Resampling Methods for the MTI Coregistration Product

Amy Galbraith, James Theiler, and Steve Bender

Space and Remote Sensing Sciences, Los Alamos National Laboratory, Los Alamos, NM 87545

ABSTRACT

Accurate coregistration of images from the Multispectral Thermal Imager (MTI) is needed to properly align bands for spectral analysis and physical retrievals, such as water surface temperature, land-cover classification, or small target identification. After accounting for spacecraft motion, optical distortion, and geometrical perspective, the irregularly-spaced pixels in the images must be resampled to a common grid. What constitutes an optimal resampling depends, to some extent, on the needs of the user. A good resampling trades off radiometric fidelity, contrast preservation for small objects, and cartographic accuracy – and achieves this compromise without unreasonable computational effort.

The standard MTI coregistration product originally used a weighted-area approach to achieve this irregular resampling, which generally over-smoothes the imagery and reduces the contrast of small objects. Recently, other resampling methods have been implemented to improve the final coregistered image. These methods include nearest-neighbor resampling and a tunable, distance-weighted resampling. We will discuss the pros and cons of various resampling methods applied to MTI images, and show results of comparing the contrast of small objects before and after resampling.

Keywords: MTI, multispectral, imagery, resampling

1. INTRODUCTION

1.1. Multispectral Thermal Imager

The Multispectral Thermal Imager (MTI) is a U.S. Department of Energy multispectral pushbroom sensor that collects data in fifteen spectral bands, from the visible to the thermal infrared.[1, 2] Each spectral band has its own linear detector array that is filtered to collect light in a desired spectral region. As the satellite travels, a two-dimensional image is formed for each band, with a ground sample distance (GSD) in the cross-track direction determined by the detector element size, elevation of the satellite above the ground, and off-nadir look angle; the GSD in the along-track direction also depends on the integration time, and the speed and pitch of the satellite motion. For near-nadir images, the ground sample distance is approximately 5 meters for the visible and near-infrared bands (bands A through D), and 20 meters for the midwave and thermal bands (bands E through O). The focal plane of the MTI contains three sensor chip assemblies (SCAs), each consisting of linear detector arrays for each band. The purpose of multiple SCAs is to increase the crosstrack field of view (nominally 12 km). Fig. 1 shows the layout of the MTI focal plane.

1.2. MTI Coregistration

The 48 collected images (3 SCAs, 16 line detectors) are maintained in the `LEVEL_1B_U` data product, which has been calibrated to units of radiance, but has not been registered or resampled. This is the input data product for the MTI coregistration process, which forms these images into a single multispectral image cube, with all spectral bands resampled to a common grid.[3] The MTI coregistration software uses a dead reckoning approach that maps every pixel in the `LEVEL_1B_U` data product to an absolute position on the surface of the earth, then resamples the projected pixels to a single[4] regularly-spaced grid. Resampling allows removal of distortions due to spatial nonuniformity of the optics and due to temporal nonuniformity of satellite motion. The physical separation between the detector arrays on the focal plane results in each image having a small spatial shift with respect to other bands and SCAs. In order to compensate for the physical distance between detector arrays, each band is collected at a different offset in time. A small overlap between SCAs allows later SCA-to-SCA

E-mail: {amyg,jt,sbender}@lanl.gov

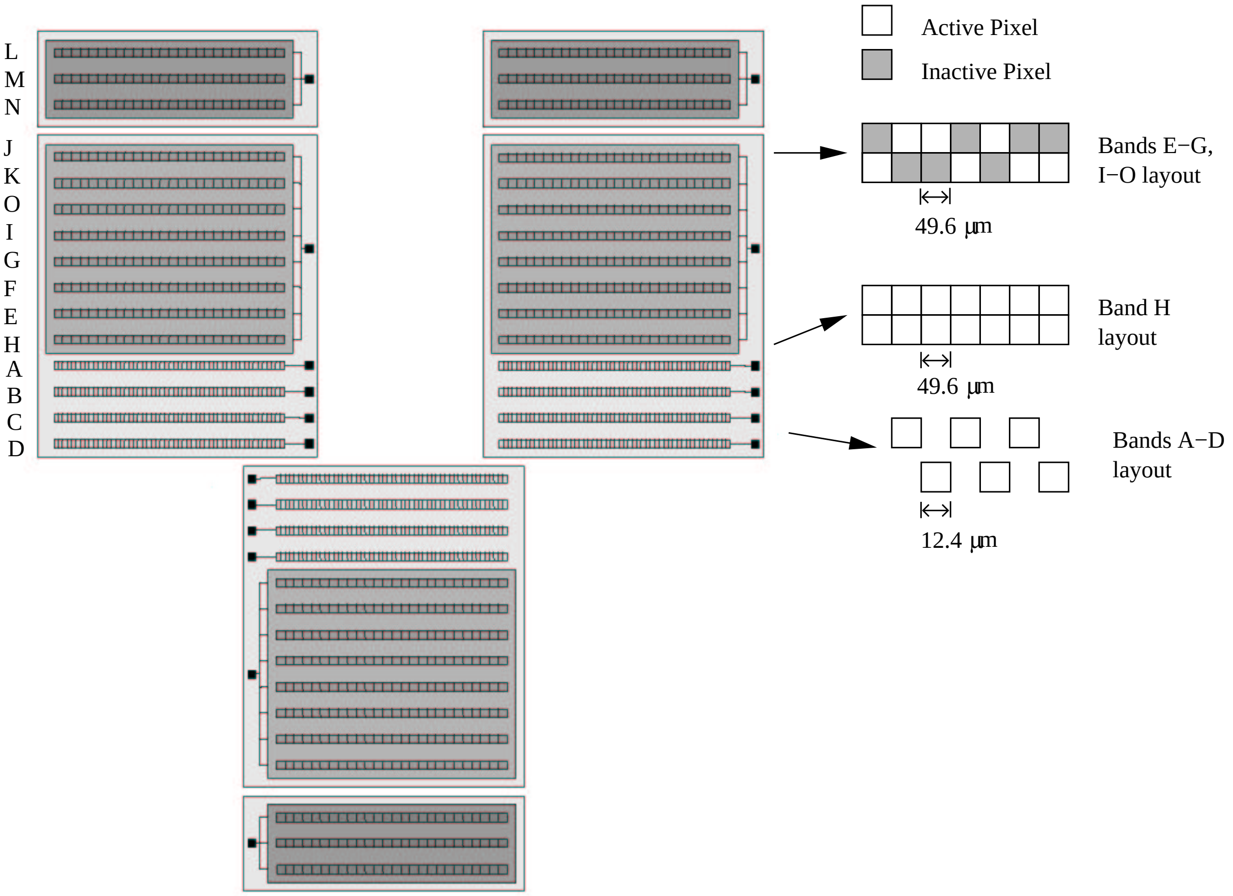


Figure 1. Layout of the MTI focal plane. The line detectors for the VNIR bands are located near the center of the focal plane where optical quality is highest. Each line detector has a delayed start time for readout, such that 48 images (3 SCAs, 16 line detectors each) are collected sequentially in time (on the order of one-half of a second). The visible/NIR bands (A-D) are arranged in a staggered pattern to reduce crosstalk and increase the signal-to-noise ratio. Also, there are redundant line detectors for bands E to O. Both line detectors for band H are read out, resulting in an H1 and an H2 band in the coregistered data product. Bands E-G and I-O also have two line detectors apiece, but active and inactive detectors are selected, resulting in only the active detectors being turned on for a given image acquisition. In practice, one complete line detector is activated to acquire each band, rather than picking and choosing detectors as shown here, but the software does have the ability to do so. The possibility of using alternate detectors to construct an image for each band is one good argument for using some form of irregular (re)sampling for the generation of coregistered MTI images.

alignment in the coregistration software as well. All of these factors result in a difficult problem to properly align and resample each band for a good reconstruction of the viewed scene. Two standard MTI data products, `LEVEL_1B_R_COREG` and `LEVEL_1B_R_GEO`, may be generated using the coregistration software; the `COREG` product is generally aligned with the orbital motion of the satellite, while the `GEO` is geo-referenced to absolute terrestrial coordinates.

Several other algorithms have been developed for coregistration of MTI images. An alternative registration algorithm that does not employ resampling, but rather, uses whole-pixel spatial translation to align the bands and SCAs, has been developed at Sandia National Laboratory for use on MTI imagery.[5, 6] In addition, a Simple Interactive Registration (SIR) product is sometimes generated by hand at Los Alamos National Laboratory. Both methods require user input, as compared to the automated coregistration method discussed here. Another automated registration method has been developed at Los Alamos, called Automated Image Registration (AIR).[7] The AIR method uses a maximum cross-correlation algorithm along with a weighted-least-squares approach to determine the optimal translations to apply to each band. The above algorithms do not incorporate knowledge of the distortions due to the optical system, spacecraft motion, *etc.*, but instead assume that translational shifts between the bands are sufficient for a registered data cube.

2. RESAMPLING METHODS

Image resampling may be defined as the reconstruction of a continuous function from observed sample points on a discrete input grid, followed by a sampling of the continuous function to a new discrete grid.[8, 9] In practice, we compute a mapping, or projection, between the input and output grid, then use interpolation to do the reconstruction and sampling. Using the direct projection approach to map pixels to the ground requires that the resampling methods used for MTI data products must be able to handle input images on an irregular grid. Distortions due to satellite motion, optical distortion, jitter, *etc.* cause slight deviations in the projection vector for each pixel in a collected image. The mapped pixel positions on the ground are almost, but not quite, regularly spaced. For display and analysis purposes, these irregular data points must be remapped to a regular grid through appropriate resampling methods. Fig. 2 illustrates the irregular spacing of points on an input *data* grid that is overlaid on a regularly-spaced, output *resampled* grid.

One straightforward approach for resampling from an irregular grid is to compute a Delaunay triangulation of the irregularly-spaced data points, and then to use linear interpolation to determine the values on the resampled grid points that are in the the interiors of the triangles. In fact, early versions of the coregistration software employed this “triangulation-gridding” method, but the approach was soon abandoned due to unreasonable memory and computation times required.[10]

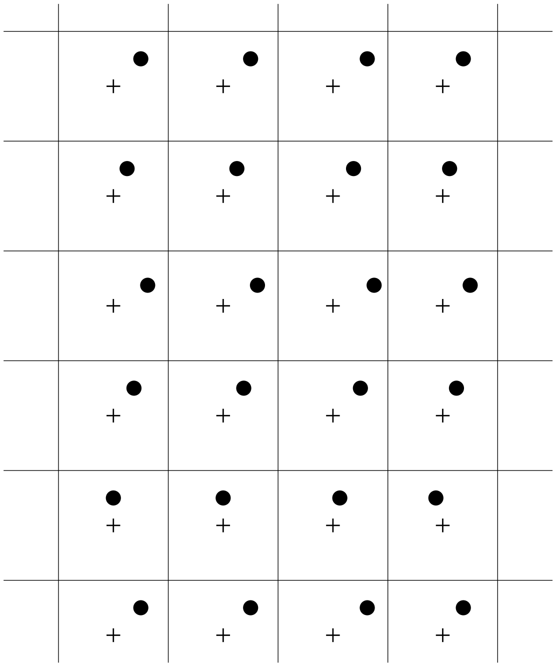
By the time of the MTI launch, a resampling method based on overlapping projected pixel areas was implemented. More recently, to address the needs of different types of users, a nearest-neighbor resampling and a distance-weighted resampling method have been added as options to the MTI registration code. The outputs of these resampled `LEVEL_1B_R_COREG` and `LEVEL_1B_R_GEO` standard data products may then be input to `LEVEL_2` algorithms as usual.

We next discuss several different approaches to image resampling, then describe our resampling algorithms in detail.

2.1. Resampling schemes

We categorize resampling algorithms into three general approaches: ideal convolution schemes, averaging schemes, and selection schemes. Each method has advantages and disadvantages associated with it.

- Ideal convolution schemes: When a good model can be found for the point spread function, and for the correct position of pixels projected to the ground, the optimal resampling schemes are based on convolution. However, these methods are sensitive to inaccuracies in those models, and in fact are somewhat more sensitive to noise in the radiance data as well. For irregularly sampled data, these schemes are also



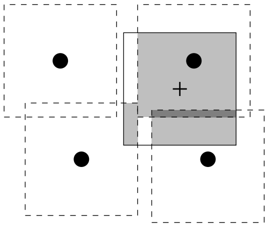


Figure 2. Left: Illustration of the resampling problem. Here, the crosses (+) are the centers of grid squares in a resampled grid on the ground. The dots (●) represent the locations on the ground corresponding to the data pixels. While the resampled grid is regular, the data grid is irregular – often, it is approximately regular, but we make no assumptions about its approximate regularity. Resampling assigns values to the crosses that are consistent with the values that have been measured at the dots. **Above:** To estimate the radiance for the resampled pixel centered at the cross (+), the area of overlap between the (solid) resampled pixel and and each of the (dashed) data pixels is computed. Here, only three data pixels overlap the resampled pixel, and so the value of the resampled pixel will be a weighted average of those three with the most of the weight assigned to the upper right data pixel.

more complicated and more expensive. The optimal function for sampled image interpolation is the sinc function,[8, 11, 12] with its one-dimensional form given by

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \tag{1}$$

Ideal sampling requires that all frequencies present in the sampled signal are preserved (no blurring occurs), so a rect function is needed in the Fourier domain. From linear systems theory, multiplication with a rect function in Fourier space is equivalent to convolution with a sinc function at each sampling location in the spatial domain.[13] Note that in our case, we do not assume regular spacing of our sample points, therefore normal interpretations of proper Nyquist sampling are not valid. However, any function of the form

$$f(x) = \sum_i \mathrm{sinc}(2f_o(x - x_i))\ z_i \tag{2}$$

will have no power beyond frequency f_o, since the sinc function has no power beyond that frequency. If we are given $f(x_i)$ for each sample location x_i, then we have a situation with N equations and N unknowns.

$$f(x_j) = \sum_i K_{ji} z_i \tag{3}$$

where $K_{ji} = \mathrm{sinc}(2f_o(x_j - x_i))$; Eq. (3) can be used to solve for the coefficients z_i. But in general, the solution to Eq. (3) involves an $N \times N$ matrix decomposition. And, since the sinc function is of infinite extent, the matrix will not be sparse. Rather than resort to direct inversion of the $N \times N$ matrix to solve for the z_i's, iterative schemes, such as projection onto convex sets (POCS)[14] or conjugate gradient[15] algorithms, are often employed; further details, especially for the case of irregular sampling, may be found in the literature.[16] Note, however, that if x_i is regularly sampled (*i.e.,* $x_i = x_o + i\Delta x$), and $f_o \Delta x = 1$, then the fact that $\mathrm{sinc}(x) = 0$ for integer x leads to $z_i = f(x_i)$.

Implementing ideal convolution is prohibitively slow due to the large neighborhood required around each input pixel location. In practice, a truncated approximation to a sinc function is used, such as cubic splines (called cubic convolution, or third-order hold interpolation) or quadratic functions (second-order hold interpolation).[8, 15] The main lobe and first sidelobes of the sinc function are typically used. Inclusion of the negative sidelobes has important implications for the resampled image; pixel values are not constrained to their original range, which can cause ringing, or overshoot, in the image radiance values.[11]

- Averaging schemes: A more robust resampling scheme is based on the principle of averaging neighboring pixels. This guarantees that a resampled pixel won't have a radiance value that is any larger or any smaller than the actually measured radiances in the region of that pixel. This has as a nominal consequence that the data will be "smoothed" – a small bright object in the scene, even if it is picked up as a bright pixel in the calibrated but unregistered `LEVEL_1B_U` data, will be "smeared out" in the resampled image.

  The area-weighted and distance-weighted resampling methods that we will discuss in more detail in the subsequent sections are both averaging schemes. Bilinear resampling is a simple method that fits in this category as well, though it is not well adapted for irregular data grids. The inefficient triangular mesh approach, described above, is also an averaging scheme.

- Selection schemes: A special case of an averaging scheme is one in which each resampled pixel is taken to be the same value as one of the pixels in the raw image data. Although spatial fidelity is compromised in this scheme, the smoothing effects of averaging are ameliorated. Smith *et al.*[5,6] has advocated this approach with a whole-pixel translation-only registration algorithm. However, this method does not deal with optical distortions, nonuniform motion of the satellite, *etc.*

With the different schemes taken into consideration, we implemented two averaging methods (area-weighted resampling and distance-weighted resampling) as well as a selection scheme (nearest-neighbor resampling). The desire for a more computationally efficient resampling than our initial triangulation-regridding approach, along with robustness requirements, led to the following algorithms, which we discuss in Sec. 2.2 through 2.4.

2.2. Area-weighted Resampling

Area-weighted resampling is an averaging scheme in which each point in the resampled grid is given a value that corresponds to the weighted average of points in the data grid whose "support" overlaps that of the resampled grid point. The weight, in this weighted average, is proportional to the area of this overlap. This is illustrated in the top right panel in Fig. 2. Although conceptually straightforward, some practical issues need to be addressed.

We comment that computing the area of overlap becomes a lot more complicated when the axes of the support squares are not aligned with the axes of the resampling grid. The automatic resampling scheme has two modes: "orb-aligned" and "geo-aligned." In the orb-aligned mode, the axes of the resampling grid are aligned with the orbit of the satellite over the ground. (See Ref. [3] for a fuller explanation.) But in the geo-aligned mode, the axes of the resampling grid are aligned with local latitude and longitude. An approximation is made in that case, which treats the individual pixel supports as if they were aligned. For the georeferenced product (`LEVEL1B_R_GEO`), clearly a geo-aligned mode is in effect.

Depending on the size of the resampled pixels, and the irregularity of the data pixels, it is possible for this method to leave some of the resampled pixels undetermined. To ensure that this does not happen, we increase the "support" of the data pixels by a fixed percentage – in practice, we find that 25% is adequate. We choose this percentage as small as possible because a large increase in the support will lead to a greater blurring of the resampled image.

Finally, we note that there is a potential computational difficulty in finding the data points that are nearest to each of the resampled pixel locations. The problem is that the data points are irregularly spaced, so their positions are not *a priori* predictable. A naive algorithm would have to check each data point to see if it overlapped a given resampling point, and that would lead to an $O(N^2)$ algorithm if there were N data points and N resampled pixels. Sophisticated search algorithms, based for instance on a k-d tree, can reduce this to $O(N \log N)$, but the software is nontrivial to implement. It is tempting to take advantage of the approximate regularity in the data positions, but this is dangerous, because it is hard to know how much regularity can be safely assumed.

There turns out to be a fairly simple solution. It is difficult, given a resampled pixel, to identify from an irregular grid the points that are close enough to the resampled pixel to overlap with it. But, given one of the

```
1. For each data pixel $i$, with position $\mathbf{x}_i$ and radiance $r_i$:
    (a) Find the point $j_i$ in the resampled grid which is closest to the $i$'th data pixel. Since the resampled grid is regular, this is straightforward; it is the pixel on the resampled grid which is at the "rounded off" value of the data pixel's position: $\mathbf{x}_j = \Delta x \lfloor \mathbf{x}_i / \Delta x + 0.5 \rfloor$.
    (b) Define the set $J_i = \{ j \mid |\mathbf{x}_{j_i} - \mathbf{x}_j| \leq n \}$ where $n$ is some number of pixels (often $n = 1$) which describes the size of a square neighborhood of points in the resampled grid that are near enough to the data pixel $i$ so that $A_{ij}$, the area of overlap between the $i$th data pixel and the $j$th resample pixel, might be nonzero.
    (c) For each one of these resampling pixels $j \in J_i$:
        i. Compute the area overlap $A_{ij}$ between the $i$th data pixel and the $j$th resample pixel.
        ii. Update for this resample pixel $j$ a running sum of areas $A_{ij}$
        iii. Update for this resample pixel $j$ a running sum of the product $A_{ij} r_i$.
2. For each resample pixel $j$:
    (a) Compute the ratio of the two running sums: $\sum_i A_{ij} r_i / \sum_i A_{ij}$
    (b) This ratio is the estimated radiance for the $j$th resample pixel.
```

Figure 3. Pseudo-code for the area-weighted resampling algorithm. Although it would be more natural for the outer loop to be over the resample pixels j, it would be more expensive because it is difficult to find the neighbors $I_j = \{i \mid A_{ij} \neq 0\}$ when the data pixels i are irregularly placed. The advantage of this approach is that in Step 1(a), the set J_i is straightforward to find because the resample pixels j are regularly gridded.

data points, it is not difficult to find the regular resampled pixels with which it overlaps. Thus, although it takes a little extra (just $O(N)$) memory, it is more efficient for the algorithm to work in what at first seems the backward way, as shown in Fig. 3.

Recently, other resampling methods have been implemented to improve the final coregistered image with respect to over-smoothing and contrast reduction by the area-weighted resampling algorithm.

2.3. Distance-weighted Resampling

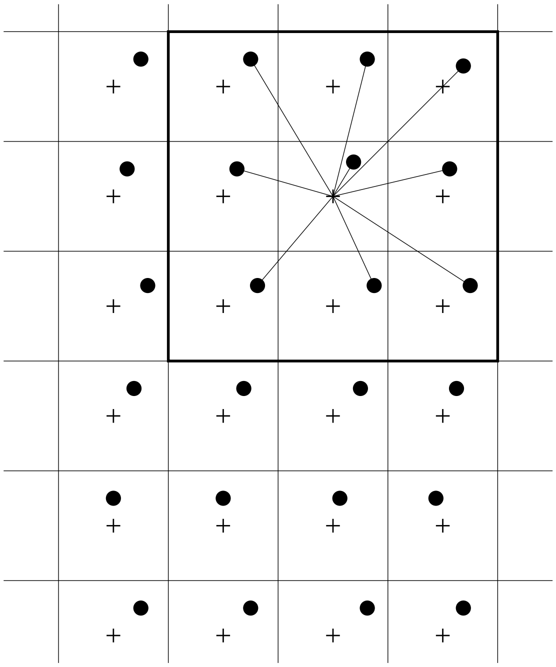
A tunable distance-weighted resampling method has been implemented which computes the value of each output pixel as a linear combination of input data points that are weighted according to their distance d from the output pixel center.

Looping over the input data grid, we first find the Euclidean distance d from a data pixel i to the center of each of its neighboring resample pixels j. As in the area-weighted resampling case, we map from the irregular input grid to the output grid because the neighbors may be found with an $O(N)$ algorithm. The number N of output neighbors can be selected with a support region. In practice, we found that a 5 by 5 region, or $N = 25$, was sufficiently large for resampling without requiring unreasonable computing time, and is used by default.

As each distance value is found, we compute a weight w_{ij} using a Gaussian-shaped function with tunable width, or, standard deviation σ,

$$w_{ij} = e^{-{d_{ij}}^2/\sigma^2} \quad . \tag{4}$$

(The standard scaling used for a Gaussian probability distribution[17] is not used because we later renormalize, and it saves computation time.) After multiplying the current data pixel radiance r_i by the weight w_{ij}, we add this weighted value to pixel i's neighboring output pixel bins. Each output pixel will have a running sum of weighted radiances.



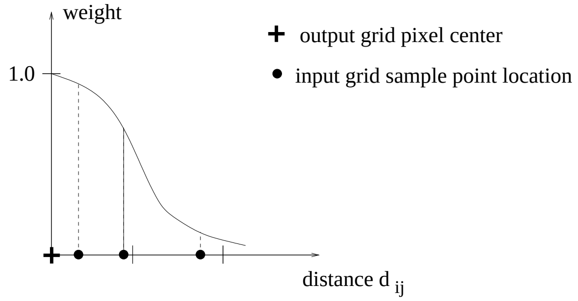

weight
1.0
+ output grid pixel center
● input grid sample point location
distance d ij

Figure 4. Left: Illustration of distance-weighted resampling. As in Fig. 2, the crosses (+) are the centers of grid squares in a resampled, regular grid, and the dots (•) represent the locations on the ground corresponding to the data pixels. The radiance for the resampled pixel centered at the cross (+) is a weighted average of the input values at the locations given by the dots (•) inside a local neighborhood with N neighbors. For this example, $N = 9$. **Above:** One-dimensional slice showing Gaussian weighting used in the distance-weighted resampling method. The values at the dot locations closer to the cross will be more heavily weighted than values at dot locations that are far from the cross. Exponential weighting does not fall off as rapidly as Gaussian weighting, and may be used to prevent underflow errors that cause non-filled pixels.

Once we loop over every data pixel, the final radiance for each resample pixel is computed by dividing the weighted input radiances by the total weights applied to that output bin. This normalizes the values back to units of radiance:

$$r_j = \frac{\sum_{i=1}^{N} \left(e^{-d_{ij}{}^2/\sigma^2} \;\; r_i \right)}{\sum_{i=1}^{N} e^{-d_{ij}{}^2/\sigma^2}} \; . \tag{5}$$

The concept of finding the distances to each resample pixel in the output grid is illustrated in Fig. 4. The weight assigned to each positive distance d_{ij} can take values between 0 and 1. The width of the weighting function may be modified with a parameter σ. The distance-weighted resampling scheme becomes, in a limiting case, a selection scheme. As σ gets small, contributions from the majority of the neighboring pixels approach zero, leaving the closest neighboring pixel to provide nearly the entire contribution to the output pixel. The resulting output is much like nearest-neighbor resampling, described subsequently in Sec. 2.4. Larger values of σ provide more smoothing, and produce images similar to those from applying the area-weighted resampling. Pseudo-code for the distance-weighted method follows that approach for area-weighted resampling in Fig. 3 with the difference that w_{ij} is used in place of A_{ij}.

2.4. Nearest-neighbor Resampling

Radiometric fidelity is sometimes the most important criterion for a user of MTI imagery. To address the needs of these users, we implemented the standard nearest-neighbor resampling algorithm. Rather than average or otherwise combine input data points, pixels in the resample grid are assigned the unaltered value of the "nearest" input data point measured by Euclidean distance. This results in improved radiometric accuracy at the expense of spatial location accuracy. Nearest-neighbor resampling can introduce stair-step errors of up to a one-half pixel shift, which can be quite noticeable along diagonal edges. For this reason, nearest neighbor resampling may not be best for visual interpretation tasks. Another type of artifact that may occur is related to the fact that each band image in the 15-band MTI image cube is collected by a separate line detector. When the pixels are shifted in each band by the nearest-neighbor resampling, different areas (and thus different physical materials) on the ground can be remapped to the same pixel. This "mispairing" of bands[18] can give an incorrect spectrum at that pixel location because different materials appear in each band rather than a smooth spectral mixture. Color display of multiple bands will have blocky colors at material boundaries.

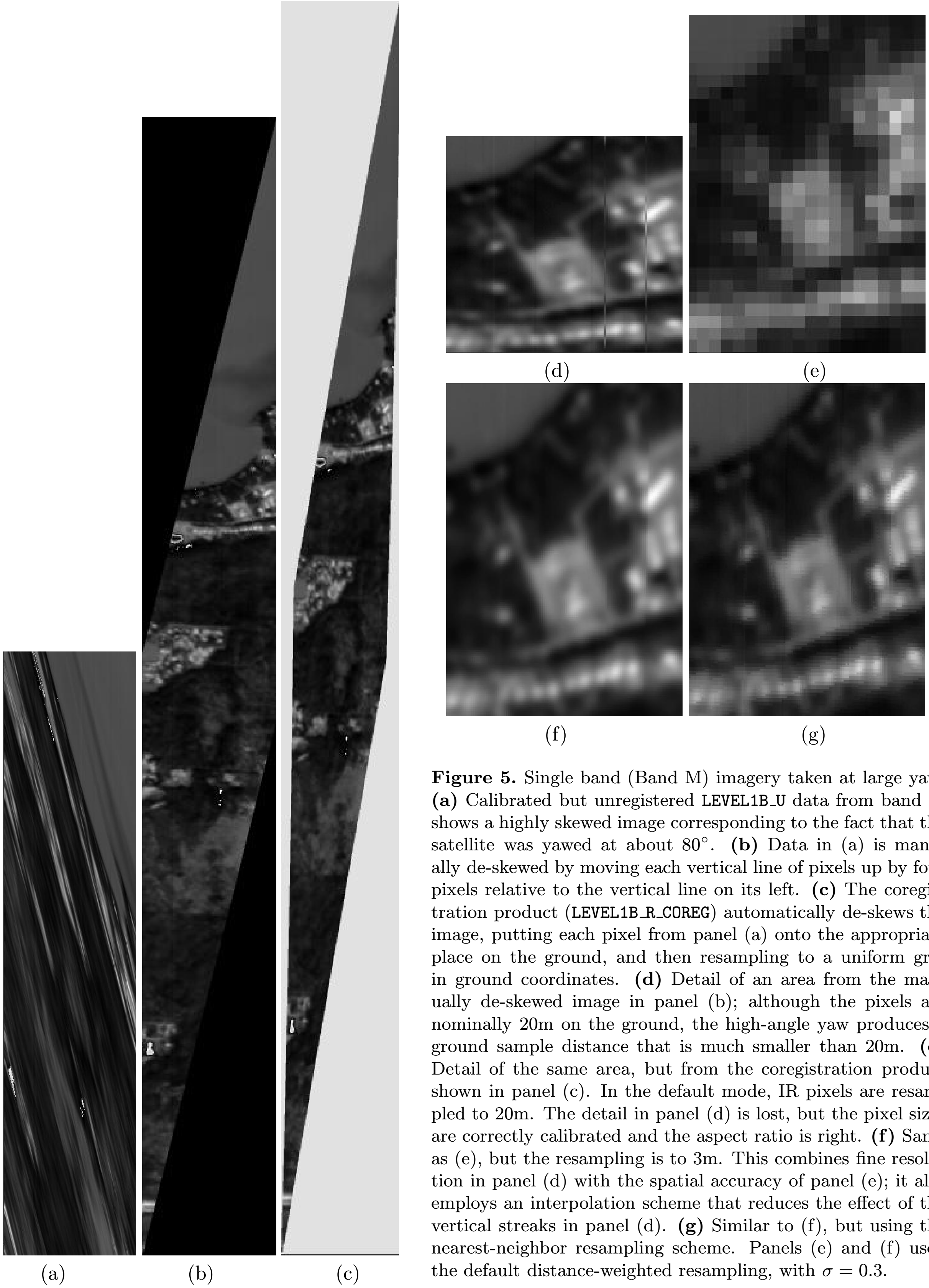


Figure 5. Single band (Band M) imagery taken at large yaw. **(a)** Calibrated but unregistered `LEVEL1B_U` data from band M shows a highly skewed image corresponding to the fact that the satellite was yawed at about 80°. **(b)** Data in (a) is manually de-skewed by moving each vertical line of pixels up by four pixels relative to the vertical line on its left. **(c)** The coregistration product (`LEVEL1B_R_COREG`) automatically de-skews the image, putting each pixel from panel (a) onto the appropriate place on the ground, and then resampling to a uniform grid in ground coordinates. **(d)** Detail of an area from the manually de-skewed image in panel (b); although the pixels are nominally 20m on the ground, the high-angle yaw produces a ground sample distance that is much smaller than 20m. **(e)** Detail of the same area, but from the coregistration product shown in panel (c). In the default mode, IR pixels are resampled to 20m. The detail in panel (d) is lost, but the pixel sizes are correctly calibrated and the aspect ratio is right. **(f)** Same as (e), but the resampling is to 3m. This combines fine resolution in panel (d) with the spatial accuracy of panel (e); it also employs an interpolation scheme that reduces the effect of the vertical streaks in panel (d). **(g)** Similar to (f), but using the nearest-neighbor resampling scheme. Panels (e) and (f) used the default distance-weighted resampling, with $\sigma = 0.3$.

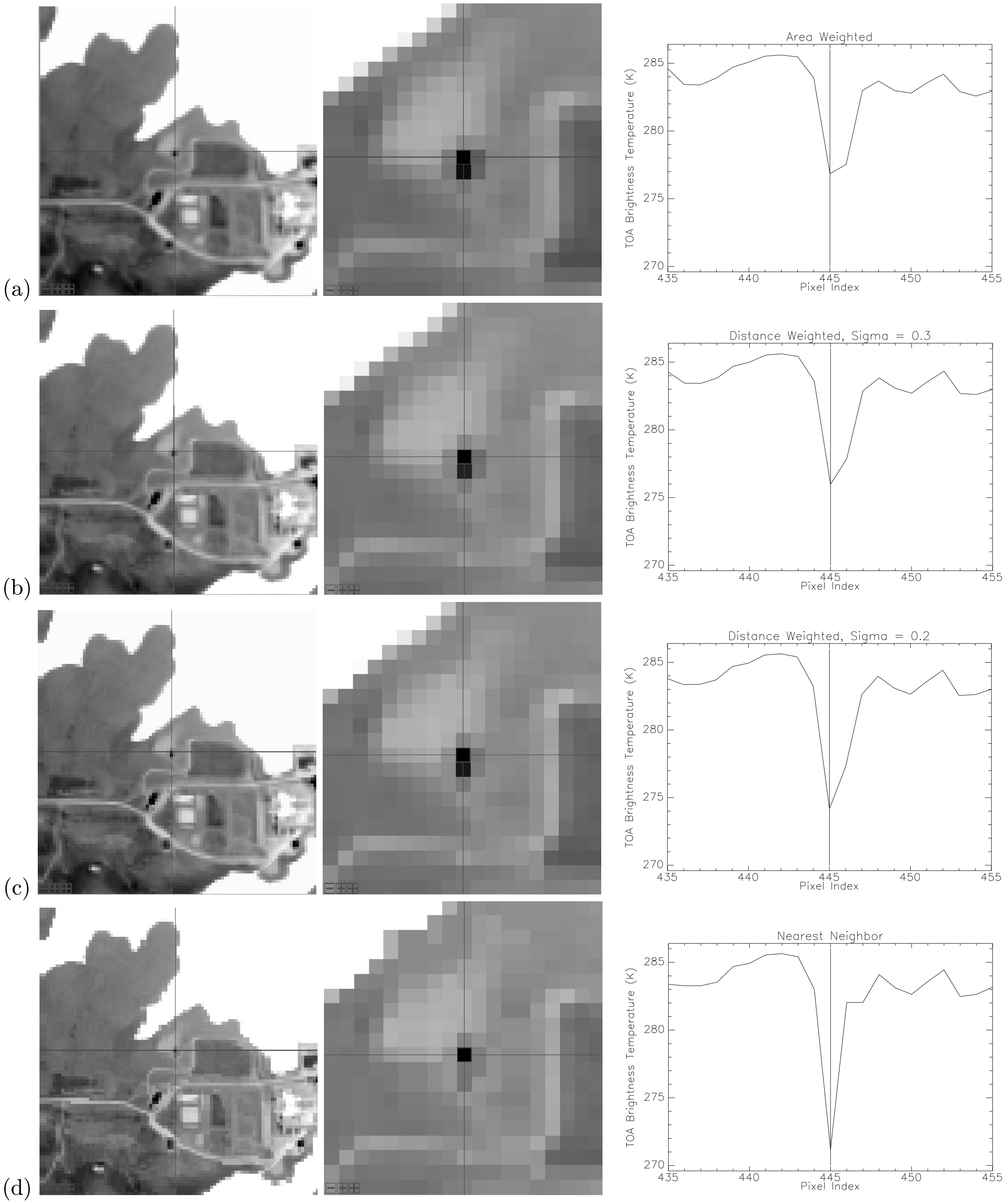


Figure 6. Contrast of a small target using different resampling methods and parameters: (a) area-weighted, (b) distance-weighted with $\sigma = 0.3$, (c) distance weighted with $\sigma = 0.2$, (d) nearest neighbor resampling. A zoomed portion of each image on the left is shown in the center. A vertical profile through each zoomed image clearly shows the increase of contrast from area-weighted, to distance-weighted, to nearest neighbor resampling. Note, however, the stair-step errors along diagonal edges that are present in the nearest-neighbor example. These errors can be up to a one-half pixel shift from the true location.

3. RESAMPLING OF LARGE YAW IMAGES

The LEVEL1B_R_COREG data product can be resampled at any user-specified grid size. For instance, rather than sample the visible bands to 5m (close to actual sampling for most typical nadir looks), the coreg can produce 20m data to more efficiently align with the 20m IR bands. Alternatively, the IR bands can be remapped to 5m to match the higher resolution of the visible bands.

An unusual spacecraft collect demonstrates another application of this ability to alter grid sizes. The satellite flew at an 80 degree yaw, so that the cross-track ground sample distance would be significantly compressed, and the motion of the pitch and roll were controlled so that the satellite boresight would sweep by the ground much more slowly than is usually the case, thereby compressing the along-track ground sample distance.

The effect is to produce a highly skewed LEVEL1B_U image, as seen in Fig. 5(a). However, when this image is de-skewed, it results in a much higher resolution image than is usually possible from a satellite with a 20m nominal GSD. Fig. 5(b-g) shows various resampling methods applied to this image that was acquired at a large yaw angle. By simply resetting the resampled pixel size to 3m, the ordinary LEVEL1B_R_COREG product is able to capture the high resolution that this unusual flight path provided.

4. CONTRAST OF SMALL OBJECTS

Radiometric contrast can be an important factor when resampling images that contain small objects of interest. The effects of resampling on a small object in an image are shown in Fig. 6. As noted in previous sections, the area-weighted and distance-weighted resampling methods introduce some blurring, but the nearest neighbor resampling can introduce staircase artifacts along edges.[12] The loss of contrast due to use of averaging-based resampling methods is due to both the type of averaging and the relative location of the viewed object with respect to the regular output grid. The relative location of the sample grid and a target is called *sample-scene phase*,[11] and affects the measured contrast between the object and its background in a sampled image. When *re*sampling an image, a similar spatial phase effect occurs; contrast can be reduced if the input pixel is split between two (or more) output pixels.

5. DISCUSSION

What, then, is the best method to coregister images? Under ideal situations, certain convolution methods can be shown to be optimal. However, these methods are susceptible to overshoot and ringing artifacts. Furthermore, for irregular sampling, these methods are computationally expensive.

Both nearest-neighbor and averaging-based resampling are more robust, but also have drawbacks, and trade off spatial fidelity for reduced contrast. For analysis of small objects, nearest-neighbor resampling is best when viewing only a single band. When viewing multiple bands, the distance-weighted and area-weighted resampling generally have smaller spatial shifts; for the distance-weighted method, one can use σ as a tuning parameter to regulate this trade-off. The optimal value of σ will depend on the viewing geometry for each scene. The ability to adjust σ is an advantage of the distance-weighted resampling, compared to the area-weighted resampling algorithm, which generally over-smoothes. Finally, we point out that even for panchromatic imagery, resampling is sometimes unavoidable; change detection studies employ multiple images taken at different times coregistered to within subpixel accuracy and then resampled.[19, 20] Distance-weighted resampling is the recommended method to use on MTI imagery for this application.

ACKNOWLEDGMENTS

We wish to acknowledge the generous support of the entire MTI team, and also extend special thanks to Paul Pope for helpful discussions on registration and change detection, and to Kim Pollack, Keri Ramsey, and Kathy Smith for their help with various software issues and other useful conversations. We also acknowledge the ingenuity of the Sandia National Laboratory MTI team for the design and execution of the unique large yaw oversampling maneuver. This work was supported by the U.S. Department of Energy.

REFERENCES

1. J. J. Szymanski, W. Atkins, L. Balick, C. C. Borel, W. B. Clodius, W. Christensen, A. B. Davis, J. C. Echohawk, A. Galbraith, K. Hirsch, J. B. Krone, C. Little, P. Maclachlan, A. Morrison, K. Pollock, P. Pope, C. Novak, K. Ramsey, E. Riddle, C. Rohde, D. Roussel-Dupré, B. W. Smith, K. Smith, K. Starkovich, J. Theiler, and P. G. Weber, "MTI science, data products and ground data processing overview," *Proc SPIE* **4381**, pp. 195–203, 2001.
2. P. G. Weber, B. C. Brock, A. J. Garrett, B. W. Smith, C. C. Borel, W. B. Clodius, S. C. Bender, R. R. Kay, and M. L. Decker, "Multispectral Thermal Imager mission overview," *Proc SPIE* **3750**, pp. 340–346, 1999.
3. J. Theiler, A. Galbraith, P. Pope, K. Ramsey, and J. Szymanski, "Automated coregistration of MTI spectral bands," *Proc SPIE* **4725**, pp. 314–327, 2002.
4. Actually the visible bands are usually resampled to a higher-resolution (nominally, 5m) grid that is nested within the lower-resolution (nominally, 20m) grid used for the infrared bands.
5. J. Smith, S. Motomatsu, J. G. Taylor, K. J. Jefferson, and B. R. Stallard, "MTI translation-only registration," in *Multispectral Thermal Imager (MTI) Symposium*, U.S. Department of Energy Office of Nonproliferation and National Security, Albuquerque, NM, 12-13 March 2001.
6. J. L. Smith, S. E. Motomatsu, J. G. Taylor, K. J. Jefferson, and B. R. Stallard, "Semi-autonomous registration of satellite imagery using feature fitting," *Proc SPIE* **4381**, pp. 447–454, 2001.
7. P. Pope and J. Theiler, "Automated image registration (AIR) of MTI imagery," *Proc. SPIE* **5093**, 2003. (this volume).
8. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, 1990.
9. J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction, Second Ed.*, Springer-Verlag, 1993.
10. We employed the routing `TRIGRID`, implemented in Interactive Data Language (IDL) by Research Systems Incorporated. More details exist in the IDL documentation.
11. R. A. Schowengerdt, *Remote Sensing, Models and Methods for Image Processing*, Academic Press, 1997.
12. J. R. Schott, *Remote Sensing: the Image Chain Approach*, Oxford University Press, 1997.
13. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, John Wiley & Sons, Inc., 1978.
14. P. L. Combettes, "The foundations of set theoretic estimation," *Proceedings of the IEEE* **81**, pp. 182–208, Feb. 1993.
15. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
16. F. Marvasti, ed., *Nonuniform Sampling: Theory and Practice*, Kluwer Academic/Plenum Publishers, New York, 2001.
17. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Inc., 3rd ed., 1991.
18. G. K. Arp, "Eye on Quality: Points of View," http://www.imagingnotes.com/winter03/eoq.htm (Apr. 12, 2003).
19. J. R. G. Townshend, C. O. Justice, C. Gurney, and J. McManus, "The impact of misregistration on change detection," *IEEE Trans. Geosci. Remote Sensing* **30**, pp. 1054–1060, Sept. 1992.
20. D. Xiaolong and S. Khorram, "The effects of image misregistration on the accuracy of remotely sensed change detection," *IEEE Trans. Geosci. Remote Sensing* **36**, pp. 1566–1577, Sept. 1998.