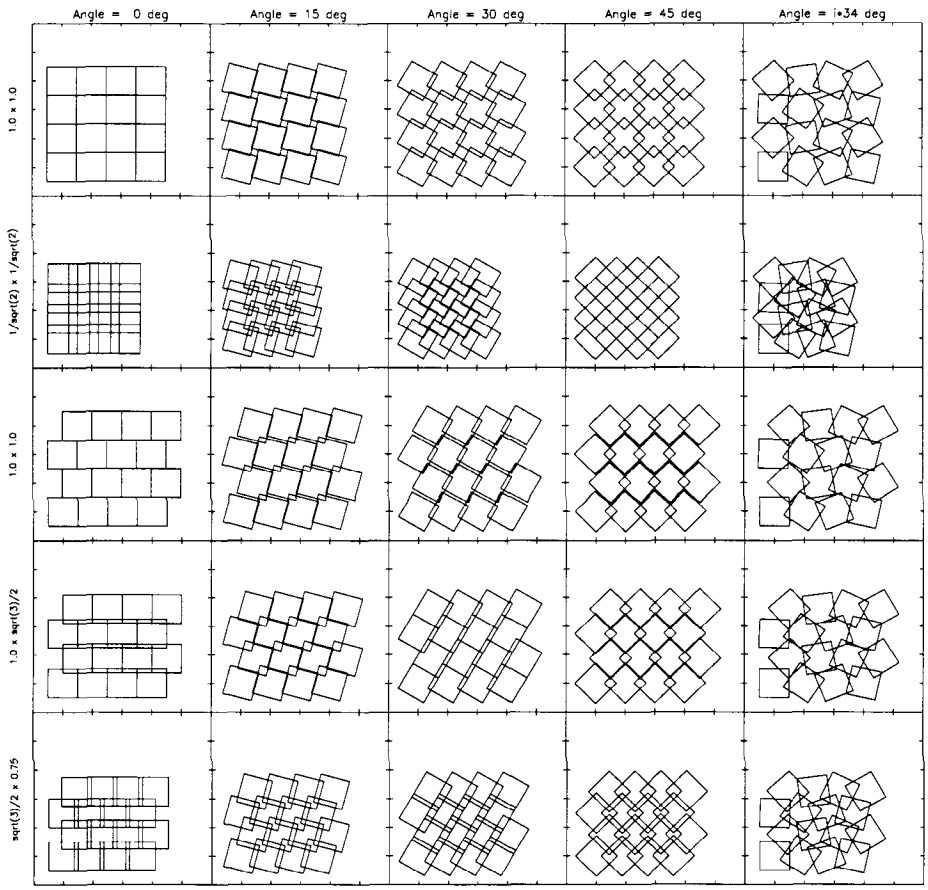


Fig. 9.— Examples of 4×4 $M = 1$ surveys on square (1st and 2nd rows), staggered (middle row), and triangular grids (4th and 5th rows) for various angles between the detector array and the grid orientation. In the last column the array orientation was rotated by 34° at each successive survey grid point. The squares indicate the outline of the entire array as pointed at each survey grid point.

5. Other Miscellaneous Details

The most flexible implementation of the dithering strategies presented here would be to have the dither steps be determined algorithmically from a small set of user-supplied parameters. For example, an observer could select: a type of dither pattern (e.g. Reuleaux triangle or random), a pattern depth $M_{pattern}$, and a scaling factor to control the overall size of the pattern. From this information, the telescope control software could calculate and execute the desired dither pattern. For the shallow survey strategy presented above, the observer would also need to supply: the survey depth, $M_{survey} < M_{pattern}$, and perhaps an index to track which grid point of the survey is being considered (software might handle this automatically).

Sometimes design or operational constraints require that the dither patterns reside in a set of pre-calculated look-up tables. In this case (which has applied to both WIRE and IRAC) the observer's ability to set the dither pattern is more limited. However, some of the limitations of using dither tables can

be mitigated if the observer is not forced to use dither steps from the tables in a strictly sequential fashion. For example, one dither table might contain an $M = 72$ Reuleaux triangle dither pattern calculated on a scale to produce the optimum FOM. If the observer is allowed to set the increment, Δi, used in stepping through this dither table, then by selecting $\Delta i = 3$ or $\Delta i = 4$, then dither patterns of $M = 24$ or $M = 18$ can be generated. Allowing non-integer increments would enable the selection of a dither pattern of any depth $M \leq 72$. This adjustment of the increment is most clearly useful for very symmetric dither patterns such as the Reuleaux triangle pattern. For a dither table containing a random pattern, non-sequential access to the table can have other uses. First, in applying the shallow survey strategy, a random dither table of length $M_{pattern}$ could be used to sequentially generate $M_{survey} < M_{pattern}$ dithers at each successive survey grid point. Selection of dither steps would wrap around to the beginning of the table once the end of the table is reached. For example a dither table of $M_{pattern} = 100$ could be used sequentially to generate 20 different patterns for an $M = 5$ shallow survey. Even better would be to have a table with $M_{pattern}$ a prime number, e.g. 101. Then, wrapping the table allows the sequential generation of $M_{pattern}$ different dither patterns for any M_{survey}, though some of these dither patterns will differ from others by only one step. Additional random patterns can be generated by setting different increments for stepping through the table. Enabling specification of the starting point in the dither table would additionally allow the observer to pick up the random dither pattern sequence at various (or the same) positions as desired. These capabilities would enable an observer to exploit the large number of *combinations* of dither steps available in a finite length dither table, in efforts to maximize the FOM. Use of a fixed dither table can also be made less restrictive if a scaling factor can be applied to the dither pattern size. A free scaling factor provides an additional means of adjusting the pattern size as desired to meet coverage or FOM goals.

For the cases presented in this paper, we have assumed that the orientation of the detector array remains fixed throughout the execution of the dither pattern and any larger survey (except for the last column of Fig. 9). However, rotation of the detector array relative to the dither pattern, either within a single pointing, or at different pointings in a deep survey, is an effective way of establishing combinations of direct pixel-to-pixel correlations that cannot be obtained using purely translational dither steps. Inclusion of rotation of the detector can lead to further improvements in the FOM of a given dither pattern or survey. In the extreme, a dither pattern could even be made entirely out of rotational rather than translational dither steps. However, without an orthogonal "radial" dither step, rotation alone is similar to dithering with steps in the x-direction but not the y-direction. The ability to implement rotations of the detector will be allowed or limited by the design and operating constraints of the telescope and instruments being used.

Bright sources can often saturate detectors and cause residual time-dependent variations in detector properties. For observations of a field containing a bright source, use of a random dither pattern may lead to streaking as the source is trailed back and forth across the detector array between dithers. In contrast the use of a basically hollow or circular dither pattern such as the Reuleaux triangle pattern, will only trail the source through a short well-defined pattern, which will lie toward the outer edge of the detector if the source position is centered in the dither pattern. If the pattern scale of the dither pattern is increased, the trail of the source can be pushed to or off the edges of the detector, though the FOM will suffer if the pattern scale is greatly increased. In other words, a hollow dither pattern with a large scale could be used to obtain a series of images looking around but not at a bright source.

Dithering may be performed by repointing the telescope, or by repositioning the instrument in the focal plane, for example through the use of a tilting optics as in the 2MASS (Kleinmann 1992) or *SIRTF* MIPS (Heim, et al. 1998) instruments. Calculation of the FOM of the dither pattern will be independent of the technique used. The self-calibration procedure, however, may be affected by effective instrumental

changes if it is repositioned in the focal plane. The alternative repointing of the telescope can be much more time consuming and may limit the use of large M dither patterns.

The combined use of two or more non-contiguous fields is transparent to the self-calibration procedure. If the same dither pattern is used on each of the separate fields, the resulting FOM will be the same as that for a single field. The FOM would be improved for the combined data set if the dither pattern is different for each of the subset. The FOM for data set of non-contiguous regions is thus similar to that obtained using the same dither strategy in a contiguous survey, except there is a small loss in the FOM because of the lack of overlap between adjacent regions.

Another means of minimizing coverage holes when using a shallow survey strategy is to oversample the depth of the survey. For example, performing the shallow survey at a depth of $M = 4$ when $M = 3$ is the intended goal will result in fewer holes at a depth of 3 for a fixed grid spacing, and in a better FOM for the overall survey. However, the cost in time of the additional exposures may be prohibitive.

The FOM as calculated here only depends upon the offsets of the dither pattern rounded to the nearest whole pixel. This means that any desired combination of fractional pixel offsets to facilitate subpixel image reconstruction may be added to the dither patterns without affecting the various aspects discussed in this paper. If using dither tables, one could have separate tables for the large scale and the fractional pixel dithers, with the actual dithers made by adding selected entries from the two tables. This could allow simultaneous and independent implementation of large-scale and subpixel dithering strategies. Only subpixel image reconstruction that demands exclusively small ($\sim$ 1 pixel) dithering would be incompatible with the dithering strategies presented here.

6. Conclusion

We have shown that proper selection of observing strategies can dramatically improve the quality of self-calibration of imaging detectors. We have established a figure of merit (Eq. 1) for quantitatively ranking different dither patterns, and have identified several patterns that enable good self-calibration of a detector on all spatial scales. The layouts of radio interferometers correspond to good dither patterns. Both the highly ordered Reuleaux triangle pattern and the unstructured random pattern provide good FOM with moderate or deep observations. This indicates that good patterns must sample a range of spatial scales without redundancy, and if this condition is met, then secondary characteristics of the patterns or instrument constraints may determine the actual choice of the dither pattern. Any dither pattern must contain steps as large as half the size of the detector array if large scale correlations are to be effectively encoded in the dithered data set. Deep surveys can take advantage of the use of a single good dither pattern. Shallow surveys can obtain good FOM by altering the dithers used at each of the survey grid points. Using a fixed pattern throughout a shallow survey makes it difficult or impossible to apply a self-calibration procedure to the resulting data sets. The use of triangular instead of square survey grids can be more efficient in executing complete-coverage surveys when the array orientation cannot be set to match the survey grid. Good dither patterns and survey strategies can be devised even in some seemingly restricted situations. The ultimate importance of dithering and a good FOM will depend on the nature of the instrument and the data and on the scientific goals. For many goals, obtaining a larger quantity of data may not be an adequate substitute for obtaining data with a good FOM.

We thank D. Shupe and the WIRE team for supplying simulated data using several different dither

patterns. W. Reach and members of the SSC and IRAC instrument teams were helpful in providing useful ideas and criticism throughout the development of this work.

REFERENCES

Cutri, R., Van Dyck, S. & the 2MASS Team, 1999, Explanatory Supplement to the 2MASS Spring 1999 Incremental Data Release, http://www.ipac.caltech.edu/2mass/releases/spr99/doc/explsup.html

Fixsen, D. J., et al. 1994, ApJ, 420, 457

Fixsen, D. J., Moseley, S. H., & Arendt, R. G. 1999, ApJ, submitted

Fruchter, A. S. & Hook, R. N. 1999, PASP, submitted

Halverson, N. W., Carlstrom, J. E., Dragovan, M., Holzapfel, W. L., & Kovac, J. 1998, Advanced Technology MMW, Radio, and Tetrahertz Telescopes, Proc. SPIE 3357, ed. T. G. Phillips (Bellingham, WA: SPIE), 416

Heim, G. B., et al. 1998, Space Telescopes and Instruments V, Proc. SPIE 3356, ed. P. Y. Bely, & J. B. Breckinridge, (Bellingham, WA: SPIE), 985

Kcto, E. 1997, ApJ, 475, 843

Kleinmann, S. G. 1992, Robotic Telescopes in the 1990s, ed. A. V. Filippenko, (San Francisco: ASP), 203

Lauer, T. R. 1999, PASP, 111, 227

Mather, J. C., et al. 1994, ApJ, 420, 439

Thompson, A. R., Clark, B. G., Wade, C. M., & Napier, P. J. 1980, ApJS, 44, 151

Williams, R. E., et al. 1996, AJ, 112, 1335

Table 1. Figures of Merit for Single Pointings

$M_{pattern}$	256 × 256 arrays								32 × 32 arrays	
	Reuleaux	Random	VLA	Geometric	DASI	WIRE	NICMOS HDF-S	2MASS	Reuleaux	Grid
6	0.002	...	0.020[e]	0.016	...	...	...	0.002	...	...
8	...	...	...	0.058	...	...	...	...	...	...
9	0.003[a]	...	0.059[f]	...	...	...	...	...	0.217[k]	...
10	...	...	...	0.094	...	...	...	...	...	...
12	...	...	...	0.112	...	...	...	...	...	...
13	...	...	...	...	0.100	...	...	...	...	...
14	...	...	...	0.139	...	...	...	...	...	...
15	0.117[b]	...	0.128[g]	...	...	...	...	...	...	...
16	...	0.162	...	0.152	...	...	...	...	...	...
18	0.153[c]	0.181	0.166	0.133	...	...	...	...	...	...
24	...	...	...	0.168	...	...	...	...	...	...
27	0.265	...	0.240[h]	...	...	...	...	...	...	...
32	...	0.286	...	0.193	...	...	...	...	...	...
39	0.307	...	0.282	...	...	...	...	...	...	...
40	...	...	...	0.193	...	0.228[i]	...	...	...	...
60	0.323	0.318	0.303	0.204	...	...	...	...	...	...
90	0.341	0.335	0.316	0.208	...	...	...	...	...	...
120	0.361	0.351	0.329	0.208	...	0.225[j]	...	...	...	...
142	...	...	...	...	...	...	0.131	...	...	...
150	0.387	0.365	0.347	...	...	...	...	...	...	...
180	0.416	0.378	0.366	...	...	...	...	...	...	...
210	0.448	0.415	0.392	...	...	...	...	...	...	...
240	0.485[d]	0.437	0.419	0.212	...	...	...	...	...	...
300	0.526	0.439	0.469	...	...	...	...	...	...	...
1024	...	...	...	...	...	...	...	...	...	...
4096	...	...	...	...	...	...	...	...	...	0.783
									...	0.889

Note. — Standard pattern sizes are: Reuleaux width = 128 pixel, Random 3 σ = 128 pixel, VLA r_{max} = 125.7 pixel, Geometric is scaled to the 256 pixel array size, DASI r_{max} = 82.5 pixel, WIRE medium and deep surveys scaled by a factor of 2, NICMOS HDF-S all camera 3 F110W data, 32 × 32 Reuleaux width = 12 pixel, Grid spacing = 1 pixel

[a] 0.037 if 3% random variations added

[b] 0.146 if 3% random variations added

[c] 0.171 if 3% random variations added

[d] 0.396 for width = 110 pixel, 0.513 for width = 144 pixel

[e] r_{max} = 16 pixel

[f] r_{max} = 32 pixel

[g] r_{max} = 125 pixel

[h] 0.176 for r_{max} = 16 pixel, 0.199 for r_{max} = 16 pixel

[i] 0.065 for the unscaled pattern

[j] 0.062 for the unscaled pattern

[k] 0.162 for width = 8 pixel

Table 2. Figures of Merit for Deep Surveys

Survey Size	Spacing (pixels)	Reuleaux		Random			VLA			Geometric		
		$M = 15$	39	$M = 6$	16	40	$M = 6$	15	39	$M = 6$	16	40
2 × 2	181	0.194	0.318	0.028	0.209	0.318	0.028	0.156	0.310	0.027	0.183	0.221
2 × 2	218	0.173	0.314	0.016	0.191	0.313	0.022	0.168	0.303	0.022	0.174	0.216
2 × 2	256	0.166	0.311	0.012	0.172	0.309	0.020	0.153	0.298	0.020	0.165	0.211
3 × 3	181	0.198	···	0.020	0.213	···	0.028	0.160	···	0.027	0.187	···
3 × 3	218	0.182	···	0.017	0.194	···	0.022	0.172	···	0.022	0.177	···
3 × 3	256	0.170	···	0.016	0.178	···	0.020	0.158	···	0.020	0.168	···
4 × 4	181	···	···	0.022	···	···	0.028	···	···	0.027	···	···

Table 3. Figures of Merit for 4 × 4 Shallow Surveys

Survey Depth	Reuleaux		Random		VLA		Geometric		Grid	
	$M = 16$	32	$M = 16$	32	$M = 16$	32	$M = 16$	32	$M = 2$	3
2	0.088	0.117	0.086	0.120	0.090	0.120	0.079	0.077	0.001	···
3	0.113	0.175	0.115	0.177	0.109	0.168	0.110	0.135	···	0.002
3[a]	0.182	···	0.165	···	0.167	···	0.136	···	···	···

Note. — All surveys used 181 pixel grid spacing.

[a]Random rather than sequential selections from the dither patterns.

Submitted The Astrophysical Journal: July 28, 1999

Calibrating Array Detectors

D. J. Fixsen[1,3], S. H. Moseley[2], and R. G. Arendt[1]

ABSTRACT

The development of sensitive large format imaging arrays for the infrared promises to provide revolutionary capabilities for space astronomy. For example, the Infrared Array Camera (IRAC) on SIRTF will use four 256×256 arrays to provide background limited high spatial resolution images of the sky in the 3 to 8 μm spectral region. In order to reach the performance limits possible with this generation of sensitive detectors, calibration procedures must be developed so that uncertainties in detector calibration will always be dominated by photon statistics from the dark sky as a major system noise source. In the near infrared, where the faint extragalactic sky is observed through the scattered and reemitted zodiacal light from our solar system, calibration is particularly important. Faint sources must be detected on this brighter local foreground.

We present a procedure for calibrating imaging systems and analyzing such data. In our approach, by proper choice of observing strategy, information about detector parameters is encoded in the sky measurements. Proper analysis allows us to simultaneously solve for sky brightness and detector parameters, and provides accurate formal error estimates.

This approach allows us to extract the calibration for many types of observations, such as surveys, from the observations themselves; little or no additional information is necessary to allow full interpretation of the data. Further, this approach allows refinement and verification of detector parameters during the mission, and thus does not depend on *a priori* knowledge of the system or ground calibration for interpretation of images.

1. Introduction

The Infrared Array Camera (IRAC) (Fazio et al. 1998) will employ four 256x256 imaging infrared arrays and the cooled telescope of the SIRTF to produce images of the sky which are

[1]Raytheon-ITSS Corp., Code 685, NASA/GSFC, Greenbelt, MD 20771

[2]Code 685, Infrared Astrophysics Branch, Goddard Space Flight Center, Greenbelt, MD 20771

[3]email address fixsen@stars.gsfc.nasa.gov

limited by the photon statistics from the natural background, which, in this spectral region (8-25 μm), is dominated by scattered and emitted light from the zodiacal dust particles. This will be typical of most future applications of infrared detectors in space. In order to produce high quality images in the presence of this strong background, the relative response of the different pixels in the detector array must be known to high precision. A technique must be developed that allows the detector properties to be determined in operation, so that the requisite stability can be experimentally verified, and changes in response can be measured and included in the analysis of the data. We present a technique by which the detector properties are determined simultaneously with the estimates of sky brightness, and formal errors developed for both instrument and sky parameters.

In this approach, we observe the same area of the sky with the detector array at a number of spatially offset positions. These observations can be used to set up a system of linear equations involving both sky brightness and detector properties. In solving this system of equations, we can deduce the sky brightness and detector gain and offset parameters. By appropriate choices of offset spacings and sky brightness distributions, this technique allows us to continuously improve our knowledge of the detector properties or detect changes. This approach embeds the relative calibration of the detector array into the survey process; all information required to produce an internally consistent survey can deduced from the survey itself. Since the data on which the calibration is based is the survey itself, it is the way to calibrate the data which is, in some sense, least susceptible to systematic errors. In the case that an *a priori* calibration is used, this technique offers a method to test internal consistency.

In this paper, we describe this least squares solution for sky and detector properties, and suggest implementations of the technique for the IRAC instrument. We present the analysis of synthetic Wide-Field Infrared Explorer (WIRE) data and real Hubble NICMOS data, in which we derive the sky brightness, detector gain and detector offset. (We had planned a demonstration of the technique on the Wide-field Infrared Explorer, but its unfortunate demise renders the point moot.) The results are encouraging, and form the basis of our plans for the analysis of the IRAC imaging data. Optimization of the observational strategy to produce the best encoding of the detector parameters in the survey observations is treated in a separate paper. This approach can offer significant insurance to the observer, in that regardless of the availability or applicability of independent relative calibration data for the instrument, sufficient information is present in the observations themselves to allow the relative calibration of the data. This provides the capability for the observer to validate the statistical properties of his data or to calibrate it as required.

Many of the results of the future observatories will derive from the analysis of survey data. The accuracy of these results will depend on a clear understanding of the statistical properties of the uncertainties in the data, their level, and spatial and temporal correlations. We present an approach for the analysis of such data, with specific application to the imaging data from the SIRTF IRAC instrument.

This comprehensive least squares approach has been successfully applied to the analysis of the data from the FIRAS instrument on COBE, in which a complex instrument model was required (Fixsen et al.1994).

2. Overview

The following equations show the derivation of the simultaneous extraction of sky brightness and instrument parameters to the data. The advantages of this system are: 1) It uses the same data for calibration and observation which saves separate observation time for calibration and uses the same time and exactly the same conditions for calibration and observation. 2) It uses a well understood process for calibration allowing for complete error analysis and flexible response in the case that unexpected errors arise. 3) It *explicitly* includes the uncertainties and correlations introduced in the calibration process in the uncertainties of the resulting data. Although we focus on an imaging array observing sections of the sky, most of the derivation is either directly applicable or easily generalized to other problems.

Although the matrices involved are large and somewhat unwieldy, the underlying process is a simple linear fit which is easily understood. The inverses of some of the matrices are assumed to exist. If there are problems inverting one or more of these matrices, it is usually an indication that some information is missing in the calibration process. So although we do not go into detail about the convergence or singularities of the process, these need to be addressed in any real system as they can show key weaknesses in the calibration process and can generally be corrected by improving the measurement strategy.

Since the details of the calibration process leave their impact on the noise characteristics of the final data set, the procedure for taking data must be be carefully designed. This is not unique to this particular process for calibration, but this procedure makes the costs of poor measurement strategies obvious.

3. Derivation of the Algorithm

The derivation of the algorithm is relatively straight forward, but specific care must be taken to avoid excessively large matrices.

We follow the Einstein summation conventions (e.g. Abram 1965) and make the notation as transparent as possible. We use different indices for the different vector spaces. Latin indices are used for the raw data and instrument pixels while greek indices are used for the derived solution and the sky pixels. We use the same variable names for the contravariant and covariant cases even though the numerical values are different, because the underlying information is the same (see Table 1).

Table 1. Variable Definitions

Variable	Definition
P	number of array pixels, e.g. $256 \times 256 = 65536$
M	number of images in the data set, e.g. 100
i, j, k, l	are indices to data $\in (1 \ldots P * M)$
D^i	data
Δ^i	model error
V^i	data variance (assumed to be diagonal)
Γ	number of observed sky locations, e.g. 500000
α, β, γ	indices to sky locations $\in (1 \ldots \Gamma), \Gamma < P * M)$
S^α	set of sky parameters
p	index to pixels $\in (1 \ldots P)$
G^p	set of gain parameters
F^p	set of offset parameters
q, r	indices to gains and offsets $\in (1 \ldots 2P)$
X^q	set of gain and offset corrections $(\delta F^p, \delta G^p)$
μ, ν, ω	indices to all parameters $\in (1 \ldots 2P + \Gamma)$

Here we adopt a simple model for the data, but in general more complex models are as easily handled as long as they are relatively linear in the range of interest, do not require large numbers of parameters to be determined, and are not underdetermined. Nonlinear parameters are notoriously slow to converge. A formal derivative must be calculated for each of the extra parameters and coded into the algorithm. Of course if some part of the parameter space is undetermined the program may not converge.

In our example, the model, H^i for the data is given by

$$H^i(G^p, S^\alpha, F^p) = G^p S^\alpha + F^p. \tag{1}$$

First we linearize the equation, yielding:

$$H^i = H_0^i + \partial_{G^p} H^i \delta G^p + \partial_{F^p} H^i \delta F^p + \partial_{S^\alpha} H^i \delta S^\alpha. \tag{2}$$

The linearization in equation (2) is best if we chose a point close to the solution. One might iterate the whole procedure to find the best solution but a simpler paradigm is to use the last solution as the starting point and either add more data to improve the solution or find a new solution with new data. Either way, only once, do we need to start without a previous solution. In that case we can let $G^p = 1$, $F^p = 0$, and $V^p = \sum_{i \in p}(D^i - F^p)^2/M$. Then with the assumption that the uncertainties are a function of pixel only we have an estimate for V^i. We will return to this estimation in section 6.

This is obviously an oversimplification but suitable as an initial estimate. After the first fit, the model (including sky and instrument parameters) can be removed and the residuals can be studied for a better expression for the variance. The gains and the offsets subsequently are available as a first approximation for the solution.

There are several potential pitfalls here particularly if the second derivative is ill-behaved. In our example these are not an issue and in general as long as the second derivative is not too large this linearization is well behaved. Consistent with this linearization, we introduce:

$$\Delta^i = D^i - H_0^i \tag{3}$$

and

$$H_\mu^i = \frac{\partial H^i}{\partial \delta^\mu} \tag{4}$$

where

$$\delta^\mu = (\delta G^1 \ldots \delta G^P, \delta F^1 \ldots \delta F^P, \delta S^1 \ldots \delta S^\Gamma) = (X^1 \ldots X^P, X^{P+1} \ldots X^{2P}, \delta S^1 \ldots \delta S^\Gamma). \tag{5}$$

Combining all of the detector parameters into the vector X is a notational convenience. The parameters are not required to have the same units; the weight matrix has all of the appropriate inverse units. Furthermore, the fundamental statistical units are units of the uncertainty, σ, not volts, counts or even electrons.

So with weights

$$W_{ii} = (V^i)^{-1} \tag{6}$$

χ^2 is calculated as

$$\chi^2 = (\Delta^i - H^i_\mu \delta^\mu) W_{ij} (\Delta^j - H^j_\nu \delta^\nu) \tag{7}$$

and its minimum is determined by

$$\begin{aligned} \frac{\partial \chi^2}{\partial \delta^\omega} &= -H^i_\omega W_{ij} (\Delta^j - H^j_\nu \delta^\nu) - (\Delta^i - H^i_\mu \delta^\mu) W_{ij} H^j_\omega \\ &= -2 H^i_\omega W_{ij} \Delta^j + 2 H^i_\omega W_{ij} H^i_\nu \delta^\nu = 0. \end{aligned}$$

Thus

$$H^i_\mu W_{ij} \Delta^j = H^i_\mu W_{ij} H^j_\nu \delta^\nu = \delta_\mu$$

or

$$\delta_\mu = H^i_\mu \Delta_i$$

and the solution for δ^μ can be expressed as

$$\delta^\mu = (H^i_\mu W_{ij} H^j_\nu)^{-1} H^k_\nu W_{kl} \Delta^l = (H^i_\mu H_{i\nu})^{-1} H^k_\nu \Delta_k. \tag{8}$$

We assume the residual errors are uncorrelated so the weight matrix is diagonal. The formalism still holds even if the errors are correlated (the covariance matrix V has off diagonal components), but the process of inverting the large matrix V (which might be $1,600,000 \times 1,600,000$) may require considerable attention itself. Also note that this does not mean that the data are uncorrelated. Indeed, the data are correlated as some of the data are derived from the same pixel or are observations of the same part of the sky with different detectors. If there are other sources of correlation (such as detector temperature) they need explicitly to be included in the model. The assumption here is that the residual errors are uncorrelated.

Identifying the weight matrix (or metric) as the inverse of the covariance matrix, only defers the question to how to determine the covariance matrix. There are two sorts of ways to attack this problem. The theoretical approach uses *a priori* knowledge about the system to estimate what the noise should be. This can include such things as the Poisson arrival of photons, the Johnson noise of the resistors and other known sources of noise. The empirical approach uses the residuals in the data itself to make an estimate of the noise. Each approach has its strengths and weaknesses. The theoretical approach often underestimates the noise because there are unmodeled sources present. The empirical approach often overestimates the noise, as it treats parts of the signal that are not properly modeled as noise. If both approaches lead to the same estimate one has reasonable assurance that the model and estimate are correct. If the approaches differ significantly there are either noise sources that are included in the estimate or signal that is not included in the model. Sometimes a detailed look at the residuals can give clues to where the extra variance is coming from. In this case, we assume that the noise variance V is known.

To facilitate the calculation, let the solution weight matrix,

$$H^i_\mu W_{ij} H^j_\nu = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}. \tag{9}$$

Equation (9) breaks the solution weight matrix into three parts. The part dealing with the instrument, A, the part dealing with the resulting sky map, C, and the connections between them, B. The covariance matrix (inverse of the weight matrix) can be broken into the same sorts of parts. It is particularly convenient for calculation in this problem that the large submatrix, $C = H^i_\alpha W_{ij} H^j_\beta$ is diagonal, since each detector observes only one sky pixel at a time. Therefore C is easily handled and inverted. Let us then consider

$$(H^i_\mu W_{ij} H^j_\nu)^{-1} = (H^i_\mu \; H_{i\nu})^{-1} = \begin{pmatrix} Q & R \\ R^T & \Psi \end{pmatrix}. \tag{10}$$

The dimensions of A and Q are $2P \times 2P$, those of B and R are $2P \times \Gamma$, and those of C and Ψ are $\Gamma \times \Gamma$. While these are large, they are not as large as the matrix of the original data. When the only interest is in the uncertainties in the array parameters, (e.g. when the calibration is used for other data) only Q is needed. Similarly, if only the sky uncertainties are required, only Ψ is needed.

The inverse or covariance can be calculated by:

$$Q = (A - BC^{-1}B^T)^{-1} \tag{11}$$

$$R = -QBC^{-1} \tag{12}$$

and

$$\Psi = C^{-1} + C^{-1}B^T Q B C^{-1}. \tag{13}$$

The covariance of the derived sky, Ψ, is composed of two parts. The C^{-1} is the direct propagation of the measurement errors to the sky. The other part $C^{-1}B^T Q B C^{-1}$ shows the additional uncertainty due to the calibration. For a well chosen set of observations this part can approach $\frac{2}{M-1}C^{-1}$, the limit set by the number statistics.

Now equation (8) is really a system of linear equations. By substituting equation (10) into equation (8) and retaining only the first $2P$ equations we have:

$$X = (Q, R) H^k_\mu \Delta_k = QY \tag{14}$$

where

$$Y = H^k_q \Delta_k - BC^{-1} H^k_\alpha \Delta_k \tag{15}$$

with dimension $2P$, is easily calculated.

The submatrix $A = (H^i_q H_{ir})$ has two parts. There is a diagonal part and an off diagonal component relating the gain and offset of each pixel. The whole matrix can be treated as P 2×2

matrices. Although there is not a unique $A^{-1/2}$, we chose one and use it in the following equations. Mathematically the choice is arbitrary, but some choices are easier to program and less susceptible to numerical instability.

Let

$$T = A^{-1/2}BC^{-1/2} = (H_q^i H_{ir})^{-1/2} H_r^j H_{j\alpha} (H_\alpha^k H_{k\beta})^{-1/2}. \tag{16}$$

Then

$$X = (A - BC^{-1}B^T)^{-1}Y = A^{-T/2}(I - A^{-1/2}BC^{-1}B^T A^{-T/2})^{-1}A^{-1/2}Y = A^{-T/2}(I - TT^T)^{-1}A^{-1/2}Y \tag{17}$$

the size of T is $2P \times \Gamma$, but it is sparse.

Finally,

$$X = QY = A^{-T/2}\left[\sum_{n=0}^{\infty}(TT^T)^n\right]A^{-1/2}Y. \tag{18}$$

This then is a form which can be handled by a modest computer. The vectors X and Y are each only $2P = 131072$ long. The matrix A can be stored as a $2P$ long diagonal part and a P long off diagonal part (A is symmetric). If we chose a symmetric $A^{-1/2}$ it requires the same storage although many other forms use no more than $4P$ storage. The matrix T is nominally large ($2P \times \Gamma$) but is sparse and has at most $2P \times M$ nonzero components. The infinite sum in general is of some concern but with a well chosen observation strategy the convergence is relatively fast and only tens to hundreds of terms are required depending on the desired precision of the result.

At this point the reader might observe that there are two obvious singularities in the matrix. These correspond to the uniform increase in the sky brightness and a cancelling decrease in the offset (or vice versa), and to a multiplication of the sky by an arbitrary amount and a cancelling effect in the gain term. These two singularities merely point out what we already know; in order to get an *absolute* calibration we need an *absolute* standard. There are several ways to deal with this issue: 1) An absolute calibration could be done in the laboratory. 2) Certain places on the sky could be determined in some other way and used to impose a condition that would break the singularity. 3) A map could be produced with an arbitrary gain and offset.

The three methods are not mutually exclusive. A map with arbitrary gain and offset can be produced which is subsequently calibrated by laboratory measurements or sky measurements or a combination of sky and laboratory measurements. The absolute calibration can be included in the fit or applied later. We choose to apply it separately as this maintains the uniformity of the algorithm whether viewing a calibration object or not.

Without treating the singularity the sum in equation (18) may not converge. If there are no dark frames to determine the offset, after each iteration we impose the condition that $\sum_p \sqrt{\sum_{i\in p} W_i}\ \delta F^p = 0$. The weight applied to the δF^p is only for computational convenience. The key is that the net offset is not allowed to change, that is the average array offset is kept

fixed. If dark frames are present we can use them to determine the offset and do not impose this condition. Similarly, a weighted mean gain is held fixed, $\sum_p \sqrt{\sum_{i\in p, i\in\alpha} S^\alpha W_i S^\alpha}\ \delta G^p = 0$.

4. Procedure

Next we show how the derived algorithm can be used in a practical program. A detailed program is given in the appendix, here we outline the steps of the program and relate them to the previous derivation.

Although the size of H is $2P \times \Gamma$ it can be treated as a set of delta functions. With care in the processing, the parts of H that are zero need never be accessed. There are $3P * M$ nonzero parts. That is each datum appears 3 times, once associated with G, F and S.

We now make use of the following relations:
$$\begin{aligned} \partial_{G^p} H^i &= S^\alpha \delta_{pi} \\ \partial_{F^p} H^i &= \delta_{pi} \\ \partial_{S^\alpha} H^i &= G^p \delta_{\alpha i} \end{aligned}$$

First we build

$$\Delta_i = W_i(D^i - G^p S^\alpha - F^p). \tag{19}$$

Then build A, in which the diagonal part is

$$\text{diag}\ \ A = \left(\sum_{i\in p, i\in\alpha} S^\alpha W_i S^\alpha \right), \left(\sum_{i\in p} W_i \right) \tag{20}$$

and the off diagonal part is

$$\text{off diag}\ \ A = \left(\sum_{i\in p, i\in\alpha} S^\alpha W_i \right) \tag{21}$$

and

$$C = \left(\sum_{i\in\alpha, i\in p} G^p W_i G^p \right). \tag{22}$$

C is diagonal.

Next we need to compute B

$$B = (\sum_{i\in p, i\in\alpha} S^\alpha W_i G^p), (\sum_{i\in p, i\in\alpha} W_i G^p). \tag{23}$$

Then

$$Y = \left(\sum_{i\in\alpha, i\in p} S^\alpha \Delta_i, \sum_{i\in p} \Delta_i \right) - BC^{-1} \sum_{i\in p, i\in\alpha} G^p \Delta_i. \tag{24}$$

Note that B is $2P \times \Gamma$ but it is sparse. We also have all of the pieces to calculate $T = A^{-1/2}BC^{-1/2}$. We now have all of the elements of equation (18). We then iterate via $Z_0 = A^{-1/2}Y$

$$Z_{n+1} = Z_0 + TT^T Z_n \tag{25}$$

until Z is stable. It is trivial then to get the solution $X = A^{-T/2}Z$. This is only the solution for the detector, but the solution for the sky is then easy to derive:

$$S^\alpha = \sum_{i\in\alpha, i\in p} (D^i - F^p) * G^p W_i / \sum_{i\in\alpha, i\in p} (G^p)^2 W_i. \tag{26}$$

This completes the solution for the detector and the sky. The calculation of a single uncertainty vector is completely analogous. However the full covariance matrix Ψ is $\sim 500000 \times 500000$. This matrix is symmetric but it is not necessarily sparse. In fact it is likely that all of the elements are nonzero. The 2.5×10^{11} components of Ψ are awkward to carry around but they contain all of the information about the correlations imposed by the calibration process. It can be stored more compactly by keeping T, and noting that

$$\Psi = C^{-1/2}(I - T^T T)^{-1} C^{-1/2} \tag{27}$$

since T is sparse and $C^{-1/2}$ is diagonal.

Now we return to the issue of variance (weight) estimation. Without some model for the noise we have a hopeless task. However with a simple model we can make some estimation of the variance. An unbiased, but poor, estimation only increases the noise (and the estimate of the uncertainty).

In the model program we assume three sources of error: 1)Poisson statistics, 2)A pixel dependent readout noise, 3)A cosmic ray induced error. The Poisson noise is easily calculated if the approximate gain of the system is known. The readout noise is best estimated by using the RMS of all of the data from that pixel (except the cosmic ray contaminated data). Cosmic rays can be identified by seeking large discrepancies. These should not be used in either the sky or variance estimation. Obviously as real data are collected a more detailed model can be developed.

After a solution is found, the model program recalculates Δ^i. Large values are assumed to be cosmic particle hits or other glitches. These data are marked to not be used in the next iteration. The remaining Δ^is are squared and summed to form an estimate of the noise. In the model program the noise is treated separately for each pixel. If large amounts of data are available this process could potentially identify subtle problems with particular pixels. If only small amounts of data are available some smooth approximation over entire detector array may be more appropriate.

5. Practical Matters

The algorithm described in the preceding sections can produce mosaics of large regions provided that at least some parts of the region (preferably all parts) contain repeated (dithered) observations. The algorithm can be applied to a data set containing spatially separate regions. There is no constraint on the size or geometry of the region(s) in the data set. It is only required that the detector gain and offset and the sky intensity (G^p, F^p, and S^α) are constant for the entire data set. These restrictions can be relaxed by explicitly parameterizing known or suspected variations.

If dark frames are available, they can be added to the data set as if they are observations of a region of sky that is separate from the rest of the data and that has an intensity $S^\alpha \equiv 0$. The addition of dark frames to the data set allows the algorithm to determine the offset components.

The algorithm can be implemented in a general manner, such that the detector dimensions and number of frames processed are adjustable. An example of such a code is given in the Appendix. A general code can be applied to different data sets from different instruments if a new "front end" is written for each type of data to ingest the data and provide the necessary initial estimates and control parameters.

The selection of the weights (W_{ii}) to use in the algorithm can be important. The nature of the data will determine the best choice of weights. Poor weighting of the data may cause spurious features to propagate through the solutions for G^p, F^p, and S^α. Cosmic ray hits on the detector will also cause spurious features in the results, if not properly handled. Data affected by cosmic rays can be given very low weights or flagged. It is best, if the effects of cosmic rays can be removed from the data before processing, though this is not always possible. The algorithm can be modified to recognize cosmic rays as outliers provided that they are not so numerous that they severly bias the results.

In most cases, the algorithm will probably be used iteratively for about 2 - 5 cycles. Subsequent iterations use the previously derived gain and offset values as inputs, and can make use of successively improved weights and exclusions of cosmic rays as well.

The example code presented in the Appendix requires free memory ~15 times larger than the size of the data set to be processed. For a data set of 27 256×256 images the algorithm takes ~450 seconds of CPU time on a 300 MHz Pentium II machine running Red Hat Linux 5.2 and IDL 5.0.3. About 270 seconds of that CPU time is spent in the calculation of the summation of equation (18), using the iterative step of equation (25) for 100 terms. The time for the procedure is linear with the number of input data elements as long as more iterations are not needed. The number of iterations required is strongly related to the connection map which is determined by the dither pattern of the input data.

Solving only for detector gains in cases where the detector offsets are negligible is a minor simplification of the algorithm and is a more robust procedure. Figure 1 illustrates the results of

using this procedure to solve for only the detector gains and sky intensities. The data used is from Wide-Field Infrared Explorer (WIRE) simulations. The model for the sky includes point sources, cirrus and a zodiacal background. The model for the 128×128 detector array included gain variations, bad pixels, and cosmic ray hits. The data set consists of 10 dithered images, one of which is shown in the upper left of Figure 1. The detector gain variations dominate the qualitative appearance of the data. The derived gain compares favorably with the true gain, with the exception of a few pixels with remaining artifacts from bad pixels or cosmic rays. There is a small (~ 1.0025) scale factor between the derived and true gains, which simply reflects the lack of absolute calibration in the procedure. The derived sky is a good representation of the real sky, with the additional noise component indicated by the second term of equation (13).

Figure 2 illustrates the application of the algorithm to real data, namely the HST NICMOS observations of the Hubble Deep Field - South. The raw data used here were 59 good 1152 and 1472 s integrations. The worst effects of cosmic rays were eliminated by calculating linear fits to the multiaccum readouts from each pixel. Fits with poor correlation coefficients were refit using a combination of linear and step functions. Additional pre-processing involved setting the median values in each detector quadrant to a common value for all frames. This helped compensate for a variable "pedestal" effect which is not modelled by our current algorithm. (The bottom 16 rows of each frame were ignored in the processing to avoid vignetting effects.) The initial gain map was assumed to be flat and unity. The initial offset map was assumed to be flat and zero. A dark file from the NICMOS reference files was used for a simulated dark frame that was processed simultaneously with the sky data. The derived sky after 2 iterations of the algorithm and truncation of the series expansion after 100 terms, is not as clean as the publicly released processed data. Spurious large scale structure is present at low levels. A faint stripe along the detector columns is visible through the brightest star in the field. The gain and offset maps are similar to calibration flat and dark reference files. In our derived gain and offset maps there are some residual defects in pixels where the bright sources in the map were observed. The gain and offset maps also contain visible quadrant errors and vertical bars from "shading" because of instrumental effects that are not adequately described by the simple method used here. Clearly there is room for improvement, but the algorithm worked well. The process allowed the simultaneous determination of sky and detector parameters using only sky measurements and dark frames. By inspecting the residuals there are indications that the offsets are not constant from observation to observation. This suggests an improved model for the data could be constructed by parameterizing and fitting these offsets.

In the case of IRAC, such an algorithm is essential. With it, we can continuously derive detector parameters from the normal observations and improve the model of the detectors as well as the model of the sky. Just such a procedure was used on the FIRAS data to improve the sensitivity by a factor of ~ 100 over the intial publication.

6. Uncertainties and Correlations

The algorithm produces a formal estimate of the uncertainties, Ψ, based on the derivation and the estimated uncertainties of the input data, V. The resulting uncertainties are only as good as the uncertainty estimates of the original data. Those uncertainties, V, are checked against the actual deviations from the model to either give an improved estimate of the input data uncertainty or an indication of short comings of the model.

The calibration process also introduces correlations into the resulting map, $C^{-1}B^TQBC^{-1}$. The correlations for a single detector, Q, can be easily generated by using a unit vector in place of the Y and carrying out the calculations as with the data. The process is slightly shorter than for the data as some steps can be ignored. Obviously this could be repeated for each of the detectors and then equation (13) can be used to generate the full covariance matrix.

There are two problems with this approach. First, the time required is proportional to the number of detectors (65536 for IRAC or NICMOS data). Second, the space for the final result is Γ^2, which is $\sim 10^{11}$, for even the modest WIRE example shown here. Storing and using such a large data set is problematical.

Fortunately the correlations for different detectors are nearly identical (see figure 3). This should not be a surprise since the detectors are locked into their relative positions and all move together in each dither move. Bad detectors in the array, cosmic rays, and rotations will obviously break this symmetry but except for the rotations the effects are minor and rather localized. So the correlations can be calculated for a few typical detectors and the results can be used for the entire data set.

7. Summary

As demonstrated with the simulated WIRE data the program can calculate the gains and offsets to the theoretical limit on the accuracy if it is given a good model of the data. As shown with the NICMOS data the program works reasonably well on real data as well even with he normal complexities of real errors and uncertainties. The uncertainties are calculated, and the correlations can be calculated with minor changes to the program. These allow the user to interpret the result without ad hoc assumptions or guesses about how the errors are related. The speed of the program allows modest data sets to be processed in a few minutes and with the availability of machines with large memories will allow the large data sets of the future to be processed in reasonable times.

8. Acknowledgements

We thank D. Shupe and the WIRE team for supplying simulated WIRE observations for testing the algorithm.

9. References

Abram, J., Tensor Calculus Through Differential Geometry, (Butterworth: London), 1965.

Fazio, G. G. et al.1998, Proc. SPIE, 3354, 1024

Fixsen, D. J. et al.1994 ApJ 420,427

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C: The Art of Scientific Computing, Second Edition, Cambridge: Cambridge University Press

A. An Example Calibration Code: `gain_offset.pro`

A code to implement the algorithm described in this paper has been written as an IDL procedure **`gain_offset.pro`**, with calls to external C functions. In outline form, the code contains the following sections:
1) check dimensions of input data
2) create pixel indexes for the data
3) build an initial estimated sky (H_0^i) and $\Delta_i = W_{ii}(D^i - H^i)$
4) build $A^{-1/2}$ and detector subcomponents of Y
5) build $C^{-1/2}$ and pixel subcomponents of Y
6) build T
7) calculate $X_p = A^{-1/2}Z$ where $Z_n = Z_0 + T(T^T Z_n)$
8) construct total gain, offset and sky images
9) calculate uncertainties and revised weights and cosmic ray flags
10) make final sky image using updated weights and cosmic ray flags

The entire IDL code is listed in the following subsections, with each subsection of code preceded by a description. The next section contains listings and descriptions of the called subroutines and functions.

Within the context of this code, a *pixel* refers to a observed location in the final sky map, a *detector* refers to a single detector element within the detector array, and a *stack* refers to a three dimensional data cube where the first two dimensions are the P_x and P_y dimensions of a single image ($P_x = P_y$ is not required) and the third dimension counts the number of images in the data

set.

```
pro gain_offset, Di, dx, dy, dth, xc, yc, Wi, n_terms, Gp, Fp, $
              skymap, sigmadet, sigmasky, cr_flag, dark_flag, $
              n_sig=n_sig, rxy=rxy
```

A.1. Find relevant dimensions

The IDL `size` function is used to read the dimensions of the input data stack, and set P_x, P_y, P, M, and the total number of data, $P \times M$. If `dark_flag` is undefined, all data are assumed to be sky data.

```
sky_size = size(Di)
Px = sky_size(1)
Py = sky_size(2)
P = sky_size(1)*sky_size(2)
M = sky_size(3)
PxM = sky_size(5)
if (n_elements(dark_flag) ne M) then dark_flag = bytarr(M)+1B
```

A.2. Find and sort pixel indices

The function `get_pixel_numbers` constructs a square grid of sky pixels large enough to contain all M dithered images, and then extracts the pixel number (α) for each D^i. These indices are returned in a data-like stack (`pix_no`). Additional returned parameters `xs`, `ys`, `xo`, and `yo` specify the size and minimum x-y coordinates of the grid in which the `pix_no` are embedded. The indices of the pixels in a sorted sequences are saved in `pix_sort`, which is then examined to determine where the data for each pixel can be found (saved in `starts`) and the total number of pixels observed (Γ = `Gamma`).

Later when we need to calculate summation over detector q, we can simply add columns in the data-like stacks. Summation over pixel α will use `pix_sort` and `starts`.

```
pix_no=get_pixel_numbers(dx,dy,dth,xc,yc,sky_size,xs,ys,xo,yo,dark_flag,rxy=rxy)
pix_sort = sort(pix_no)
starts = [0,uniq(pix_no(pix_sort))+1]
Gamma = n_elements(starts)-1
```

Table 2. Input and Output Parameters

Parameter	Type[a]	Description
`Di`	I	data D^i in a stack of M $P_x \times P_y$ images.
`dx`	I	array of M x-offsets for the images.
`dy`	I	array of M y-offsets for the images.
`dth`	I	array of M rotations (in radians) for the images.
`xc`	I	detector x-coordinate of the center of rotation of the array.
`yc`	I	detector y-coordinate of the center of rotation of the array.
`Wi`	I	weights W_i in a $P_x \times P_y \times M$ data-like stack.
`n_terms`	I	maximum number of terms in to use in the summation.
`Gp`	I/O	$P_x \times P_y$ array containing the initial/final gain map.
`Fp`	I/O	$P_x \times P_y$ array containing the initial/final offset map.
`skymap`	O	final solution for the sky intensities.
`sigmadet`	O	final solution for the detector uncertainties.
`sigmasky`	O	final solution for the pixel (sky) uncertainties.
`cr_flag`	I/O	$P_x \times P_y \times M$ data-like stack containing the suspected/derived locations of CR hits (0=hit,1=OK).
`dark_flag`	i	array of M bytes with values of 1 or 0, indicating sky data or dark frame for each of the M images in the data set. All data are assumed to be sky data if `dark_frame` is unspecified.
`n_sig`	i	number of sigma above test value to trigger CR flag. If unspecified, a default value is used within the `make_sig` procedure.
`rxy`	i	ratio of the x/y dimensions of the individual detectors. It is assumed to be 1.0 if unspecified. The sky is mapped on square pixels the same size as the y-dimension of the detectors.

[a]I = input, O = output, i = optional input

A.3. Estimate S^α and find Δ_i

The first part of this section builds an initial guess of the sky intensities as given by equation (25). S^α is calculated from each D^i, and a weighted average over all observations of a given pixel is determined. For later convenience `Sa` is held in a data-like stack.

However, in order to deal with data from rotated frames and at fractional pixel positions, the following additional steps are taken. A rough skymap is constructed, any holes are filled by averaging values from the 8 adjacent pixels, a 2-dimensional spline is fit to the map, and then `Sa` values for the true locations of the data are interpolated from the spline fit. The `Sa` values for dark frames are set to 0.0 exactly.

The final part of this section calculates $\Delta_i = W_{ii}(D^i - H_0^i)$, which is saved as `Di_H0i`.

```
wG = Wi
for i=0L,M-1 do wG(*,*,i) = Wi(*,*,i)*Gp^2
skymap = make_map(Di,pix_no,pix_sort,xs,ys,wG,Gp,Fp,wtmap)
wG = 0
ws = skymap*wtmap
ws=shift(ws,1,0)+shift(ws,0,1)+shift(ws,-1,0)+shift(ws,0,-1)+ $
   shift(ws,1,1)+shift(ws,1,-1)+shift(ws,-1,1)+shift(ws,-1,-1)
w=shift(wtmap,1,0)+shift(wtmap,0,1)+shift(wtmap,-1,0)+shift(wtmap,0,-1)+ $
   shift(wtmap,1,1)+shift(wtmap,1,-1)+shift(wtmap,-1,1)+shift(wtmap,-1,-1)
h = where((wtmap eq 0) and (w gt 0),nh)
if (nh ge 1) then skymap(h) = ws(h)/w(h)

Sa = fltarr(Px,Py,M)
spl2,skymap,a,b,c
sz = sin(dth)
cz = cos(dth)
for k=0L,M-1 do begin
  if (dark_flag)then begin
    i = lindgen(Px,Py) mod Px - xc(k)
    j = lindgen(Px,Py) / Px - yc(k)
    x = dx(k)+rxy*(xc(k)+cz(k)*i)+sz(k)*j-xo
    y = dy(k)+yc(k)-rxy*(sz(k)*i)+cz(k)*j-yo
    spv2,skymap,a,b,c,x,y,z
    Sa(*,*,k) = z
  endif
endfor

Di_H0i = Di
```

```
for i=0L,M-1 do begin
  Di_H0i(*,*,i) = Wi(*,*,i) * (Di(*,*,i)-Gp*Sa(*,*,i)-Fp) * cr_flag(*,*,i)
endfor
```

A.4. Construct $A^{1/2}$ and detector part of Y

This section calculates $A^{-1/2}$, saved as `irootA`, as described in §3. $A^{-1/2}$ is a $2P \times 2P$ matrix than can be divided into 4 diagonal quarters. The P diagonal elements from each quarter of $A^{-1/2}$ are stored in $P_x \times P_y (= P)$ dimensioned layers of the `irootA` stack in the following sequence: [upper left, lower right, upper right, lower left].

Then the parts of Y involving summation over detectors are calculated (left side of equation 23), and saved in `delHD`. `delHD` $= H_q^k \Delta_k$ or $H_\alpha^k \Delta_k$ of equation (15). It is not necessary to actually calculate and save Y itself.

The inverse square-root of A is not unique. Presently, is seems that the asymmetric form used here works better than the symmetric form which was also tested.

```
irootA = fltarr(Px,Py,4)
Gp_A = total(Sa*Wi*Sa,3)
Hp_A = total(Sa*Wi,3)
Fp_A = total(Wi,3)
dHp_A = dblarr(Px,Py)
for i=0L,M-1 do dHp_A = dHp_A + Wi(*,*,i)*(Sa(*,*,i)-Hp_A/Fp_A)^2
D_A = sqrt(Fp_A*dHp_A)
D2_A = sqrt(Gp_A*Fp_A - Hp_A^2)
Q_A = sqrt(Gp_A + Fp_A + 2*D_A)

irootA(*,*,0) = sqrt(Fp_A)/D_A
irootA(*,*,1) = 1./sqrt(Fp_A)
irootA(*,*,2) = -Hp_A/sqrt(Fp_A)/D_A
irootA(*,*,3) = 0

delHD = fltarr(2*P+Gamma)
delHD(0:P-1) = total(Sa*Di_H0i,3)
delHD(P:2*P-1) = total(Di_H0i,3)
```

A.5. Construct $C^{1/2}$ and sky part of Y

This section calculates $C^{-1/2}$ (`irootC`) and the pixel (sky) summation required in the right hand side of Y as in equation (24). The sums are saved in `delHD`. Since C is diagonal, only the diagonal of $C^{-1/2}$ is non-zero and saved in `irootC`.

At this point the variable `pix_ind` contains the pixels renumbered $(0...\Gamma-1)$ so that they simply enumerate the observed sky pixels, rather than correspond to particular coordinates in the arbitrary grid defined by `get_pixels_numbers`.

```
irootC = fltarr(Gamma)
pix_ind = lonarr(Px,Py,M)
for i=0L,Gamma-1 do begin
  here = pix_sort(starts(i):starts(i+1)-1)
  delHD(2*P+i) = total(Gp(here mod P)*Di_H0i(here))
  pix_ind(here) = i
  irootC(i) = total(Wi(here)*Gp(here mod P)^2)
endfor
Di_H0i = 0 ; save memory
irootC = 1./sqrt(temporary(irootC))
```

A.6. Construct T

This section calculates $T = A^{-1/2}BC^{-1/2}$, which is a sparse matrix with dimensions $2P \times \Gamma$. The locations of non-zero elements in T are the same as those in B matrix. The first $P \times \Gamma$ part of the B matrix is the "gain" part, with elements $\sum S^{\alpha}W_iG^p$, and the second $P \times \Gamma$ part is the "offset" part, with elements $\sum W_iG^p$. The summations here extend over all data i that encompass observations of pixel α with detector p.

Unless there are duplicate pointings of the array, each sum in B will include only a single term. Therefore, there is a direct mapping between the data, D^i, and non-zero elements in B and T. This allows us to retain and work with T in a data-like stack, `T`, as long as we can keep track of the correct p, α locations of each of the elements. The p location of each element is tracked by the element's location with respect to the first two dimensions of the stack, and the α location of each element is tracked by the `pix_ind` stack. In the event that there are duplicate pointings, and thus at least some of the sums in B have more than one term, we can still use `T` in this stack format, as long as the sums are properly combined later when `T` is used in further calculations.

The operations needed to transform B to T are basically multiplications of each column of B by a vector, and each row of B by another vector. The process is made slightly more complex because 1) the dimension of B is $2P$ instead of P (so certain calculations need to be done twice), 2)

$A^{-1/2}$ has a second set of off-diagonal terms (so calculations involving `irootA` have an additional part), and 3) the format of `T` requires the "row" multiplication of B by $C^{-1/2}$ to use the `pix_sort` and `starts` arrays for locating all elements in a "row" of `T`.

```
T = [[[temporary(Sa)*Wi]],[[Wi]]]
for i=0L,2*M-1 do T(*,*,i) = irootA(*,*,i/M)*T(*,*,i)*Gp $
                           + irootA(*,*,i/M+2)*T(*,*,(i+M)mod(M*2))*Gp
for i=0L,Gamma-1 do begin
  here = pix_sort(starts(i):starts(i+1)-1)
  T(here) = T(here) * irootC(i)
  T(here+PxM) = T(here+PxM) * irootC(i)
endfor
T = [[T(*,*,0:M-1)],[T(*,*,M:*)]]
irootA = reform(irootA,2*P,2)
```

A.7. Calculate the series expansion

This section is where the bulk of the calculation takes place. First $Z_0 = A^{-1/2}Y$ is calculated. From equation (24) we see that

$$Z_0 = A^{-1/2}\left(\sum S^{\alpha}\Delta_i, \sum \Delta_i\right) - A^{-1/2}BC^{-1/2}C^{-1/2}\sum G^{p}\Delta_i. \tag{A1}$$

The sums in this expression have been saved in `delHD`, and $A^{-1/2}BC^{-1/2} = T$. So first the right-hand term of the above expression is calculated, $TC^{-1/2}\sum G^{p}\Delta_i$, which is then subtracted from the left-hand term of the expression.

The sparse matrix multiplication required in this first step, `T (irootC*delHD)`, is of the form $\mathbf{b} = \mathbf{A}\ \mathbf{x}$ where $\mathbf{b}$ and $\mathbf{x}$ are vectors and $\mathbf{A}$ is a matrix. The multiplication is carried out using a call to an externally compiled C function `spax.so`. This function is based on the function `sprsax` in "Numerical Recipes in C", (Chapter 2.7). The `spax` function is described in detail later. External C routines provide faster sparse matrix multiplication than equivalent routines in IDL.

Once Z_0 is calculated, we use equation (24) in the form $Z_{n+1} = (T(T^T Z_n)) + Z_0$ to calculate the series expansion of equation (18) to a sufficient number of terms. Rather than calculating and storing TT^T to multiply against Z_n, we perform the multiplication in two steps: $T^T Z_n$ and $T(T^T Z_n)$. `spax` is used again for the latter multiplication, and a minor variant, `sptx`, handles the multiplication of the transpose of the sparse matrix.

The variables `dZ0` and `dZ1` are used to track the changes in Z (`Zn`) through each step of the summation. `sZ` tracks the ratio of these changes between one step and the next. These can be used to monitor the convergence of the series. The while loop is set to sum over at least 5 terms

and no more than n_terms terms. Ideally the loop will terminate after an intermediate number of terms when sZ becomes small, i.e. successive terms become rapidly smaller corrections.

After each term of the series is added to Z_n, the gain parts are adjusted to have a 0.0 mean value. Z_0 is subjected to this normalization as well, before the series summation begins. This eliminates the singularity which otherwise would allow the sky intensities and detector gains to be altered by reciprocal factors.

The final step is to recover the actual gain and offset corrections, $X_p = A^{-1/2}Z$.

```
Z0 = fltarr(2*P)
status = call_external('spax.so','spax',T,[[pix_ind],[pix_ind]],$
  irootC*delHD(2*P:*),Z0,2*P,2*PxM)
Z0 = -Z0 + irootA(*,0)*delHD(0:2*P-1) + irootA(*,1)*shift(delHD(0:2*P-1),P)
Z0(0:P-1) = Z0(0:P-1) - total(Z0(0:P-1))/P ; normalization
Z0 = float(Z0)
Zn = Z0
dZ0 = Z0
sZ = fltarr(n_terms+1)
i = 0L
while (((i lt 5) or (sZ(i) gt 0.05)) and (i lt n_terms)) do begin
  i=i+1
  TTtZn = fltarr(2*P)
  TtZn = fltarr(Gamma)
  status = call_external('sptx.so','sptx',T,[[pix_ind],[pix_ind]],$
                         Zn,TtZn,2*P,2*PxM)
  status = call_external('spax.so','spax',T,[[pix_ind],[pix_ind]],$
                         TtZn,TTtZn,2*P,2*PxM)
  TTtZn(0:P-1) = TTtZn(0:P-1) - total(TTtZn(0:P-1))/P ; normalization
  dZ1 = TTtZn+Z0-Zn
  dZ1 = dZ1 - total(dZ1)/(2*P)
  sZ(i) = total(dZ0*dZ1)/total(dZ0^2)
  dZ0 = dZ1
  Zn = TTtZn+Z0
endwhile
Xp = irootA(*,0)*Zn + shift(irootA(*,1),P)*shift(Zn,P)
T = 0 ; save memory
pix_ind = 0 ; save memory
```

A.8. Find the new G^p and F^p

This section adds the gain and offset corrections to the initial guesses, and derives the sky intensities from the data using the function `make_map`.

```
Gp = Gp+reform(Xp(0:P-1),Px,Py)
Fp = Fp+reform(Xp(P:2*P-1),Px,Py)
skymap = make_map(Di,pix_no,pix_sort,xs,ys,Wi*cr_flag,Gp,Fp,wtmap)
```

A.9. Calculate uncertainties

The penultimate section of the code is intended to calculate the uncertainties on gain, offset, and sky, and to identify cosmic ray (CR) hits in the data. The uncertainties can be used in weighting the data if another iteration of the procedure is run with the new gain and offset maps as initial guesses. The revised CR flag is also input for a next iteration. The `make_sig` subroutine may need to be revised to best suit the characteristics of data sets obtained with different instruments.

```
make_sig,Di,pix_no,pix_sort,xs,ys,Wi,Gp,Fp,skymap,wtmap,$
  sigmadet,sigmasky,cr_flag,n_sig=n_sig
```

A.10. Make the final map

The final section simply recalculates the sky map, so that the output sky map reflects the output gain, offset, weights, and cosmic ray flags.

```
skymap = make_map(Di,pix_no,pix_sort,xs,ys,Wi*cr_flag,Gp,Fp,wtmap)
end
```

B. Subroutines (Functions)

B.1. `get_pixel_numbers`

This routine finds the area of sky spanned by the observations and then determines the nearest pixel number within that area for each datum. The offsets (`dx`, `dy`) and rotations (`dth`, `xc`, `yc`) provided should be consistent with their use in the transformations used here.

```
function get_pixel_numbers, dx, dy, dth, xc, yc, sky_size, xn, yn, $
```

```
    xo, yo, dark_flag, rxy=rxy

if (n_elements(rxy) ne 1) then rxy = 1.0
xs = sky_size(1) & ys = sky_size(2) & zs = sky_size(3)
cz = cos(dth) & sz = sin(dth)
xsp = fltarr(xs,ys,zs)
ysp = fltarr(xs,ys,zs)
j = indgen(ys)
for k=0L,zs-1 do for i=0,xs-1 do begin
  xsp(i,*,k)= (sz(k)*(j-yc(k))+rxy*(cz(k)*(i-xc(k)) +xc(k))+dx(k))*dark_flag(k)
  ysp(i,*,k)=(-sz(k)*(i-xc(k))*rxy +cz(k)*(j-yc(k)) +yc(k) +dy(k))*dark_flag(k)
endfor
xo = min(xsp)-1
yo = min(ysp)
xsp = round(xsp-xo)
ysp = round(ysp-yo)
xn = max(xsp)+1
yn = max(ysp)+1
xo = round(xo)
yo = round(yo)

pixels = xsp+xn*ysp

return,pixels
end
```

B.2. make_map

This function constructs a skymap by taking the weighted average of the gain and offset corrected data as a function of pixel number. Ultimately it might be desirable to use a more complex mapping algorithm that makes use of the actual observation coordinates and possibly provides mapping at subpixel resolution.

```
function make_map, skies, pix_no, pix_sort, xs, ys, weights, gainmap, $
  offmap, wtmap

P = n_elements(gainmap)
skymap = fltarr(s,ys)
wtmap = fltarr(xs,ys)
starts = [0,uniq(pix_no(pix_sort))+1]
```

```
Gamma = n_elements(starts)-1

for i=0L,Gamma-1 do begin
  h = pix_sort(starts(i):starts(i+1)-1)
  skymap(pix_no(pix_sort(starts(i)))) = $
            total((skies(h)-offmap(h mod P))/gainmap(h mod P)*weights(h))
  wtmap(pix_no(pix_sort(starts(i)))) = total(weights(h))
endfor

h = where(wtmap ne 0)
skymap(h) = skymap(h)/wtmap(h)

return,skymap
end
```

B.3. make_sig

This function derives uncertainty estimates, and updates the estimates of the weights and the cosmic ray flagging.

```
pro make_sig, skies, pix_no, pix_sort, xs, ys, weights, gainmap, offmap, $
              skymap, wtmap, sigdet, sigsky, cr_flag, n_sig=n_sig

if (n_elements(n_sig) ne 1) then n_sig = 20.
s = size(skies)
P = s(1)*s(2)
starts = [0,uniq(pix_no(pix_sort))+1]
Gamma = n_elements(starts)-1
deldet2 = skies
newweight = fltarr(s(1),s(2))
skyweight = fltarr(s(1),s(2),s(3))
sigdet = fltarr(s(1),s(2))
sigsky = fltarr(xs,ys)

for k=0L,s(3)-1 do $
  deldet2(*,*,k) = skies(*,*,k) - gainmap*skymap(pix_no(*,*,k)) - offmap
deldet2 = deldet2^2

for i=0L,Gamma-1 do begin
  h = pix_sort(starts(i):starts(i+1)-1)
```

```
  if ((n_elements(h) ge 3) and (total(deldet2(h)) ne 0)) then $
     skyweight(h) = n_elements(h) / total(deldet2(h))
endfor

newweight = total(weights*cr_flag,3) / total(deldet2*weights*cr_flag,3)
h = where(gainmap lt 0.2,nh)
if (nh ge 1) then newweight(h) = newweight(h)*1e-6
for i=0L,s(3)-1 do weights(*,*,i) = newweight < (100*median(newweight))
cr_flag = (weights*deldet2 lt n_sig^2) or $
          (skyweight*deldet2 lt n_sig^2) or (skyweight eq 0)

; The rest is simply diagnostic and should have no impact on further iterations.
skyweight = 0
deltadet = skies
wtmap = fltarr(xs,ys)
for i=0L,Gamma-1 do begin
  h = pix_sort(starts(i):starts(i+1)-1)
  wtmap(pix_no(pix_sort(starts(i)))) = total(weights(h))
endfor

for k=0L,s(3)-1 do deltadet(*,*,k) = deldet2(*,*,k) * $
    wtmap(pix_no(*,*,k)) / (wtmap(pix_no(*,*,k))-weights(*,*,k)*0.99999)
deltadet = deltadet*cr_flag

sigdet = sqrt(total(deltadet,3)/s(3))
for i=0L,Gamma-1 do begin
  h = pix_sort(starts(i):starts(i+1)-1)
  sigsky(pix_no(pix_sort(starts(i)))) = total(deltadet(h))/n_elements(h)
endfor
sigsky = sqrt(sigsky)
chi2 = total(deltadet)
nu = total(cr_flag)-(2*P+Gamma)

print,''
print,'  chi^2 = ',strtrim(chi2,2),'  chi^2/nu = ',strtrim(chi2/nu,2),$
  '  nu = ',strtrim(long(nu),2),'  n_cr = ',strtrim(long(total(cr_flag eq 0)),2)
print,''
end
```

B.4. spl.pro

This is a one-dimensional spline fitting routine.

```
pro spl, d, c, u

n=n_elements(d) & c(0) = 0 & u(0) = 0.
for i=1,n-2 do begin
  p = 0.5*c(i-1)+2.
  c(i) = -0.5/p
  u(i) = d(i+1)+d(i-1)-2*d(i)
  u(i) = (3.*u(i) - 0.5*u(i-1))/p
endfor
for i=n-2,0,-1 do c(i) = c(i)*c(i+1)+u(i)
c = c/6.

end
```

B.5. spl2.pro

This function uses `spl.pro` to do two-dimensional spline fitting.

```
pro spl2, d, a, b, c

sd = size(d) & n = sd(1) & m = sd(2) & a = d*0 & b = a & c = a
u = fltarr(m)
aa = fltarr(m)
bb = fltarr(n)
cc = fltarr(n)
for i=0,n-1 do begin
  spl,d(i,*),aa,u
  a(i,*) = aa
endfor
u = fltarr(n)
for j=0,m-1 do begin
  spl,d(*,j),bb,u
  b(*,j) = bb
endfor
for j=0,m-1 do begin
  spl,a(*,j),cc,u
```

```
  c(*,j) = cc
endfor
end
```

B.6. spv2.pro

This procedure evaluates a two-dimensional spline fit at selected locations.

```
pro spv2, d, a, b, c, x, y, z
j = fix(y) & q = y-j & p=1-q & k=j+1 & i = fix(x)
dt = p*(d(i,j)-q*(p+1)*b(i,j)) + q*(d(i,k)-p*(q+1)*b(i,k))
ct = p*(a(i,j)-q*(p+1)*c(i,j)) + q*(a(i,k)-p*(q+1)*c(i,k))
i=i+1
du = p*(d(i,j)-q*(p+1)*b(i,j)) + q*(d(i,k)-p*(q+1)*b(i,k))
cu = p*(a(i,j)-q*(p+1)*c(i,j)) + q*(a(i,k)-p*(q+1)*c(i,k))
p=i-x & q=1-p
z = p*(dt-q*(p+1)*ct) + q*(du-p*(q+1)*cu)
end
```

B.7. spax.c

The function `spax.c` multiplies vector `x` by a sparse matrix `sa`. In application here, `sa` = `T`, `ija` = `[[pixels],[pixels]]`, `x` and `b` depend on the usage but have dimensions `Gamma` and `2*P` respectively, `n` = `2*P`, and `nm` = `2*NxM`.

The code is quite simple. For each element of T, `k%n` is the row (detector) number, and

Table 3. `spax.c` Variable Definitions

Variable	Definition
`sa`	array of the non-zero elements of the matrix
`ija`	array of the column numbers of the `sa` elements
`x`	vector the matrix is multiplied by
`b`	result
`n`	number of rows in the matrix
`nm`	number of `sa` elements

`ija[k]` is set to the column (pixel) number. The product of `sa` with the element of `x` from the corresponding row, is added to the appropriate row of `b`. If there were duplicate pointings of the detector, then `sa` will contain some elements that have identical indices, and the procedure here performs the summation that were neglected in the construction of `T` (§A.6).

It was necessary to modify the `sprsax` routine in "Numerical Recipes in C" (Press et al. 1992) because that routine and associated matrix storage format are designed for square matrices.

```
int spax(argc, argv)
int argc; void *argv[];
{
    long k;
    float *sa = (float *) argv[0];
    long *ija = (long *) argv[1];
    float *x = (float *) argv[2];
    float *b = (float *) argv[3];
    int n = *(int *) argv[4];
    int nm = *(int *) argv[5];
    for (k=0;k<nm;k++) b[k%n] += sa[k]*x[ija[k]];
    return(0);}
```

B.8. `sptx.c`

`sptx.c` is nearly identical to `spax.c` except that swapping the row (`k%n`) and column (`ija[k]`) indices results in multiplying the transpose of `sa` against `x`. The dimensions of `x` and `b` must also be swapped with respect to `spax` because the matrix is not square.

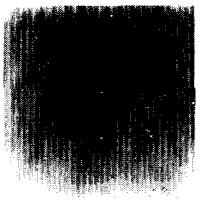
```
int sptx(argc, argv)
int argc; void *argv[];
{
    long k;
    float *sa = (float *) argv[0];
    long *ija = (long *) argv[1];
    float *x = (float *) argv[2];
    float *b = (float *) argv[3];
    int n = *(int *) argv[4];
    int nm = *(int *) argv[5];
    for (k=0;k<nm;k++) b[ija[k]] += sa[k]*x[k%n];
    return(0);}
```

Fig. 1.— The top left image shows one of ten frames of simulated WIRE data. The top center image shows the detector gains derived from the data, while the top right image shows the actual gains used to generate the simulated data. The lower left graph shows the histogram of the differences between the derived and actual gains. The bottom middle and right images show the derived sky intensities and the true sky used to generate the simulated data.

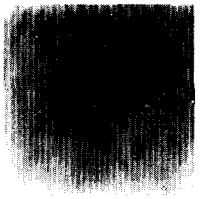
Fig. 2.— The raw data is one of the NICMOS multiaccum frames after fitting linear fits to the readouts from each pixel and removing the worst of the cosmic rays. The other pairs of derived and reference images are each shown on equivalent scales. The derived gain and offset maps only cover the upper 256×240 detectors in the 256×256 array.

Fig. 3.— The panels show six columns of the $2P \times 2P$ matrix $A^{T/2}QA^{1/2}$ for a 256×256 detector array and an idealized data set collected using a dither pattern consisting of 36 pointings evenly spaced along the sides of a Reuleaux triangle. The columns are reformated into $256 \times (256*2)$ arrays. From left to right and top to bottom the columns correspond to those containing the correlations for G^p ($p = [128, 128], [16, 128], [16, 16]$) and F^p ($p = [128, 128], [16, 128], [16, 16]$). Correlations against G^p and F^p map into the bottom and top half, respectively, of each panel. Black indicates strong positive correlations. Displayed ranges for G^pG^p, F^pF^p, and $G^pF^p = F^pG^p$ correlations are [1.5 10^{-3}, 1.55 10^{-3}], [1.2 10^{-3}, 2.0 10^{-3}], and [-8 10^{-5}, 8 10^{-5}] respectively.

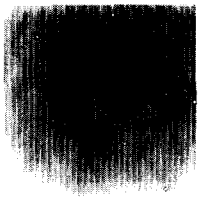
Raw Data

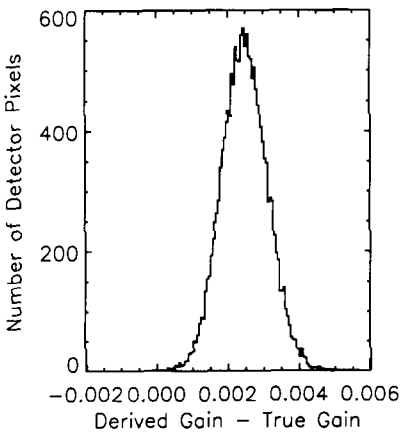


Derived Gain

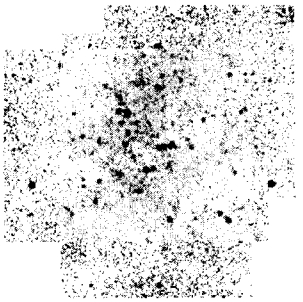


True Gain

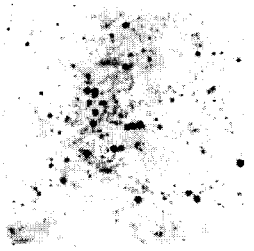




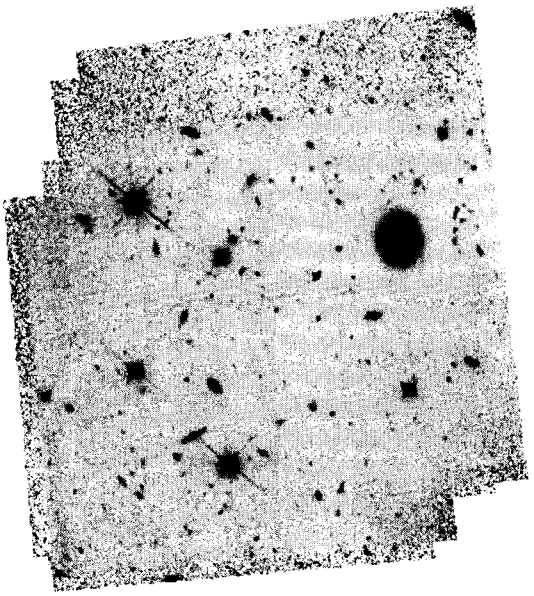
Derived Sky



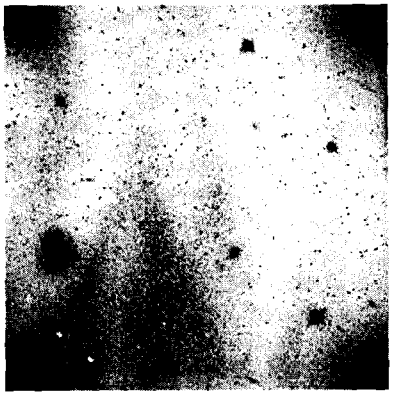
True Sky



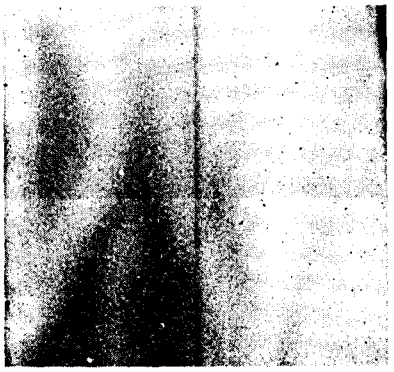
Derived Sky



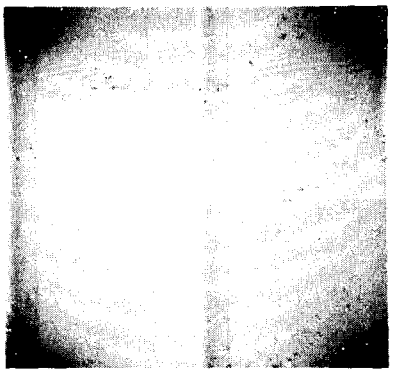
Raw Data



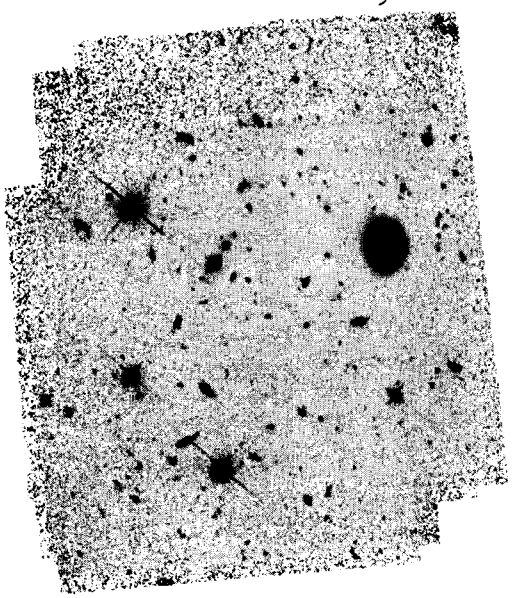
Derived Gain



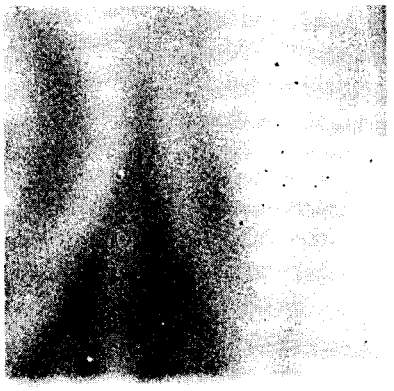
Derived Offset



Released Sky



Cal. Flat



Cal. Dark