ON-ORBIT DETECTION, IDENTIFICATION, AND TRACKING OF GEOGRAPHIC TARGETS WITH A LONG WAVE INFRARED CAMERA

Joseph Carragher[1], Alexander Tyler[1], Brian Gunter[1*], Darren Rowen[2], Phaedrus Leeds[2], and Shon Cortes[2];
[1]Space Systems Design Laboratory, Georgia Institute of Technology, [2]xLab, The Aerospace Corporation,
[*][brian.gunter@aerospace.gatech.edu]

Abstract. *High resolution imagers and multi-satellite systems can create large and unsustainable downlink requirements for space missions. On-board processing can improve the efficiency of this data collection through the autonomous identification of targets of interest. This study demonstrates a methodology that leverages open-source software and commercial-off-the-shelf (COTS) hardware to autonomously track and identify known surface targets in real-time from imagery gathered from a Long Wave Infrared (LWIR) camera. The targets for this study were restricted to lakes and islands, but the approach can accept and search for a wide range of alternate geometries.*

Introduction. With an increase in imager resolution and the deployment of complex multi-satellite systems, downlink capabilities have become a limiting factor in many systems. The utilization of on-board, autonomous processing provides an alleviation for this downlinking problem as well as a method of lower latency processing than what can be done via a ground station. Through the use of publicly available computer vision tools and COTS hardware, the image processing loop can be effectively automated to allow for high quality identification and tracking of objects on the ground from orbit.

The utilization of a LWIR camera provides a unique capability as opposed to standard visual spectrum imagers–the direct view of thermal signatures.[1] When used to view thermally distinct ground objects (e.g. the view of a lake against a landscape or an island against an ocean), the raw image is capturing shapes representing distinct IR signatures in greyscale, as shown in Fig. 1. This convenient signature can be exploited using common computer vision tools, in particular OpenCV[i], to capture these objects in a fine manner agnostic of the visual brightness or clutter of the scene.

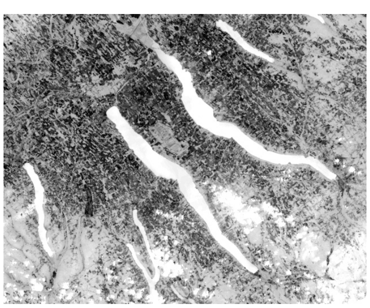

Figure 1. Landsat-8 TIRS Image of Seneca Lake.[2]

[i]https://opencv.org

Computer Vision Logic. The logic of this image processing can be split into three main modes: (a) Searching, (b) Identification, and (c) Tracking.

Searching Mode. The searching loop starts with the reception of an undistorted image–on which edge detection is performed through the usage of Otsu masking. This technique slices the image into a user defined number of masks describing different regions of intensity on the histogram. The threshold chosen between these masks is done in a way to minimize the intra-class variance of each. For the previous mentioned thermally distinct images due to the LWIR imager, the distinct models in the histograms generated are ideal for the application of Otsu's method.[3] From these image masks, canny edge detection is used via 'cv2.Canny' to gather a list of edges in each scene.

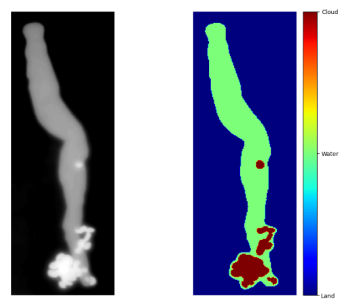

Figure 2. 3-Layer Otsu masks (right) of Seneca Lake IR target (left).

From these edges, for each Otsu mask, contours are found using 'cv2.findContour' and the contour with the largest area is selected–leaving a single contour for each layer in the image. Each contour is then vectorized and converted into a Shapely polygon. The orientation of this polygon is then characterized using its longest and shortest axes. This is done by finding the two farthest points on the body and assigning the longest axis to the line connecting them. The shortest axis is then defined as the perpendicular bisector to that longest axis. Notably, ambiguity with respect to the orientation and the 180 degree offset of that orientation is alleviated by defining the shortest axis as pointing from the shorter side of said axis to the longer side of it.

Thus, the software has now detected and characterized a set of objects in the scene.

Identification Mode. Two optimizations can be made here that are enabled by the catalog stored onboard.[4] The first is predicated on the catalog storing the area of each object. Since the area of the objects detected can be estimated using the altitude of the satellite and the size of the object in the view of the camera, objects with areas outside a user-defined confidence interval can be filtered out of the comparison set. The second optimization can be made by utilizing a rough knowledge of the satellite's latitude and longitude. If the catalog also contains the latitude and longitude of each entry, the entries can be cut into tiles. Then, only tiles near the satellite will be selected to compare against. By doing both of these, we significantly cut down the number of comparisons to only entries roughly near the satellite and roughly the size of the objects detected.

With this reduced comparison set, two methods are employed to attempt to identify the objects detected in frame: (a) comparing moment invariants between objects and (b) evaluating the Jaccard index of objects. To start, Shapely polygons are created for each of the catalog entries in the reduced comparison set. For every catalog polygon and detected object, the four third order affine moment invariants[5] are computed and the Euclidean distance between them is calculated. In testing, this method is not reliable enough to determine matches, but can be used to reliably detect most non matches. Thus, it is employed as a rapid method of evaulating if the Jaccard index of the objects–the ratio of the intersection of the two sets over their union[6]–needs to be evaluated. In the case that it does, since the Jaccard index is not invariant to rotation, a Golden Section optimization method is utilized to maximize the Jaccard index as a function of the rotation of the catalog polygon using the previously estimated orientation as a starting point.

After going through every combination of catalog polygon and detected polygon, the entry with the highest Jaccard index, if above a certain threshold, is taken to be the correct match to the object in the scene. Now that the object in the scene has been identified, the software stack can move to tracking the object in the frame.

Tracking mode. Independent methods are used for tracking linear movement and rotational movement.

For each frame, a bounding box is drawn to encompass the edges detected for the object previous identified. This is used to perform median optical flow tracking enabled by 'cv.TrackerMedianFlow' to track linear motion of the object in the frame. The center of this bounding box is recorded each frame and compared to the previous frame to quantify the movement of the object between frames.

For rotational tracking, OpenCV's ORB (Oriented Fast and Rotated BRIEF) is utilized to quantify the orientation change between frames. By matching the descriptors between two frames, the corresponding points can be found to build a Homography matrix. The transformation described by this matrix can be used to calculate the rotation angles for each axis and finally, the total rotation angle of the object between frames can be computed.

The tracking loop repeats so long as the object remains in the frame. If it leaves, the software reverts to the searching mode to find a new object to identify and track.

Validation. To validate the software loop described above, test targets were made that replicate a sharp IR gradient that will be experienced when viewing features like lakes on-orbit. The targets are straightforward in design–a painted metal plate is laser cut with the desired feature (e.g. a lake). This results in a metal feature surrounded by paint. While this type of target can only show two signatures, it is possible to create a three-layer target. By lowering the power level and increasing the speed of the laser cutter, a third layer is created by burning the paint instead of etching it away completely. This results in an intermediate temperature layer between the two regions in the prior target design.

These targets allow for full hardware-in-the-loop (HWIL) testing of the algorithms and COTS hardware in the system.

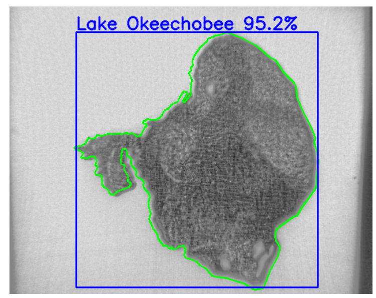

Figure 3. Lake Okeechobee identified and tracked.

Over the course of 10 runs, the computer vision software matched the target in Fig. 3 with a mean Jaccard index of 0.95426 with a standard deviation of 0.0097026.

Discussion. The capabilities of COTS hardware and publicly available computer vision tools have a great potential to alleviate the issue of throttled downlink requirements for multi-satellite systems operating with high data volumes. With the addition of easy to manufacture HWIL testing suites, the barrier to development of these systems for IR optical systems is low. While these examples and some of the optimizations made are specific to geographical bodies, the basic premise of matching polygons can be applied to many cases beyond that of identifying lakes.

References.

[1] Teledyne FLIR, *Uncooled LWIR OEM Thermal Camera Module*.

[2] U.S. Geological Survey, "LC08 L1TP 016030 20200618 20200824 02 T1 TIR," 2020.

[3] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan. 1979.

[4] World Wildlife Foundation, "Global lakes and wetlands database." https://www.worldwildlife.org/pages/global-lakes-and-wetlands-database.

[5] J. Flusser and T. Suk, "A moment-based approach to registration of images with affine geometric distortion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, pp. 382–387, Mar. 1994.

[6] P. Jaccard, "Etude de la distribution florale dans une portion des alpes et du jura," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, Jan. 1901.