Ortho-Project Notes

Marvin Smith

October 16, 2012

Current Progress

- Revamped and restructured the entire ortho module. Reviewed the math and structured the code in an effort to improve readability.
  - Note: DEM Correction is still being worked on.
- Built baseline structure for OrthoProject unit tests. Need to start validating that the large number of geometric utilities are accurate to a desired level of precision. I have tested them during development, however this is something that needs to be implemented long-term.
- Beginning to implement coordinate conversion utilities, testing DEM Module, and working on utilities for parsing GS1 NITF Headers. This is required to integrate DTED and geographic imagery as we are dealing with complex coordinate systems and different units of measure.

Metrics

GeoImage

- Lines of Code: 10223
- Libraries: OpenCV, Boost Filesystem, GDAL
- New Features since 9/14
  - `GEO::GeoImage::get_image( int CV_TYPE )`
    - You can now retrieve the image from a GeoImage in any format you wish.
  - `GEO::GS2::GS2_Header`
    - Get and set functions for various GS2 NITF header items.
    - Currently aircraft tail number and focal length. More planned.
  - `GEO::CoordinateBase`
    - Framework in place to implement Coordinate Conversion using GDAL.
    - Currently containers in place for UTM, Geodetic, and LambertConic
    - Still developing overall architecture for API

OrthoProject

- Lines of Code: 5333
- Libraries: OpenCV, Boost Filesystem, GeoImage

Issues

Pixel-by-pixel into Homorgraphy

I don't see a means of converting the pixel-by-pixel search into a homography. This is because once we have the ground point, we must search for occlusions. A homography has no method of testing for occlusions as it cannot search. I think the best method is to create a highly optimized search function, then apply each pixel independently as a thread in a mass parallel pipeline.

Overall Scope

Integration of DTED is going to require a lot more cycles than I anticipated. This project requires well-developed algorithms and libraries to manage the large number of coordinate systems, file types, header requirements, and processor/memory demands. DTED in particular is dangerous as loading large tiles can kill system memory. Once I get a baseline working for small samples, I need to investigate a more robust and scaleable method for handling very large datasets. Also I have not begun to consider what happens if holes are not filled, as the value is $-2^{16} + 1$. This along with the fact that most geographic coordinates do not fall directly along a dted post, interpolation between coordinate posts will further tax our algorithm.

Future Plans

- Finish OrthoProject code, specifically...
  - Make DEM Correction work for test imagery
  - Make DEM Correction work for Geographic imagery
  - Read camera intrinsics for Geographic imagery from nitf headers.
  - Build test bench for batch testing.
  - Create static library build to run on better machines (GS2 Servers have 64 cores and are x64 RHEL).
  - Begin laying down framework for parallelization.
- Create tools to determine camera intrinsics from geographic imagery.
  - Have user select N+ ground points
  - Create system of equations required to solve
  - Use least squares if solving homography
  - If not linear, possibly Levenburg Marquardt

  - Finally, I have very well tested Genetic Algorithm code which I am confident will do the trick.
  - Create a single file format for saving camera models.

- Once all of this is solved, begin stitching/bundle adjustment process