Multi-resolution Image Registration Algorithm (MIRA): Robust Automated Image Registration using Python

Warren J. Hack (Space Telescope Science Institute), Nadezhda Dencheva (Space Telescope Science Institute)

A problem with combining images taken at different epochs remains the difficulty in determining the offset between the images in an automated manner. Automated multi-scale techniques have been adopted for use in aligning images taken with ground-based telescopes with great success. However, several characteristics of Hubble Space Telescope (HST) images made direct application of the ground-based multi-scale techniques problematic. This paper describes how an algorithm developed for automated image alignment of Earth observing satellite imagery was merged with a multi-scale analysis of the images to produce a new image registration task. This task was developed using Python, with C extensions for efficiency, and has undergone initial testing to successfully align HST images of the core of 47 Tuc and, separately, mosaics of HST images of the Orion Nebula without any differences in the parameter settings. More work needs to be done to improve the run time and to generalize it to work on non-HST images before making it publicly available on all platforms which run Python.

Application to actual HST observations

Initial verification of MIRA included running it on pairs of images containing vastly different types of sources: specifically, an extended source which filled the field of view and a crowded field of point sources. Images from the HST Advanced Camera for Surveys (ACS) taken of the Orion Nebula and 47 Tucanae (globular cluster) were identified as initial test cases. The original WCS information in the headers of the images was used by PyDrizzle to combine each pair into a combined mosaic and an aligned pair of exposures on the same pixel grid as the combined mosaic to determine whether the images align properly. The Orion images used for this test can be seen in Figure 2[a] and [b] while the 47Tuc images can be seen in Figure 3[a] and [b]. Errors in the WCS information in the image headers result in poor alignment in these pairs of images. An RGB image was generated for each pair using one aligned output image as the Red channel, and the other as the Green and Blue channels (creating cyan). The original alignment with the errors can be seen in Figure 2[d] for a expanded view of a portion of the Orion mosaic and Figure 3[c] for the 47Tuc mosaic. It can be seen from these images that a significant offset exists between the WCS information of these input exposures.

MIRA was then used to compute the offsets between the images in each pair with no user provided parameters except the input filenames. The computed offsets were then used to combine the images again using PyDrizzle. A new set of RGB images was generated for each pair in exactly the same way as before. The same expanded view of the aligned Orion images can be seen in Figure 2[e], while Figure 3[d] shows the alignment in the 47Tuc mosaic.

Future Development

This algorithm still requires a considerable amount of work to make it a fully general routine for image registration. That work would include:

- **Testing** on a larger set of image types to verify the robust nature of the algorithm.
- **Verify** that it can work on large mosaics of data, instead of just pairs of images, and characterize the amount of image overlap which is required for a good fit.
- **Iterate** on the initial solution to identify a larger set of sources for generating a more accurate final fit.
- **Improve** the operational speed of the algorithm. Current testing indicates it takes about 2 minutes to process each ACS/WFC with an undistorted size of about 4200×4200 pixels.
- **New** medium algorithm, one that can be up to 9 times faster than traditional median filters, has been developed. This new median filter will be implemented within MIRA soon to provide initial significant improvements.
- **Support** non-HST images. This will require making PyDrizzle support non-HST images while requiring some adherence to FITS WCS keyword standards.

References


MIRA relies on a combination of techniques derived both from astronomical image analysis and from earth observation satellite image analysis as performed in the geosciences. The algorithm relies on multi-resolution analysis for preparing the images for source identification, and on the feature-based registration algorithm described by Dai and Khormaz (1999) for identifying the sources. Computation of the shifts for a set of images undergoes the following steps:

1. **Build initial mosaic relationship for exposures using WCS header information and PyDrizzle (Hack, 2002).**
2. For each detector chip, create an instance of a Chip class. This class will take the chip’s image and:
   - Correct for distortion and generate a distortion-free image for this chip only.
   - Generate multi-resolution views using median filtering, doubling the filter size for each successively lower resolution.
   - Starting with the lowest resolution view, use Laplacian-of-Gaussian filter to generate contours of all sources in image. Close contours with the strongest edges are identified using the Thin and Robust Zero-Crossing technique described by Dai and Khormaz (1999).
   - Generate the modified Freeman chain-code for each source’s contour (Li et al., 1995). These chain codes are invariant to scale, rotation, translations as well as being robust against noise in the contours.
   - Compute the invariant moments and center-of-gravity of the image within each source’s contour.
   - For each successively higher resolution, extract sources, chain-codes and moments in the same way as described in steps 3 and 4 from sources identified in the regions extracted from the lower resolution views of the image.
3. Combine source lists from chips associated with the same exposure into a single ‘Observation’ class.
4. Identify an exposure as the reference to use as the starting point for computing the offsets. This will usually just be the first image given by the user as input.
5. Perform image matching between the identified reference image and the next image in the list:
   - Compute 3 matrices based on the moments, center-of-gravity, and chain-codes for all the sources from the reference image and the comparison image.
   - Search for potential matches with the matrices. For any pair of regions, there will be a threshold which defines the limit for differences between the sources in terms of the moments and contour coding.
   - Generate a reduced-potential matrix set based on the resulting set of potential matches which represents the ‘distance’ between the features in each pair. Find the ‘mode’ of these distances and identify all the pairs which fall within that mode.
   - Use the 3 best matched points to perform an initial linear fit between the images and report the results.
6. The offsets computed by this algorithm are then fed to PyDrizzle to generate the final aligned images.

Current Status

This algorithm has been tested only with a few pairs of images representing widely discrepant types of observations. It has successfully computed the offsets for these pairs with no inputs specified by the user demonstrating its potential for automatically and robustly computing offsets for astronomical images without regard for cosmic-ray contamination. This program has been developed using Python, with C extensions for some pixel-based operations, and can be supported on any platform which the STScI_Python package can be ported: including Linux, Solaris, Mac, Mac/Intel, and Windows.