**A NEW APPROACH TO CREATE IMAGE CONTROL NETWORKS IN ISIS.** K. J. Becker, K. L. Berry, J. A. Maple, J. C. Walldren , U. S. Geological Survey, 2255 N. Gemini Dr., Flagstaff, AZ (kbecker@usgs.gov)

**Introduction:** A global digital elevation model (DEM) [1] and monochrome basemap [2] of Mercury were created using a least-squared bundle adjustment [3] technique that corrects errors in image pointing attitude and solves for local radius at each control point. New methods were used to create a feature-based control point network that contains 94.75 million measures from 100,432 overlapping image observations covering the entire surface of Mercury. This effort required the development of new tools in the Integrated Software for Imagers and Spectrometers (ISIS3) [4] system to address the complexities encountered in processing very large image datasets with varying illumination and geometric conditions.

These tools utilize specialized image and data processing techniques, some available in open source libraries, and compute cluster environments to increase the quality and efficiency in the generation of cartographic and topographic products. The new applications were made available in the ISIS 3.4.13 public release.

**Objectives:** The goals of this new approach are to: 1) significantly improve the image tie point measurement accuracy, 2) increase control point density, 3) reduce both human and computer resources required for producing and processing large control point networks, 4) efficiently identify and add new images to existing networks, and 5) create interpolated DEMs directly from bundle-adjusted control networks. Creating a DEM and a geometrically controlled basemap share common objectives, but creating a DEM from interpolation has one additional unique requirement: very high control point density.

Creation of the image control point network requires the most effort in the process of producing high quality controlled map products. For large datasets, the challenges are significant. Established techniques in ISIS3 [5] are adequate when the dataset is small to modest, however as the dataset increases in size, productivity and quality are *quickly* diminished. Therefore, a reliable and efficient process must be applied to ensure a complete and high quality network.

**Image Matching:** We developed a feature-based matching (FBM) application called *findfeatures* using the OpenCV 3.1 [6] open source library. The advantages of *findfeatures* are: 1) the matching is feature-based (FBM) rather than area-based (ABM), 2) users can select and interchange all OpenCV algorithms to detect, extract and match features, 3) all OpenCV algorithms can be customized though user-specified parameterization, 4) accurate a priori pointing information is not required, 5) OpenCV algorithms are threaded in the library and therefore available at the application level, and 6) a set of robust outlier detection algorithms were developed to identify and remove false positives, which is new to image matching in ISIS3.

*Feature Detection:* Matching features in common regions of overlapping images using FBM algorithms increase the accuracy and quality of the network. However, one must consider the use of scale and rotation invariant detectors, such as SIFT [7], if the a priori pointing data is unavailable or inaccurate. Otherwise, algorithms that assume spatial consistency, such as FAST [8], can be used with excellent results. *findfeatures* provides the ability to use the SPICE information in ISIS3 to quickly *project* images with perspective transforms to support the use of both basic types of OpenCV detector algorithms. We found the FAST detector algorithm is effective in producing a high density of features in many remotely sensed images.

*Feature extraction:* Once features are detected, a description of the feature, called a *keypoint descriptor*, is extracted using algorithms that characterizes the feature in various forms. For example, the SIFT algorithm essentially computes the distribution of gradient magnitudes and orientation from pixels around the feature.

*Matching Features:* These descriptors are used in final matching steps to identify *potential* common features. OpenCV provides several matching algorithms that are intended for use with specific types of descriptors. *findfeatures* will select an appropriate matcher algorithm, such as BruteForce or FlannBased, contingent upon the type of descriptor if one is not explicitly specified. Note that this does not always guarantee a valid match, resulting in false positives, or *outliers*. These false matches must be removed prior to bundle adjustment through specialized techniques.

*Robust Outlier Detection:* Outlier detection is the final step that ultimately improves the quality and integrity of the control point network. *findfeatures* implements a five-step outlier detection process for each set of matching image pairs. At each step, matches are rejected allowing only those surviving to be passed on for bundle adjustment.

First, a ratio test comparing the best and second best matched feature for each point in an image match pair removes points that are not distinct. Second, matched feature points are checked for symmetry. If a matched point in one image does not correspond to the same point in the other, the point is removed. Third, the homography matrix (a spatial projection from one image's perspective into the other) of the first image to the other is computed using the RANSAC algorithm removing high residual matches. Fourth, the fundamental matrix (which represents the stereo relationship) between the first and second image is computed using the RANSAC algorithm. Feature points that exceed the specified tolerance using the RANSAC algorithm are removed. A final homography matrix is recomputed from the results of the previous steps and matches with a high residual are removed.

This robust outlier detection process is an important new technique that provides a higher quality network prior to the bundle adjustment.

**Resource and Data Management:** Fundamental image preparation for control work in ISIS3 is common processing that includes ingestion, optional radiometric calibration, and application of NAIF SPICE [9] ephemeris kernels (*spiceinit*) to prepare them for geometric and observation classification.

*Data Mining:* A new general-purpose application called *isisminer* was developed providing many of the capabilities of a GIS-enabled database as well as extended capabilities. This application reads from many data sources such as databases, comma-separated values (CSV), parameter-valued languages (PVL) and ISIS cube (labels and GIS footprints) files.

GIS-enabled SQLite databases can be created using *isisminer* processing features that stores image statistics (*camstats*) and GIS surface footprints (*footprintinit*). SQL queries can be applied in the database to identify images in certain geographic areas and constrain observation properties.

*Scripting: isisminer* has the ability to run UNIX shell commands and applications, including other ISIS3 programs. This feature combines mining with immediate processing of the query results in a single run of *isisminer*. Data is read from the GIS overlap CSV files and geometric and observational constraints are applied in *isisminer* prior to running *findfeatures*, creating image-based control networks. This approach allows *isisminer* scripts to be designed that focuses on custom image-based processing techniques that are well suited for dispatch to cluster environments.

*Combining networks:* An application called *cnetcombinept* was developed to combine these small image networks into a large single global network. This application uses kd-trees [10] to efficiently organize and search for common image measurements (coordinates) in close proximity control points and combine them based upon difference statistics. This approach minimizes the size of the final network, while retaining density, by transferring unique measures from multiple common feature points into a single found in all overlapping image-based networks. This adds depth of all images containing the same feature (i.e., image measures to points) thus increasing the integrity of the network.

*DEM Generation:* To produce a DEM from the bundle adjusted images, *jigswaw* must be set to solve for radius. This computes a local radius value at each control point in the output network which can then be used to interpolate a DEM. The density of the control points determines the scale of the resulting DEM. The new application, *cnet2dem*, was developed to construct a kd-tree of the adjusted latitude and longitude in body-fixed X,Y,Z form and create an ISIS DEM cube in a user specified cartographic map projection. Several algorithms are provided to choose from as well as statistics at each output map point to evaluate the resulting topographic map values and integrity.

*Threading:* Most all of the OpenCV algorithms are threaded for efficient use of compute resources. *findfeatures* provides user controls over the number of CPUs that can be utilized to execute the feature matching algorithms to better utilize processing environments.

*Managing Point Density:* The *findfeatures* application is designed to create dense control networks. Control network density can be minimized while maintaining adequate coverage of the control point distribution between image overlaps for purposes of controlling images to create a mosaic. For these cases, *cnetthinner* was developed to greatly reduce the density while preserving integrity (image point depth) and point distribution.

**Future Work:** Improvement to *isisminer* and *findfeatures* are planned. *findfeatures* needs enhancements to better process large image sizes. It is currently well suited for small to modest size framing instruments, but many of the new instruments produce much larger images in pixel dimensions. Testing of small body datasets as well as data from other instrument types will also be investigated. We will investigate adding threaded processing to *isisminer* to increase efficiency and reduce processing time.

**References:** [1] https://pds.jpl.nasa.gov/ds-view/pds/viewDataset.jsp?dsid=MESS-H-MDIS-5-DEM-ELEVATION-V1.0. [2] https://pds.nasa.gov/ds-view/pds/viewProfile.jsp?dsid=MESS-H-MDIS-5-RDR-BDR-V1.0. [3] Edmundson, et al. (2011) *GSA Abst. with Program, 43*, paper 100-6, p. 267. [4] Anderson, et al., (2004) *LPS*, 35, abstract 2039. [5] Garcia, et al. (2015) LPS, 46, abstract #2782. [6] Bradski (2000) *Dr. Dobb's Journal of Software Tools*, drdobbs.com.[7] Lowe, (2004) *Inter. Journal of Comp. Vision*, 60, 2, 91-110. [8] Rosten, et al., (2006) *Comp. Vision–ECCV*, 430-443. [9] Acton, (1966) *Planet. Space Sci. 44*, 65-70. [10] https://github.com/jlblancoc/nanoflann