SPARSE MULTI-IMAGE CONTROL: THE AUTOCNET LIBRARY J. R. Laura, K. Rodriguez, A. C. Paquette, U.S. Geological Survey, Flagstaff Arizona, jlaura@usgs.gov

Introduction Accurate control and registration of remotely sensed, non-Earth targeting, planetary imagery is both critical for science applications and challenging due to the lack of sufficiently accurate sensor position and/or pointing information. Current techniques are pseudo-automated with significant human interaction required to seed correspondences between images, reject outliers, apply block bundle adjustment, and iteratively repeat until an acceptable solution can be identified [4]. Rapidly increasing data volumes and the need for globally registered foundational data products [7, 2] require a number of improved automation techniques, including sparse *n*-image correspondence identification. That is the identification of tie points between some number, *n* of images. To address this need, we have developed a Python library to support the sparse identification of image correspondences, *n*-image matching, outlier rejection, and block bundle adjustment called AutoCNET.

The AutoCNET library is developed as a hybrid photogrammetric-computer vision (CV) approach to solve the *n*-image correspondence problem for three reasons. First, we initially leveraged the freely available OpenCV computer vision library. This library provided a large quantity of existing functionality that supported rapid prototyping and allowed for more constrained identification of functionality that would be required. We no longer depend upon this library due to installation complexity and lack of performance working with large images. Second, our hybrid CV based approach allows us to operate with or without an initial, rigorous camera model. While this removes a potentially valuable piece of information, it significantly simplifies processing. We note that back estimation of a rigorous camera model using CV techniques is a proven and widely accepted technique. Third, by using a CV approach, we are able to leverage the vast quantity of cutting edge terrestrial work; this work is largely in the CV and not photogrammetry domain.

Architecture AutoCNET is developed inline with standard open source software development practices. We host the source code on the USGS Astrogeology GitHub organizational page (https://github.com/USGS-Astrogeology/autocnet), utilize the TravisCI continuous integration environment for unit and functional testing, provide a number of tutorial notebooks using the Jupyter notebook technology, provide documentation via ReadTheDocs (http://autocnet.readthedocs.io/), and continuous delivery using the Anaconda Python build environment. AutoCNET is a modular library of functionality and not a delivered end-to-end processing pipeline. This is an intentional design decision due to the complexity and heterogeneity of sparse control problems. For example, different outlier detection methods are more suitable for different data characteristics. Therefore, we supply a suite of methods that a user can test, select, and bulk apply.

Data Representation: As indicated above, a graph-based storage representation, with nodes representing input images and edges representing the adjacency or overlap between images, was selected. A graph-based representation affords a number of low overhead benefits. First, the graph theory field provides a number of existing graph manipulation libraries (NetworkX) with a full suite of standard graph manipulation routines (e.g., sub-graph identification, clustering, graph-based metrics, etc.). Second, graphs are commonly used to represent large data sets due to ease of partioning using a variety of techniques.

Correspondence Identification: The first step in matching *n*-images is the identification of features known as interest or keypoints. Keypoints are Rotation, Scale, and Transformation (RST) invariant features (groups of pixels) within an image and are identified as the minima and maxima of the image after iteratively applying a Difference of Gaussians function [8]. Keypoints are matched using either an approximate K-nearest neighbors algorithm or a more performant brute force Graphics Processing Unit (GPU) based approach. The latter requires a CUDA enabled NVIDIA GPU. Once extracted, we store all keypoints for the lifetime of the project; we have developed an iterative processing library that can reuse previously discarded keypoints.

Outlier Detection: The primary difficulty in planetary *n*-image matching is not the process of identifying image correspondences, but rather the balancing of type I and type II errors (errors of omission and errors of commission, respectively). AutoCNET makes available standard symmetry based (e.g., AB = BA) correspondence checking and Lowe's ratio test that measures the ambiguity between the top two candidate matches for each correspondence [8]. Estimation of the geometric relationship between two or more images is also a common outlier detection mechanism and AutoCNET supports the computation of pairwise homographies (though we note that these are planar and topography induced error will be a concern) as well as the Fundamental Matrix (F). AutoCNET also supports Maximum Likelihood Estimation of a refined F matrix that minimizes triangulation error [6]. Sidiropoulos [10] show that these outlier detection techniques are inadequate as the total pixel count increases. Therefore, AutoCNET also provides CPU- and GPU-based coupled decomposition techniques that iteratively

match and decompose large images into corresponding sub-images.

Refinement: We note that a purely feature-based CV approach does not provide adequate positional accuracy as feature-based correspondence detection techniques (as well as some template based approach such as Normalized Cross Correlation) offer, at best, pixel level accuracy [3]. Therefore, we have implemented both a naive template based matcher and the Ciratefi algorithm. Ciratefi is an RST invariant template based matcher [1] that allows us to avoid geometric reprojection.

Sparsity: The cost of bundle adjustment increases exponentially with the number of identified image correspondences. Therefore, sparsity with good geometric coverage is essential. AutoCNET offers two methods to improve coverage and sparsity. First, radial suppression [5] is a heuristic solution technique that ranks pairwise correspondences by some cost function and attempts to select the 'strongest' correspondences with good spatial distribution using a user define maximum correspondence count. AutoCNET also supports n-image Voronoi diagram computation using the reprojective Homography (that is accurate enough for a bounding box projection) to assess the total area each correspondence is attempting to cover. This metric is critical in identifying regions with too few correspondences.

Interoperability: As part of the AutoCNET project, we seek to iteratively apply matching, outlier detection, and bundle adjustment techniques; bundle adjustment is just another outlier detection method in this context. Therefore, we have wrapped the ISIS3 bundle adjustment functionality (Jigsaw) using the SIP library for access via Python. This allows for native Python calls to execute a component of the ISIS API and view the resultant information as a Pandas DataFrame. Additionally, we have wrapped (or developed and wrapped) a number of GPU based libraries to support more performant matching, (e.g. CudaSift and CudaCoupledDecomposition).

CTX Current work focuses on n-image correspondence identification and control of data collected by the Mars Reconnaissance Orbiter Context Camera (CTX) [9]. The CTX camera is a line scan sensor with a 5000 pixel linear array at approximately six meters per pixel. The CTX sensor has been selected for three reasons. First, CTX data exhibits wide variability in total spatial extent per image (see Figure 1). The disparity in extent and homogeneity of the surface require additional a priori matching constraints using either standard techniques or coupled decomposition techniques. Second, the CTX data set has over 99% global coverage and greater than 60% repeat coverage making this a high priority candidate data set for both another controlled base and future medium resolution change detection studies. Finally, in seeking to utilize a hybrid photogrammetric-CV approach, we have explicitly assumed that a rigorous camera model is not available for correspondence identification. This approach has held for a framing camera and CTX allows us to determine if this assumption will hold for a more complex line scan sensor.

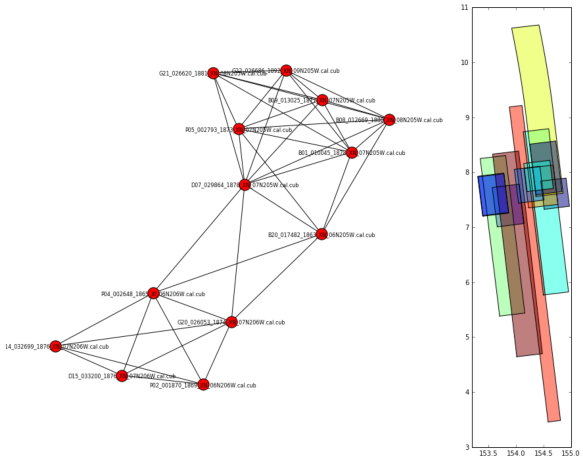

Figure 1: Thirteen test images in the Athabasca Valles region as a graph object (left) and visualized footprints (generated using ISIS3 and the ISIS3 camera model).

Conclusion We have developed an open source library that supports iterative matching of images. This work leverages proven terrestrial matching techniques to provide scalability and automate the matching process. We have identified two primary areas of future work. First, our ISIS3 interoperability efforts write an intermediary control network to disk and we seek to work entirely in memory. Second, we are testing our library and expanding necessary functionality by creating a small relatively controlled CTX mosaic. We seek to expand this work and integrate additional users during FY18.

References

[1] Sidnei Alves de Araùjo and Hae Yong Kim. In: *Integr. Comput.-Aided Eng.* 18.1 (Jan. 2011), pp. 75–90. [2] B. A. Archinal et al. "Foundational Data Products". In: *This Issue*. 2017. [3] Misganu Debella-Gilo and Andreas Kb. In: *Rem. Sens. Env.* 115.1 (2011), pp. 130 – 142. [4] K. L. Edmundson et al. "47th LPSC 1376". In: [5] S. Gauglitz et al. In: *18th IEEE Intr. Conf. on Image Proc.* 2011. [6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*. 2nd ed. 2003. [7] J.R. Laura et al. "Planetary Sci. Vision 2050 Wshp, Abst. 8110". In: 2017. [8] David G. Lowe. In: *Int. J. Comput. Vsn.* 60.2 (Nov. 2004), pp. 91–110. [9] Michael C. Malin et al. In: *Journal of Geophysical Research: Planets* 112.E5 (2007). E05S04, n/a–n/a. [10] P. Sidiropoulos and J. P. Muller. In: *IEEE Trans. on Image Proc.* 24.7 (2015), pp. 2124–2139.