

Image navigation and registration performance assessment tool set for the GOES-R Advanced Baseline Imager and Geostationary Lightning Mapper

Frank J. De Luccia^{*a}, Scott Houchin^a, Brian C. Porter^a, Justin Graybill^a, Evan Haas^a, Patrick D. Johnson^a, Peter J. Isaacson^a, Alan D. Reth^b

^a The Aerospace Corporation, 2310 El Segundo Blvd, El Segundo, CA 90245-4609;

^b Chesapeake Aerospace, LLC, PO Box 436, Grasonville, MD 21638-0436

ABSTRACT

The GOES-R Flight Project has developed an Image Navigation and Registration (INR) Performance Assessment Tool Set (IPATS) for measuring Advanced Baseline Imager (ABI) and Geostationary Lightning Mapper (GLM) INR performance metrics in the post-launch period for performance evaluation and long term monitoring. For ABI, these metrics are the 3-sigma errors in navigation (NAV), channel-to-channel registration (CCR), frame-to-frame registration (FFR), swath-to-swath registration (SSR), and within frame registration (WIFR) for the Level 1B image products. For GLM, the single metric of interest is the 3-sigma error in the navigation of background images (GLM NAV) used by the system to navigate lightning strikes. 3-sigma errors are estimates of the 99.73rd percentile of the errors accumulated over a 24 hour data collection period. IPATS utilizes a modular algorithmic design to allow user selection of data processing sequences optimized for generation of each INR metric. This novel modular approach minimizes duplication of common processing elements, thereby maximizing code efficiency and speed. Fast processing is essential given the large number of sub-image registrations required to generate INR metrics for the many images produced over a 24 hour evaluation period. Another aspect of the IPATS design that vastly reduces execution time is the off-line propagation of Landsat based truth images to the fixed grid coordinates system for each of the three GOES-R satellite locations, operational East and West and initial checkout locations. This paper describes the algorithmic design and implementation of IPATS and provides preliminary test results.

Keywords: Image Registration, Image Navigation, Remote Sensing, Satellite

1. INTRODUCTION

The Image Navigation and Registration (INR) Performance Assessment Tool Set (IPATS) was developed by the GOES-R Flight Project to provide a capability for evaluation of post launch INR performance of the Advanced Baseline Imager (ABI) and the Geostationary Lightning Mapper (GLM). In the case of ABI, the Level 1B imagery products are of three types, full disk (FD), continental US (CONUS), and Mesoscale images that can be tasked to any local region of the FD. The CONUS images have a spatial extent of 5000 km EW by 3000 km NS, and the Mesoscale images have a spatial extent equivalent to 1000 km EW by 1000 km NS at nadir. These images are resampled to a fixed grid (FG) of East-West (EW) and North-South (NS) angular coordinates defined with respect to the ideal satellite orbital location.¹ The sampling interval is uniform in both the EW and NS directions and is equal to 14 μ rad, 28 μ rad or 56 μ rad, depending upon the spectral band. INR performance is specified for five metrics, (a) navigation (NAV) error, the difference between the location of a pixel on the fixed grid and its true location, (b) frame-to-frame registration (FFR) error, the difference in navigation errors for corresponding pixels in two consecutive images in the same band, (c) within frame registration (WIFR), the difference between the radial angular separation of two pixels on the fixed grid and their true radial angular separation, (d) swath-to-swath registration (SSR), the relative location error of two adjacent pixels lying on opposite sides of an image swath boundary in the same band, and (e) channel-to-channel registration (CCR), the difference in navigation errors for corresponding pixels in two difference spectral bands in the same frame.² In the case of GLM a navigated background image, an intermediate Level 1 beta product, is used to navigate the lightning events that constitute the L1B products from this instrument.³ Locations of pixels in GLM background images are navigated to the same EW and NS angles used for the ABI fixed grid, but these locations are not uniformly spaced as they are for ABI. The metric of interest for a GLM background image is the navigation error of its pixels.

IPATS employs image registration techniques to evaluate all of these ABI and GLM metrics. For ABI NAV, IPATS utilizes Landsat images projected to the fixed grid as truth data. These Landsat images, known as Ground Control Point

(GCP) chips or simply “chips”, are of limited spatial extent and are selected to include the landmarks⁴ used by the GOES-R ground system’s Product Monitor (PM) as well as many other high contrast local regions on the earth. When the portion of an ABI image overlapping a Landsat chip is cloud free or nearly so, IPATS finds the shifted position of the ABI sub-image that best correlates with the Landsat chip to estimate navigation error. GLM navigation error is estimated in an analogous way, except that the ABI images closest in time and spectral band provide the truth data. ABI WIFR error is derived from the estimated navigation errors of pairs of ABI sub-images that overlap different Landsat chips. To evaluate FFR, CCR and SSR, IPATS utilizes user specified evaluation regions or windows located on a uniform grid as well as in the neighborhood of Landmarks. IPATS crops out sub-images of the two ABI images appropriate for estimating the metric of interest based on the specified evaluation regions and finds the translated or “shifted” position of one sub-image relative to the other that provides the best correlation. The shift corresponding to the maximum or “peak” correlation provides the estimate of relative navigation error. IPATS INR metric estimates are aggregated to generate error statistics for each single image and for all the images acquired in a specified time period. ABI INR requirements are specified in terms of the 3-sigma error, defined as 99.73rd percentile of the errors accumulated over each 24 hour period from local noon to local noon.

All ABI INR metrics except SSR are generated in the normal operational mode without interaction with Mission Operations. A Mesoscale image consists of exactly two swaths swept out by two scans of the ABI. To evaluate SSR, two successive Mesoscale images are tasked by Mission Operations to Earth locations such that the bottom of the lower swath of the first Mesoscale image overlaps the top of the upper swath of the second Mesoscale image. Evaluation regions are chosen that lie entirely within the overlap region of the two tasked Mesoscale images. The sub-image registration errors measured within the overlap region then provide measures of relative navigation error of adjacent pixels on opposite sides of swath boundary.

In the subsequent sections of this paper we describe the algorithmic approach employed by IPATS to calculate INR metrics and statistics, the software implementation of IPATS, and preliminary test results using ABI proxy images developed by the GOES-R program.

2. ALGORITHMIC APPROACH

2.1. Modular design

As indicated in the Introduction, correlations between image pairs lie at the heart of the processing required to generate any of the INR metrics with the exception of WIFR, which is derived from NAV error measurements. However, there are multiple algorithms available for performing image correlation. One such algorithm utilizes the Pearson correlation coefficient (CC) and another utilizes Normalized Mutual Information⁵ (NMI), and both of these algorithms have been incorporated in IPATS as algorithmic alternatives for performing the crucial image correlation step. Similarly, there are alternative algorithms employed for image preprocessing prior to correlation, such as the interpolation method used in resampling of images and filtering methods for edge enhancement. The processing paths for the INR metrics are not unique, and it is desirable to make available to the IPATS user alternative processing paths for optimization of IPATS performance when the on-orbit data become available. Furthermore, some processing steps are common across all metrics, such as the resampling of the images to the resolution scale at which the correlation is performed. The desire to incorporate alternative processing paths for INR metrics and to isolate and address common processing steps in the same way across these metrics have led us to a modular approach in the design of IPATS algorithms.

Figure1 depicts the IPATS algorithmic concept at top level. The processing for each metric, depicted in the top diagram in the figure, is subdivided into a chain of more elementary steps or algorithmic “modules” that perform similar functions for any metric, as depicted in the bottom diagram. The user selects the metric, a specific algorithm for each module and other configuration parameters that determine the precise processing to be performed by IPATS. The algorithms available for selection are described in subsequent subsections.

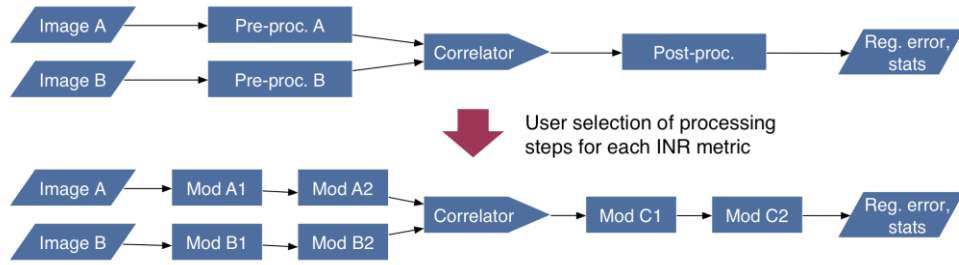


Figure1. Decomposition of processing for each INR metric into algorithmic components allows the user to select different processing paths for IPATS performance optimization, as well as isolating common algorithmic components for optimized design and implementation.

2.2. Image preprocessing

Image preprocessing consists of the following algorithmic modules:

- **Sub-image pair identification.** For ABI NAV and for a given ABI image under test, each Landsat chip of the corresponding band drawn from a database described in Section 3.3 determines the sub-image of the ABI image to be correlated with the Landsat chip, assuming the ABI image and Landsat chip sufficiently overlap one another. For all other ABI metrics (except WIFR) a user supplied array of evaluation windows is used to identify the sub-images of the two ABI images under test to be correlated. An example of such an array of evaluation windows is shown in Figure 2. One of the two sub-images to be correlated must always be larger than the other to allow the smaller image to be shifted within the larger image in both the EW and NS directions to determine the shift that produces the peak correlation. Since the correlation is in general performed after resampling one or both input images to a different resolution scale, the details of the location and sizes of the sub-images to be cropped from the input images are rather involved. More details are provided in the implementation description in Section 3.1 below. Image pair identification for GLM NAV is similar to that performed for the ABI non-NAV metrics in that a user supplied array of evaluation windows is used to select the sub-images of the GLM background image and the ABI “truth” image to be correlated.



Figure 2. Each diamond within the FD ABI image shown above represents the center location of a 64 x 64 sub-array of contiguous image pixels. These sub-arrays define the evaluation regions used for identification of sub-image pairs for all metrics except ABI NAV. In this example the evaluation region centers are equally spaced, and there are 25 such regions lying on the equator.

- **Sub-image resampling.** The shifts of one sub-image within another in the correlation processing step produce a two dimensional discrete array of correlation values. We refer to the two sub-images being correlated as lying in the “image plane” and the array of correlation values as lying in the “correlation plane”. The sub-image shifts

are quantized at the angular spacing between adjacent pixels. Since highly accurate estimates of registration error at the sub-pixel level are desired, it is necessary to reduce quantization impacts on sub-image registrations to a negligible level. There are two ways to do this: (a) Resample both sub-images to a common resolution scale finer than that either of the sub-images, and (b) Refine the estimated location of the peak correlation value in the correlation plane using the correlation values in the neighborhood of the peak value. The IPATS algorithmic design allows the user to select either or both of these methods to improve the accuracy of registration error estimation. For the special case of CCR processing, it should be noted, sub-image resampling to a common resolution scale is necessary for any two ABI images in bands having different resolution prior to correlation, irrespective of the need to choose the common resolution scale to be fine enough to provide the desired registration accuracy.

ABI NAV also represents a special case for sub-image resampling in that the Landsat chips are generated at 12x finer resolution than the native resolution of the corresponding ABI band. The possible resolution scales at which sub-images may be correlated for ABI NAV are 12x, 6x, 4x, 3x, 2x and 1x finer resolution than the native resolution of the ABI image under test. The Landsat chips are downsampled via uniform local averaging to the resolutions coarser than 12x finer than native ABI resolution.

IPATS provides three different interpolation algorithms for upsampling of images:

- Nearest neighbor interpolation
- Bilinear interpolation
- Bicubic interpolation (not yet implemented in code at the time of this writing)

IPATS provides two different algorithms for refinement of the correlation peak location:

- Parabolic refinement, in which the local maximum is calculated for the unique parabola passing through the unrefined peak value and its two nearest neighbors in each of the EW and NS directions.
- Centroiding, in which the centroid is calculated for the $N \times N$ sub-array of correlation coefficients centered on the unrefined peak, where N is an odd positive integer specified by the user.
- **Edge Enhancement Filtering.** Once the two sub-images to be correlated are resampled to the correlation resolution scale, IPATS provides the user with the option to perform edge enhancement filtering. Two edge enhancement filters are available, the Sobel filter⁶ and the Roberts filter⁶. Both filters are convolutional in nature, with the Sobel filtering performed by convolution of the image with a 3 x 3 kernel and the Roberts filtering performed by convolution with a 2 x 2 kernel.
- **Sub-Image Pair Screening.** IPATS incorporates a number of screening tests prior to the computationally intensive correlation step to terminate further processing of sub-image pairs that are unlikely to produce sufficiently strong correlations. An image pair must pass all of these tests in order to move on to the correlation module. Three of the most important screening tests are as follows:
 - Estimated peak correlation must exceed a user-specified lower bound. Since most registration errors are expected to be at the sub-pixel level when the ABI and GLM INR systems have been initiated and optimized, the correlation value of the two unshifted sub-images will be a good approximation of the peak value that will be found application of the correlation module.
 - Estimated measurement uncertainty must be less than a user-specified upper bound. The method for measurement uncertainty estimation that we have devised for application in IPATS is described in Section 2.5 below.
 - The minimum of the fractions of good pixels in the two sub-images must exceed a user-specified lower bound. In any ABI image it is possible that some or all of the pixels in a sub-image extracted by IPATS will be deemed of poor quality by ABI system processing. This test uses the pixel quality flags generated by ABI system processing to limit the fraction of poor quality pixels allowed in a sub-image that advances to the correlation module.

2.3. Correlation

The correlation module ingests two sub-images at the same resolution, in which one sub-image is larger than the other in both the EW and NS directions. The smaller sub-image is then shifted in both directions to all possible locations coincident with an isomorphic sub-array of the larger sub-image. For each such location a similarity metric is computed, either the Pearson correlation coefficient or the NMI, based on user selection. The results of these computations are captured into a two-dimensional array of similarity metrics, or “correlation array”, one for each shifted location. The size of the correlation array is always odd, with the center value corresponding to zero shift in both the EW and NS direction. The size of the correlation array is determined by the maximum anticipated registration error, a user provided input to IPATS, and by the resolution scale at which the correlation is performed, another user provided input. Some additional padding of the correlation array is required to insure that the raw correlation peak, prior to application of the peak location refinement algorithms, lies in the interior of the array, and furthermore that the sub-array centered on the peak required for both peak location refinement algorithms lies within the correlation array.

2.4. INR metric statistics

The INR metrics generated upon IPATS processing through the correlation module represent measurements of local absolute navigation error in the case of NAV and relative local navigation error in the case of all the other INR metrics except WIFR. These local navigation errors are aggregated into statistics at two levels, the image level and the multi-image level, where the latter is defined as the set of all the images of the same type falling within a prescribed time window. As mentioned in the introduction, since ABI INR requirements are specified with respect to a 24 hour period from local noon to local noon, the 24 hour time period is the time window of greatest interest.

The navigation error statistics that IPATS generates include the following: minimum, maximum, median, mean, median absolute deviation from the median (MAD), standard deviation, and the Pth percentile, where P is user specified. The 99.73rd percentile error is of special interest since the ABI requirements are expressed in terms of this statistic. IPATS provides the capability to generate versions of these statistics in which the individual errors are weighted by the estimated measurement uncertainty of each local navigation error. IPATS also calculates the combined measurement uncertainty of all the errors utilized in computing the standard deviation and provides the option to remove the measurement uncertainty from the standard deviation. This removal is performed by assuming that the standard deviation of the measured navigation errors is the root sum of squares (RSS) of the standard deviation of INR intrinsic errors and the standard deviation of IPATS measurement errors, and solving for the INR intrinsic component. The use of estimated measurement uncertainty to modify statistics will only be utilized if upon further study the measurement uncertainty is found to be sufficiently reliable as an absolute measure of uncertainty. IPATS provides the capability to use estimated measurement uncertainty in a more limited role to screen out sub-image pairs prior to correlation, as described above, and also post correlation to cull the measured errors deemed to be most reliable.

The WIFR error and its statistics are also calculated within IPATS in post-correlation processing. The angular separation of the nadir lines of sight (LOS) of the centers of two ABI sub-images that have undergone NAV error estimation is first computed based on the system provided navigation of these centers. The nadir LOS angular separation is then calculated for the case in which the angular locations of the sub-image centers are corrected using the IPATS estimated NAV errors. The nadir LOS angular separation calculated in this manner is considered to be the “truth”. The WIFR error is then calculated as the difference between these two nadir LOS angular separations. IPATS provides the capability to generate all of the statistics described above for both single image and multi-image aggregates of WIFR errors.

2.5. Measurement uncertainty estimation

Perfectly navigated and perfectly registered images of the same earth scene acquired under different conditions will differ from one another to some extent as arrays of measured radiances. As a result, when two such perfectly navigated and perfectly registered images are passed through an image correlator, a non-zero misregistration will typically be estimated. Under the circumstances specified for these image pairs, such misregistrations are “false” misregistrations. False misregistrations will vary in magnitude and direction, and constitute a “noise background” or “noise floor” that results in a minimum detectable *true* misregistration. Reliable detection of misregistrations due to true INR errors is therefore limited by classical signal-to-noise considerations. The “signal” due to a true misregistration must exceed the “noise floor” created by false misregistrations in order to be reliably detected.

We have developed an analytic method for estimating typical levels of false misregistrations. Such typical levels will be quantified by a “measurement uncertainty” (MU) metric. We take images to be rectangular $N \times M$ arrays of radiances.

We tag the radiances by their pixel locations within the image, i.e., $R(i, j)$ is the radiance at location (i, j) within the image, where $i = 1, 2, 3, \dots, N$ and $j = 1, 2, 3, \dots, M$. In our convention the image rows extend along the EW direction, so increments in the row index i represent NS displacements and increments in the column index j represent EW displacements. Then, consider a vector space structure for $R(i, j)$ which allows rescaling and perturbations using the rules of scalar vector multiplication and vector addition, respectively. The amount of spatial structure in an image is key to both the effectiveness of image registration techniques and to the method for determination of registration MU described herein. Therefore, we choose to describe images in terms of the deviation of the radiance in each pixel from the mean radiance over the image, normalized by the mean radiance, called $V(i, j)$.

Assuming that the image pixel coordinates (i, j) correspond to uniformly spaced navigation (or geolocation) parameters, such as the fixed grid angles used by ABI, then the translations of the image $V(i, j)$ span a two dimensional subspace of the $N \times M$ dimensional image space representing any possible incorrect navigation (or registration) of the image. The subspace can be represented mathematically as follows:

$$P(x, y) = V(i, j) + (x - j) \cdot T_x + (y - i) \cdot T_y \quad (1)$$

where $P(x, y)$ is the two dimensional plane of rigid translations of the single image, x and y are real numbers representing displacements in the EW and NS directions, respectively, and T_x and T_y are tangent vectors defining rigid translations in the x and y directions, respectively. Note that x and y have units of pixels, and are allowed to take on fractional values.

Given the granularity of the lattice over which an image is defined, the most natural way to define the tangent vectors is via one-pixel shifts in the image in the x and y directions:

$$T_x = V(i, j + 1) - V(i, j), \quad T_y = V(i + 1, j) - V(i, j) \quad (2)$$

In defining these tangent vectors we assume that the images are chipped out from larger images such that additional rows of pixels are available to define the tangent vector components on the image edges, where needed.

The essence of the MU estimation technique is to consider typical perturbations of the image that are *not* associated with translations and to estimate the misregistrations that the correlator would generate when ingesting the original and perturbed images. Consider a perturbed image $R''(i, j)$ and the associated image structure form $V''(i, j)$ as defined in equation (1). Then define the normalized difference vector $\delta V(i, j)$ as follows:

$$\delta V(i, j) = V''(i, j) - V(i, j) \quad (3)$$

The normalized image difference D is defined as the norm of this difference vector:

$$D = \|\delta V(i, j)\| \quad (4)$$

Now consider the set of perturbed images having a common magnitude of the image difference relative to an unperturbed image vector. The tips of the arrows of the perturbed image vectors all lie on a sphere S of radius D with the tip of the unperturbed vector at its center. Since the dimensionality of the image space is $N \times M$, the dimensionality of this sphere is $N \times M - 1$. Let δR_x denote the length of the projection of a vector on this sphere onto the tangent vector T_x . The associated value of the false misregistration δx in the x direction is obtained by dividing by the radiance change per pixel for a translation in the x direction. A suitable estimate of this object is the norm of the tangent vector T_x , since this object is the magnitude of the change in radiance for a one pixel shift in the x direction. Finally, we must consider the average magnitude of the false misregistration associated with the set of perturbed vectors lying on the sphere S . We assume that the perturbed vectors are uniformly distributed on the sphere S .

Putting all of the above considerations together, we can write the RMS expected false registration in the x direction as follows;

$$\langle \delta x_{false}^2 \rangle^{1/2} = \frac{1}{\|T_x\|} \left\{ \frac{\int_S x^2 d\sigma}{\int_S d\sigma} \right\}^{1/2} \quad (5)$$

where the integrals are surface integrals over the sphere S and $d\sigma$ is the surface element on the sphere. Note that the argument x can be along any coordinate direction in the $N \times M$ dimensional space, since the distribution of perturbed image vectors is assumed to be isotropic. In other words, the orientation of the plane of rigid rotations of the unperturbed image is immaterial under our assumptions.

Before proceeding with the calculation of the surface integrals, note that the expression for the RMS x misregistration given by equation (5) has the correct dependencies on contrast C , the normalized image difference D , and the number of pixels in the image $N \times M$. As the contrast C increases, so does the norm of the tangent vector, since higher contrast entails greater pixel-to-pixel image differences. Since the norm of the tangent vector is in the denominator, the magnitude of the false misregistration decreases with image contrast. Since D is the radius of the sphere, the projection of a vector on the sphere in the x direction clearly increases with D , as one would expect. Finally, as the number of pixels in the image and hence the dimensionality of the sphere increases, the magnitude of the projection of a perturbation vector on a single axis will decrease. This is due to the greater number of directions in which a vector on the sphere can point, resulting in less likelihood of significant projection in any one direction that is singled out. This behavior may be less obvious than those involving C and D , but will be borne out by the calculation of the integrals.

$$\langle \delta x_{false}^2 \rangle^{1/2} = \frac{1}{\|T\gamma\|} \frac{1}{\sqrt{NM}} D \quad (6)$$

$$\langle \delta y_{false}^2 \rangle^{1/2} = \frac{1}{\|T_y\|} \frac{1}{\sqrt{NM}} D \quad (7)$$

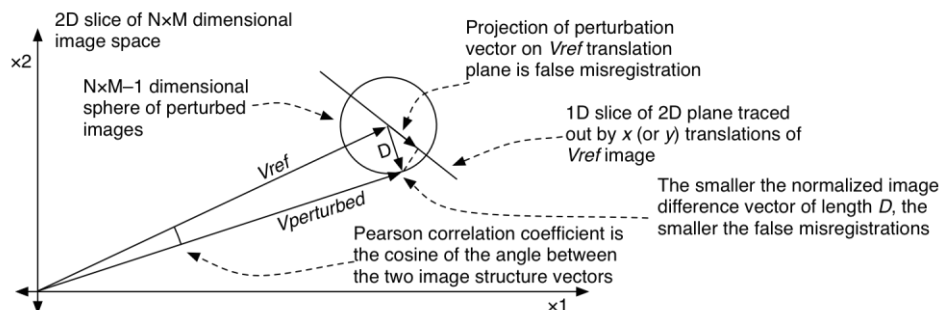


Figure 2. 2D representation of geometrical Measurement Uncertainty estimation concept.

A refined version of MU that takes into account the spatially correlated nature of image perturbations is under study but has not yet been incorporated into IPATS. Depending on the relative magnitude of MU compared to INR metric standard deviations, MU may be handled in different ways. If MU is negligible compared to INR metric standard deviations, then the 3-sigma metrics may be treated as entirely associated with system INR performance. However, if MU is comparable to INR metric standard deviations, then our algorithmic approach allows the removal of MU from the measured standard deviations of misregistrations, leaving a metric more representative of system INR performance. This removal is accomplished by considering the total measured standard deviation of misregistrations to be the root-sum-of-squares (RSS) of the INR intrinsic standard deviation and the MU standard deviation, and solving for the INR intrinsic standard deviation.

3. IMPLEMENTATION

3.1. IPATS Image Pair Selector and Evaluator (IPSE)

At the highest level, the IPATS Image Pair Selector and Evaluator (IPSE) is a C++ based tool that evaluates a set of input images or directories of images. It first catalogs all of the image files within the directories specified, loading

metadata from each image. Through a set of configurable parameters, the user can specify what types of evaluations should be performed and the parameters for each evaluation type. The IPSE then performs each of the desired evaluations at one or more locations within each input image or image pair, and then outputs the evaluation results to a SQLite database file, where the results can be queried and compared by an analyst.

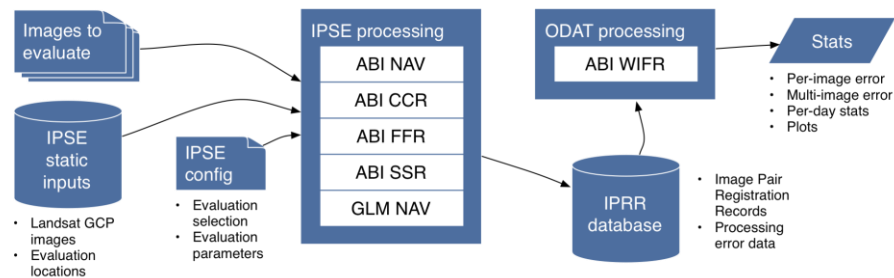


Figure 3. High-level architecture and data flow for IPATS

For each evaluation type, the IPSE must determine what images, from the arbitrary set of files provided by the user, should be compared against each other or against truth data:

- NAV: All input images are compared against one or more images from a library of image chips described in Section 3.3, so each input file is put on a list of images to be evaluated.
- FFR: The IPSE sorts the input images by nominal collection time (the start time extracted from the filename). Starting with the earliest image, the IPSE looks through the list of images until it finds the very next image (by nominal start time) for the same satellite position, band and image type (Full Disk, CONUS or Mesoscale) as the initial image. For Mesoscale images, the IPSE also checks to make sure there is a minimum amount of geographic overlap between the two images (a user configurable parameter). Once the pair of images is identified, the pair is placed on a list of image pairs to be evaluated.
- CCR: From the sorted list of input images, the IPSE looks at each group of files that have the same nominal start of collection time, which is guaranteed to be identical for all bands for a single collection, regardless of the precise start time for that specific band. The user specifies band pairs to be compared as well as any band specific edge enhancement processing. For each user specified pair of bands, the input images for that band pair for that collection time are placed on a list of image pairs to be evaluated.
- SSR: Details to still be worked out
- GLM NAV: Details to still be worked out

At the core of the IPSE, common to all evaluation types, is a single image evaluation module that can compare any two input files at a given location, over a given pixel span centered on that location. For FFR, CCR and SSR, this allows the actual image processing and evaluation to be performed by the exact same code, minimizing inadvertent differences in evaluation. This module functions as follows:

The IPSE compares the pixel sizes of the two images in the pair (e.g., 28 μ rad to 14 μ rad), nominating one image to be treated as the lower resolution image and the other to be the higher resolution image. If the pixel sizes are different, this selection is obvious. If they are the same, the center wavelengths of the bands of the two images are compared, and the image with the larger center wavelength is nominated as the lower resolution image. If the center wavelengths are also the same, the image collected later in time is considered the lower resolution of the two images. Once the lower and higher resolution images are determined, the IPSE determines the ratios between the two image resolutions and between each image resolution and the target resolution. From these ratios, IPSE calculates the number of additional pixels outside the evaluation region that are necessary to calculate the resampled chips with inducing edge effects. This evaluation requires that the ratio of the resolutions of the two images is an integer, that the pixel alignment between the two images is co-sited for odd resolution ratio and centered for even resolution ratio, and that the ratio between each of the image resolutions and the target resolution is also an integer or 1/integer.

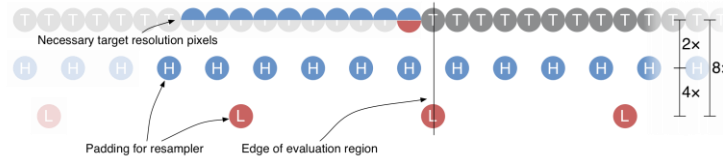


Figure 4. The IPSE calculates the amount of padding necessary beyond the evaluation region necessary to generate the desired target resolution image for both the higher and lower resolution images. The darker red “L” circles show original lower resolution pixels, and the lighter blue “H” circles show original higher resolution pixels, in this case where the two images have a resolution ratio of four. In this illustration, the sub-pixel factor is set to eight. The red and blue half circles in the top row show the necessary pixels at the target resolution for the lower and higher resolution sources, in order to process the image with maximum anticipated registration error set to one lower resolution pixel, with at most a three pixel wide peak refinement kernel, and with an edge enhancement filter with a width of three pixels. This results in one pixel of padding being necessary from the lower resolution image and six pixels of padding being necessary from the higher resolution image. Ghosted out pixels on the left show pixels that are either not necessary (from the source images) or not calculated (at the target resolution).

The IPSE then considers the specified evaluation center location and evaluation region size, specified in pixels. First, the IPSE adjusts the specified center location so the edges of the desired region perfectly align with the centers of pixels in the lower resolution image. For example, if the evaluation pixel width is even, the fixed-grid X coordinate of the evaluation center must fall exactly between two lower resolution pixel centers. The IPSE then expands that region, in both the lower resolution and higher resolution by the amount of padding necessary to ensure that the desired image region necessary for cross-correlation can be provided without inducing edge effects. That expanded region is extracted from each input image file. The IPSE assumes that whatever expanded pixel region is necessary for each matching location is available from both input sources.

The IPSE resamples up or downsamples by local averaging each of the two images to the target resolution, as appropriate. If edge enhancement has been specified for the pair, either the Sobel or Roberts filter is applied to both resampled images.

The two processed images are then pre-screened (as described in Section 2.2) using the Image Pair Characterization (IPC) module. If IPC results do not satisfy user-defined thresholds, evaluation of the image pair at that location is aborted, preventing execution of the cross-correlation procedure (a computationally expensive step in the process). But if IPC returns a good or fair status based on pre-screening tests, cross-correlation of the two images is calculated. The location of the raw peak correlation value is identified, and then this location is refined using both parabolic and centroiding methods. The raw and refined misregistration values are included in an Image Pair Registration Record (IPRR), which includes a complete record of all the parameters required to reproduce the image pair registration, including user supplied parameter values, identification of image data sources, and all values calculated by the IPSE. All IPRRs are captured in the IPSE output database.

If evaluation is aborted for any reason, whether due to pre-screening status or detected error, an error report is added to the IPRR output database for that evaluation.

For FFR, CCR and SSR, the application of this common core evaluator module is straightforward. The IPSE configuration includes a pointer to a database of evaluation locations, with flags indicating which locations should be used for which image types and for which band resolutions. For each image pair, the IPSE searches the evaluation location database for locations contained within the geographic intersection of the two images, and then evaluates those two images at each of those locations in turn.

NAV, however, does not represent a comparison between two input images at one or more arbitrary locations within the images, but is the comparison of one input image to one or more chips of limited geographic extent from a library of Landsat chips, where the input image takes the role of the lower resolution image and the Landsat chip takes the role of the higher resolution image. However, unlike for the other cases, each Landsat chip is of limited small size, and the desired evaluation region represents the entire geographic region of the chip. In this case, there is no additional image data to provide padding for the cross-correlation region, and thus before the IPSE can execute the common evaluator module on the ABI/Landsat image pair, the IPSE must work inwards from the full extent of the Landsat chip to determine what portion of that chip will be consumed as cross-correlation and edge-effect elimination padding, leaving what is left, rounded down to the fixed-angle grid of the ABI image, as the evaluation region.

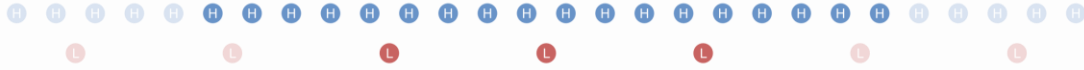


Figure 5. For an example case, the IPSE calculates the padding necessary for a Landsat chip at a 4x higher resolution than the ABI image. In this example with a 28 pixel wide Landsat chip, the lower resolution evaluation region can have a width of at most 3 pixels

Once this calculation is made, a software wrapper is placed around the Landsat chip such that it functions as if it were an ABI input image, and then the ABI image under test and the wrapped Landsat chip are handed off to the common evaluator.

The IPSE code uses storage classes and a few methods from the Open Source Computer Vision (OpenCV) library⁷ to leverage platform-specific speed enhancements.

3.2. IPATS Output Database Analysis Tool (ODAT)

Generically, IPATS measures registration error between two image pairs and stores the individual registration error and associated metadata in a SQL (Structured Query Language) database where each registration error record is denoted as an Image Pair Registration Record (IPRR). “Registration error” in this section refers to registration error in the EW (X) direction, registration error in the NS (Y) direction, or both, depending upon the context. A separate tool called the IPATS Output Data Analysis Tool (ODAT) was developed to perform data analysis on groups of IPRRs, with the goal of computing several metrics to measure operational ABI and GLM system-level INR performance over a large set of individual registration error measurements. ODAT was also developed to analyze results of IPATS during debugging, testing, and optimization.

A single Image Pair Registration Record (IPRR) contains all of the required information to describe a single registration error between two arbitrary image sub-images processed within IPATS. Registration error can be exactly reproduced using only the information contained within an IPRR and re-running IPATS. For example, an IPRR contains the registration error, the mother image(s) information (such as file name), the chipped locations of the sub-images with respect to their mother image, the input parameters to IPATS, various debugging parameters, measurement uncertainty, and other miscellaneous information. The vast majority of the ancillary information is tabular and redundant across IPRRs; therefore, a standard SQL (tabular) database is sufficient to store the data. A SQLite3 database was chosen for the initial IPATS testing because it is simple to set up and is file-based.¹⁷ A minimalist C++ SQL Object Relational Mapper (ORM) was developed to abstract the SQL-specific architecture away from the core IPATS code. The SQL ORM provides an easy-to-use interface for C++ developers on the team to interact with the database without requiring SQL-specific knowledge. The ORM automatically generates and executes the required SQL statements based on the end-user (programmer) defined C++ objects.

ODAT was developed with two goals in mind: to provide a standard interface to the IPATS IPRRs for advanced users, and to provide an easy-to-use graphical interface for users who simply want to compute a set of stock statistics and plots (see Section 2.4). ODAT has been developed in Python (version 2.7 or higher). Python is becoming increasingly popular for data analysis due to its powerful language constructs (it is a fully object-oriented language), its increasingly refined data analysis packages, and the fact that it is open source and free.^{9,10} In addition, Python is multi-platform, enabling easy deployment to Windows, Linux, and Mac environments.⁹ In order to keep the number of external Python packages to a minimum, ODAT uses stock Python 2.7 modules such as Tkinter for the graphical user interface (GUI) and the SQLite3 connector for the database.⁸ The only required external package is SciPy, and ODAT makes heavy use of the SciPy sub-modules NumPy, Matplotlib, and Pandas for the data manipulation and analysis portions of the code due to the flexibility and performance these packages offer.^{10,11,12,13,14}

Launching ODAT will result in a simple window that allows the end-user to browse to the database containing the IPRRs (the “IPRR database”) and a variety of options for the calculations/plotting capabilities within the tool. From the main window (Figure 6a), one can launch a window that allows the end-user to directly see every table and value within the database. The majority of users will view the images in the database (using the “View Images” window) or view the IPRRs in the database (using the “View IPRRs” window - Figure 6b). Setting the options and parameters within ODAT is accomplished from the main window (Figure 6a). The first option is simply selecting the IPRR database to be analyzed. Several calculation/plot options are provided including: desired units for output results (choice of pixels with respect to the lower resolution image or micro-radians in the global angular coordinate system), choice of mean/median

for plotting groups of IPRRs, choice of outlier rejection methods, and definition of temporal bounds for time-based statistics.

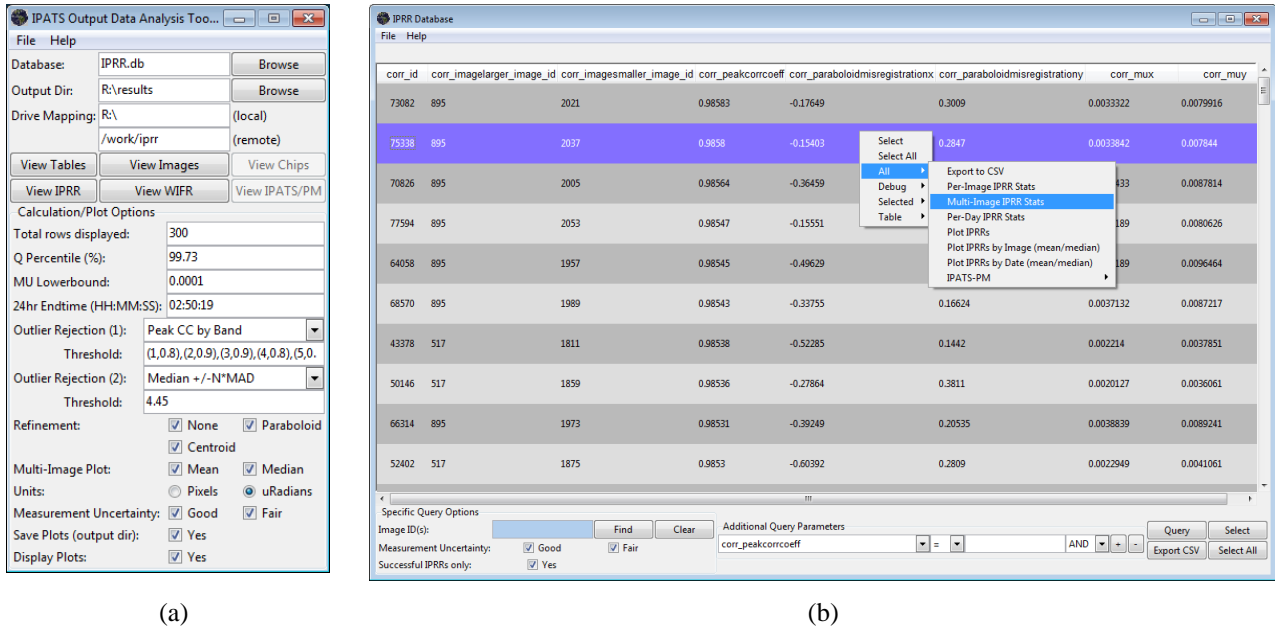


Figure 6 (a) ODAT primary window (b) ODAT IPRR Table

End-users can view individual IPRRs by clicking “View IPRR” from the primary ODAT window. Clicking “Query” will result in a table filled with rows where each row represents an IPRR in the database (Figure 6b). From here, users can elect to filter the data, export the data, manipulate the data, or query IPRRs associated with a set of input images. This window allows the end-user to set an arbitrary restriction on the IPRRs returned via query. For example, the end-user may only want IPRRs that are with respect to a particular image band and have a registration error less than a specified threshold. Users are able to query on any parameter (or column) within the database by graphically stringing together a query with comparison operators ($=$, $<$, $<=$, $>$, $>=$, etc.) and logical operators (and/or). The graphical flexibility allows the end-user to query and cull the data to exactly what is needed without knowledge of SQL. ODAT abstracts complex queries from the end-user by internally generating the required SQL syntax the user has graphically constructed. After querying the data, a visual table is displayed that allows the user to analyze the data returned.

Once a user has queried a particular subset of IPRRs within the database, analysis can be performed. At the most basic level, ODAT can export the resulting IPRR table to a comma-separated value (CSV) text file, which can then be imported into another program for analysis such as a spreadsheet program. For the majority of analysis purposes, the end-user will not have to export the data. ODAT stores the SQL returned tabular data in memory using versatile and efficient SciPy Pandas DataFrame objects.¹⁰ Having the data in the Pandas DataFrame allows ODAT to group the data in complex ways such as over 24-hour periods or by image band using the Pandas groupby functionality.¹³ When the data is grouped accordingly, the DataFrame objects are used to generate several stock statistics such as the mean, median, minimum, maximum, and standard deviation of the registration error. The analysis of the IPATS registration error requires several custom statistical computations, which are performed by binding custom functions to the DataFrame objects (similar to subclassing; however, Python allows one to essentially bind a new method to an existing object).

Plotting the data in IPRRs can aid in analysis, particularly in determining if there are gross outliers that may affect the resulting statistics. Stock ODAT plots include a scatter plot of the registration error in the X and Y directions, registration error over time (computed as a mean/median), or as the mean/median of registration errors for a set of images. Most plots include error bars where the error is the computed measurement uncertainty. The plots are generated using the SciPy Matplotlib package.¹²

After analyzing the statistics and plots, the end-user may decide that outliers are skewing the results and choose to apply one of the custom outlier-rejection methods. ODAT provides two levels of outlier-rejection, global and local. Global outlier-rejection is defined as a single or set of user-defined thresholds applied to all IPRRs, without knowledge of variable grouping. Local outlier-rejection is defined as a variable list of thresholds, where the variable list is generated using local groupings of the data. For example, the end-user may want to reject all IPRRs that have a registration error outside the median $\pm N \times MAD$, where N is a user specified positive integer, but generate variable thresholds over a 24-hour period and per band. Once again, the Pandas DataFrame grouping capability is used with custom functions to perform the outlier rejection. ODAT allows the end-user to “string” the outlier rejection methods together such that they are performed in sequence.

3.3. Multispectral Landsat Chip Library

For ABI NAV correlation tests, a Multispectral Landsat Chip Library (MLCL) was created to allow for comparison of spectrally similar bands between ABI and Landsat instead of correlating all ABI bands to a single Landsat spectral band or a simple Boolean Land/Water Mask. The Landsat 8 Sensor System was utilized to make “truth” chips to evaluate the ABI NAV performance because of the high fidelity of the Landsat 8 navigation performance and significantly higher spatial resolution¹⁵. Table 1 summarizes the GOES-R ABI and corresponding Landsat 8 spectral and spatial characteristics.

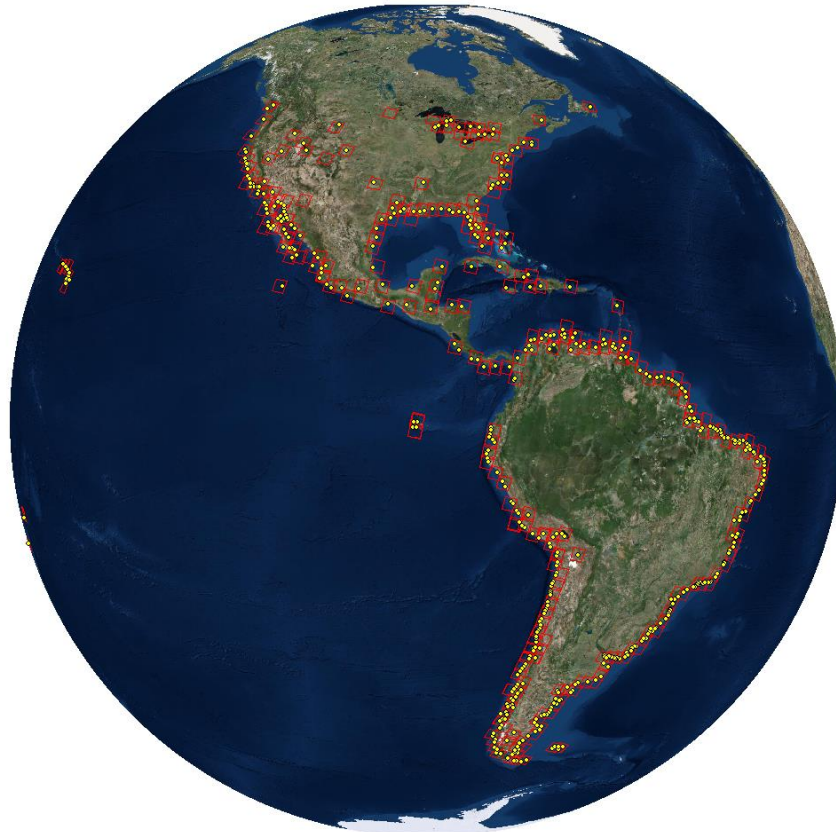
Table 1. Summary of ABI and Landsat 8 bands utilized for NAV correlation. Red filled boxes indicate ABI absorption bands not utilized in NAV correlations. Landsat 8 TIR bands are acquired at 100 m GSD but resampled to 30 m for delivery. The ABI Instantaneous Ground Field of View (IGFOV) is specified at nadir.

GOES-R ABI Band	GOES-R ABI Wavelength Range (um)	GOES-R ABI Nominal IGFOV (km)	Landsat 8 Band	Landsat 8 Wavelength Range (um)	Landsat 8 Nominal GSD (m)
1	0.45-0.49	1	2 - Blue	0.45-0.51	30
2	0.59-0.69	0.5	4 - Red	0.64-0.67	30
3	0.846-0.885	1	5 - NIR	0.85-0.88	30
4***	1.371-1.386	2			
5	1.58-1.64	1	6 - SWIR	1.57-1.65	30
6	2.225-2.275	2	7 - SWIR	2.11-2.29	30
7	3.80-4.00	2	7 - SWIR	2.11-2.29	30
8***	5.77-6.6	2			
9***	6.75-7.15	2			
10***	7.24-7.44	2			
11	8.3-8.7	2	10 - TIR	10.60-11.19	100 (30)
12***	9.42-9.8	2			
13	10.1-10.6	2	10 - TIR	10.60-11.19	100 (30)
14	10.8-11.6	2	10 - TIR	10.60-11.19	100 (30)
15	11.8-12.8	2	11 - TIR	11.50-12.51	100 (30)
16	13.0-13.6	2	11 - TIR	11.50-12.51	100 (30)

***Atmospheric band that does not see to the ground. Landsat chips are not generated for these bands, and they are excluded from NAV evaluation.

The MLCL inputs are comprised of a single CSV text file that contains GCP records of longitude, latitude, Landsat Path/Row, a unique integer ID and identifying landmark name and the full Landsat 8 scenes downloaded from USGS’s EarthExplorer Site (<http://earthexplorer.usgs.gov/>). The CSV file was manually populated with predefined GOES-R product monitor (PM) landmarks, as well as locations of spectrally significant points along coastlines and inland features. The most cloud-free Landsat 8 scenes available that underlie the GCP locations were downloaded, as well as scenes from multiple seasons for areas with diverse seasonal variations, where possible. Code was written to automatically extract and unzip the Full Landsat Scenes (delivered as gzipped tar files), organize the scenes into a proper directory structure, construct a reduced resolution quick-look geotiff image and create a master polygon shapefile of each scene’s footprint and metadata, including date, filename, path/row, and cloud and snow cover. This allowed for a

geographic information system (GIS) of the full scenes that can be easily queried and visualized with the GCP point locations, as well as discover areas that require multi-seasonal or less cloudy scenes to be acquired from future Landsat collection orbits. Currently, the MLCL consists of 530 GCP locations and 456 Landsat Scenes, as shown in Figure 7, and is still being updated as more Landsat scenes become available.



Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community

Figure 7. MLCL Landsat 8 scene footprints (Red) and GCP locations (Yellow).

Standalone code was written in IDL to generate the individual MLCL chips for use in within IPATS and a corresponding output CSV file that contains metadata for each of the output image chips. Each of the input GCP's is cycled over and the following steps are performed (Figure 8):

- Subset the corresponding full Landsat scene into a 4200x4200 image centered on the GCP
- Compute each image's longitude, latitude and height and convert to ortho-rectified ABI fixed grid (FG) coordinates (details provided below) for the three GOES-R satellite positions (137°W, 89.5°W and 75°W)
- Subset the result into a rectangle with no "background" pixel values
- Convert digital numbers (DN) into at-satellite radiances
- Average into a pixel ground sample distance (GSD) 12 times finer than the corresponding ABI spectral band
- Subset the output further to make certain that the chip is perfectly nested within the native ABI pixel grid

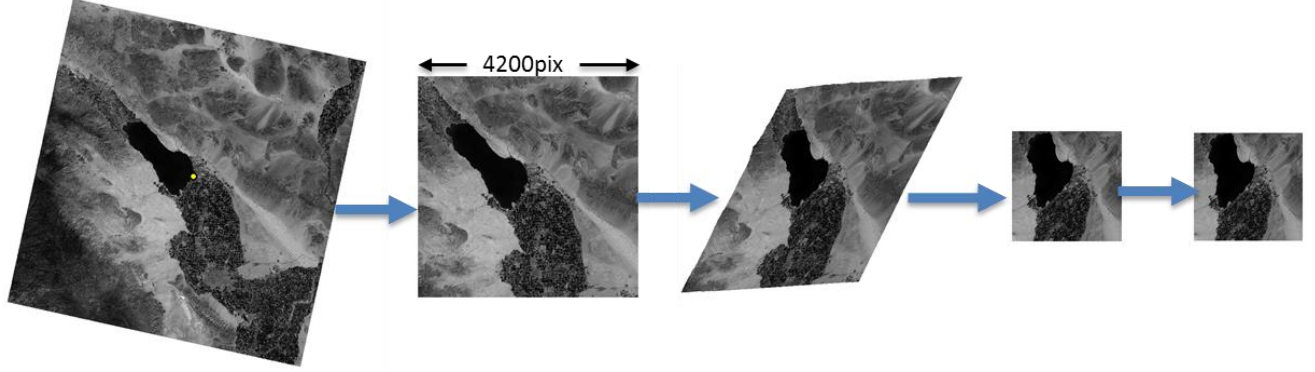


Figure 8. From left, full Landsat scene with GCP location (Yellow), 4200x4200 pixel subset, FG remapping to 1/12 the ABI GSD, subset to non-background pixels, final subset to native FG pixel grid

For image registration, the chips must be generated in the fixed grid coordinate system used by GOES-R. When acquired from the USGS EarthExplorer system as described above, the Landsat image files are map projected to a universal transverse Mercator (UTM) projection. Thus, the image files must be re-projected to the GOES-R fixed grid. The per-pixel re-projection process can be summarized as follows: (a) Convert UTM coordinates to geographic coordinates (latitude/longitude). (b) Extract height-above-ellipsoid (HAE; WGS84/EGM96) topography value from the SRTM 90 m dataset.¹⁶ (c) Convert geographic coordinates to Earth-Centered, Earth-Fixed (ECEF) (also referred to as “earth-centered rotational”/ECR) coordinates. (d) Convert ECEF coordinates to fixed grid coordinates (angles). Details on steps (c) and (d) are provided below.

Conversion of geographic coordinates to ECEF (X, Y, Z):

$$X = (N(\phi) + H) * \cos(\phi) * \cos(\lambda) \quad (8)$$

$$Y = (N(\phi) + H) * \cos(\phi) * \sin(\lambda) \quad (9)$$

$$Z = (N(\phi) * (1 - e^2) + H) * \sin(\phi) \quad (10)$$

where ϕ is the latitude in degrees, λ is the longitude in degrees, H is the HAE SRTM topography elevation, and $N(\phi) = \frac{a}{\sqrt{1-e^2\sin^2(\phi)}}$ where a is the semi-major axis and e is the first eccentricity of the WGS84 ellipsoid.

Conversion of ECEF coordinates to fixed grid angles/coordinates, assuming a spacecraft position \mathbf{p} (p_x, p_y, p_z) in ECEF coordinates:

$$\mathbf{w} = (\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z) / |\mathbf{p}| \quad (11)$$

$$\mathbf{v} = (0, 0, 1) \quad (12)$$

$$\mathbf{u} = \mathbf{v} \times \mathbf{w} = (-p_y, p_x, 0) / |\mathbf{p}| \quad (13)$$

Define: $\mathbf{A} = \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \end{bmatrix}$, $\mathbf{B} = \mathbf{A}^T$

For each Landsat pixel:

$$\mathbf{u}_1 = \begin{bmatrix} X - p_1 \\ Y - p_2 \\ Z - p_3 \end{bmatrix} \quad (14)$$

$$\mathbf{v}_1 = \mathbf{B} \mathbf{u}_1 \quad (15)$$

Extract fixed grid coordinates/angles using the following relationship, which defines the line-of-sight vector from the spacecraft to the point on the Earth’s surface in terms of the fixed grid angles:

$$\mathbf{v}_1 = \begin{bmatrix} \sin \theta \\ \sin \varphi \cos \theta \\ -\cos \varphi \cos \theta \end{bmatrix} \quad (16)$$

where θ and φ are the X and Y fixed grid angles, respectively.

The result of this process is a set of fixed grid angles/coordinates (θ, φ) for each pixel in the input Landsat image. From this intermediate product, a set of resampled and trimmed chips for IPATS processing is generated, as described above. Note that this conversion process will fail for any location not visible from the satellite's position, so the following validity check is included, where the point is NOT visible if the inequality evaluates as TRUE:

$$H(H - s_x) < s_y^2 + \frac{r_{eq}^2}{r_{pol}^2} s_z^2 \quad (17)$$

where H is the satellite's height above the center of the earth, r_{eq} and r_{pol} are the equatorial and polar radius of the ellipsoid, and the s_x and s_z are defined as follows:

$$s_x = H - r_c \cos(\phi_c) \cos(\lambda - \lambda_0) \quad (18)$$

$$s_z = r_c \sin(\phi_c) \quad (17)$$

where ϕ_c is the geocentric latitude: $\phi_c = \tan^{-1}\left(\frac{r_{pol}^2}{r_{eq}^2} \tan(\phi)\right)$ and r_c is the geocentric distance to the point on the ellipsoid: $r_c = \frac{r_{pol}}{\sqrt{1 - e^2 \cos^2(\phi_c)}}$.

This process is repeated for each Landsat spectral band, the Landsat QA/QC image (to retrieve cloud and snow cover) and the height array (used to retrieve min, max and mean elevation). For each spectral chip, metadata is inserted as a new record in the output CSV, and a floating precision flat binary file is output with an associated ENVI ascii header file into an output directory for use within IPATS. Four example chips are illustrated in Figure 9 below, and an example output CSV record is demonstrated in Table 2.

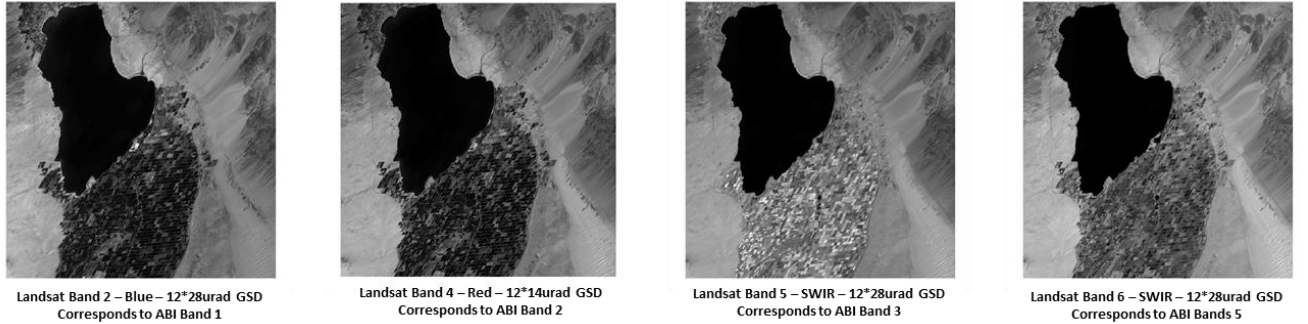


Figure 9. Four spectral examples for a GCP chip at the 89.5°W GOES-R satellite longitude

Table 2. Example output CSV record specifying a single chip's metadata

NAME_S16	ORIGLAT_R	ORIGLON_R	GCP_ID_U	LANDMARK_S24	NEWLAT_R	NEWLON_R	FILENAME_S128	ROWS_U	COLS_U	BANDS_U	PROJLON_R	PROJLAT_R	BANDNUM_U	ANGGSD_R	RSMULT_U
P45R33	38.939918	-123.665208	1000	P45R33-1000	38.93982959	-123.6655477	LC80450332013289LGN00_B4_75_1000.dat	696	696	1	-75	0	4	1.17E-06	12
TARGETABIGSD_R	MIN_X_R	MAX_X_R	MIN_Y_R	MAX_Y_R	WAVELENGTH_R	WAVEUNITS_S48	CIRRUS_R	CLOUD_R	SNOW_R	DATE_D	MINALT_R	MAXALT_R	MEANALT_R	COMMENTS_S1024	
0.000014	-0.095549417	-0.094738583	0.101780583	0.102591417	0.655	Micrometers	0.0291	0.0679	0.0328	10/16/2013	-36.17	789.939	138.482		

4. PRELIMINARY TEST RESULTS

Preliminary IPATS science testing has been focused on demonstrating expected behavior for a variety of NAV tests. For NAV, a given ABI chip is correlated to a Landsat Chip as described earlier. A free parameter in this correlation is the factor by which the ABI chip is upsampled, also called the Sub-Pixel Factor (SPF). The raw misregistration is expected to change as SPF is increased because of the resolution change. With SPF=1, the raw misregistration is resolved to the nearest 1 pixel, while with SPF=12, the misregistration can be resolved to the nearest 1/12 pixel. In contrast, when the parabolic method is used to refine the misregistration, the expectation is that misregistration will vary less as SPF is increased. Figure 9 shows the X direction misregistration medians versus SPF for both raw and parabolic refined

misregistrations. Here, a large number of proxy ABI images in band 3 were compared to the Landsat chip library to produce the calculated misregistrations. The median X direction misregistrations are calculated for each SPF to show group behavior. We see the biggest difference between parabolic refined and raw misregistrations at SPF=1. The parabolic refinement is able to resolve the misregistration quite accurately ($\sim 1/3$ pixel), despite the low resolution of ABI images. As these refinements are relatively constant as a function of SPF, consistent behavior (in the parabolic refined plot) is verified. These results were first filtered so that the peak correlation coefficient (peak CC) values were greater than 0.8 for each attempted correlation. This filtering ensures the validity of the misregistrations and removes bad (cloudy) ABI sub-images.

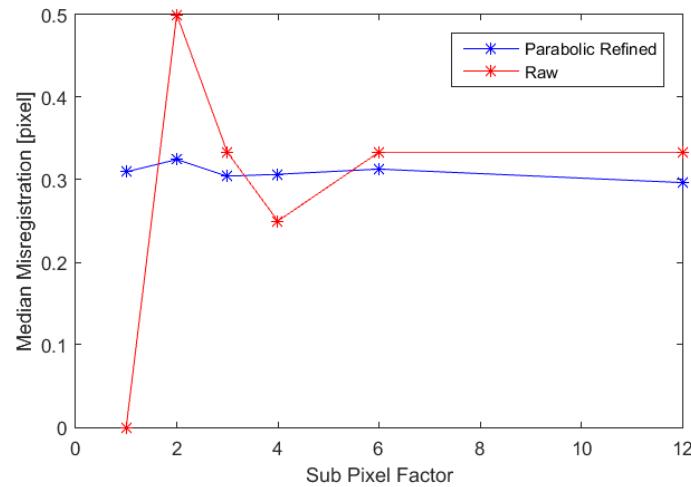


Figure 9. Median misregistration versus Sub-Pixel Factor.

After initial tests on misregistration behavior, misregistration dispersions were analyzed for consistency. MU is a measure of dispersion and was compared to misregistration Median Absolute Deviation (MAD). Figure 10 shows dispersion behavior as a function of SPF for the same dataset used to generate Figure 9. Again, we observe stable behavior for the MAD of the parabolic refined misregistrations, while MAD of the raw misregistrations exhibits greater variation. We also note that MU results are far lower than MAD misregistrations. As stated earlier, this suggests that the MAD is likely measuring true INR dispersion. The large size of the Landsat chips and image evaluations regions contributes to the MU being so small.

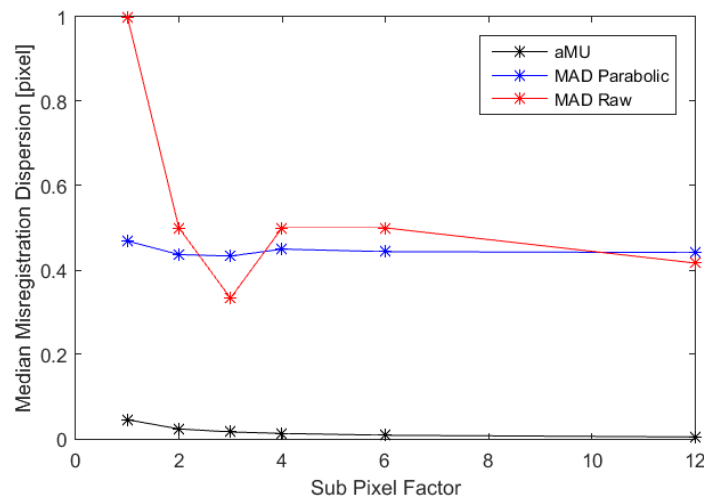


Figure 10. Misregistration dispersion versus Sub-Pixel Factor.

Initial IPATS NAV test result match intuition and exhibit consistent behavior. This will allow for studies to optimize parameters such as SPF, peak CC filtering, and other types of filtering. As a large part of the IPATS code is shared

between the various evaluation types (NAV, FFR, CCR, etc.), these results also contribute to the validation of those modes.

5. SUMMARY

IPATS development and test is nearing completion at the time of this writing. Preliminary performance results indicate that the tool set is functioning properly and produces results as expected for tests performed to date. IPATS has been designed and implemented to meet all key objectives: (a) the capability to generate all specified INR metrics, including individual measurements and statistics, (b) user selectability of alternative algorithmic processing modules for generation of these metrics, allowing optimization of processing on a per-metric basis, (c) identification and optimization of algorithmic processing modules common across all metrics, (d) formulation of a method for measurement uncertainty estimation, allowing the measurements most contaminated by measurement error to be filtered out prior to INR metric generation, and potentially allowing compensation of measurement uncertainty impacts on INR metrics that remain after filtering, (e) output data manipulation, filtering and visualization capabilities, and (f) output database capture of analysis results and all the IPATS configuration information needed to reproduce these results.

REFERENCES

- [1] Product Definition and Users' Guide (PUG) Volume 3: Level 1B Products for Geostationary Operational Environmental Satellite R Series (GOES-R) Core Ground Segment, Revision D, 13 May 2015, <http://www.goes-r.gov/users/docs/PUG-L1b-vol3.pdf>
- [2] GOES-R Series Mission Requirements Document (MRD), GOES-R-Program/Code 410, 410-R-MRD-0070, Version 3.20, January 2016, <http://www.goes-r.gov/syseng/docs/MRD.pdf>
- [3] Goodman, S., Blakeslee, R., Koshak, W., Mach, D., Bailey, J., Buechler, D., Carey, L., Schultz, C., Bateman, M., McCaul, E., Stano, G., "The GOES-R Geostationary Lightning Mapper (GLM)", Atmospheric Research, Volumes 125-126, 1-84 (May 2013)
- [4] Carr, J. L., J. Fox-Rabinovitz, D. Herndon, S. Reehl, "Verifying the Accuracy of Geostationary Weather Satellite Image Navigation and Registration", Ninth Annual Conference on Future Operational Environmental Satellite Systems, 93rd American Meteorological Society Meeting (2013).
- [5] Studholme, C., D. L. G. Hill and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment", Pattern Recognition, 32(1), 71-86 (1999).
- [6] N. Senthilkumaran, R. Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches", Int. J. of Recent Trends in Engineering and Technology, Vol. 1, No. 2 (November 2009).
- [7] Open Source Computer Vision (OpenCV), 02 October 2014, <http://opencv.org/> (04 February 2016)
- [8] SQLite Consortium, "About SQLite," SQLite, 2016, <https://www.sqlite.org/about.html> (02 February 2016).
- [9] Python Software Foundation, "Python Language Reference, version 2.7," 2016, <http://www.python.org> (02 February 2016).
- [10] McKinney, W., [Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython], O'Reilly Media, (2012).
- [11] Van der Walt, S., Colbert, C., and Varoquaux, G., "The NumPy Array: A Structure for Efficient Numerical Computation," Computing in Science & Engineering, 13, 22-30 (2011).
- [12] Hunter, D. J., "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, 9, 90-95 (2007).
- [13] McKinney, W., "Data Structures for Statistical Computing in Python", Proceedings of the 9th Python in Science Conference, 51-56 (2010).
- [14] Jones E., Oliphant E., Peterson P., et al., "SciPy: Open Source Scientific Tools for Python," 2001-, <http://www.scipy.org/> (02 February 2016).
- [15] Landsat 8 Data Products, December 2015, <http://landsat.usgs.gov/landsat8.php>
- [16] Shuttle Radar Topography Mission (SRTM), January 2014, <http://glcf.umd.edu/data/srtm/description.shtml>