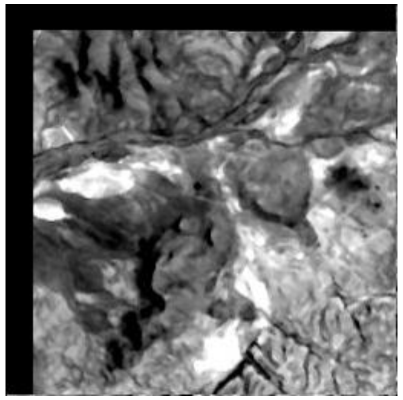


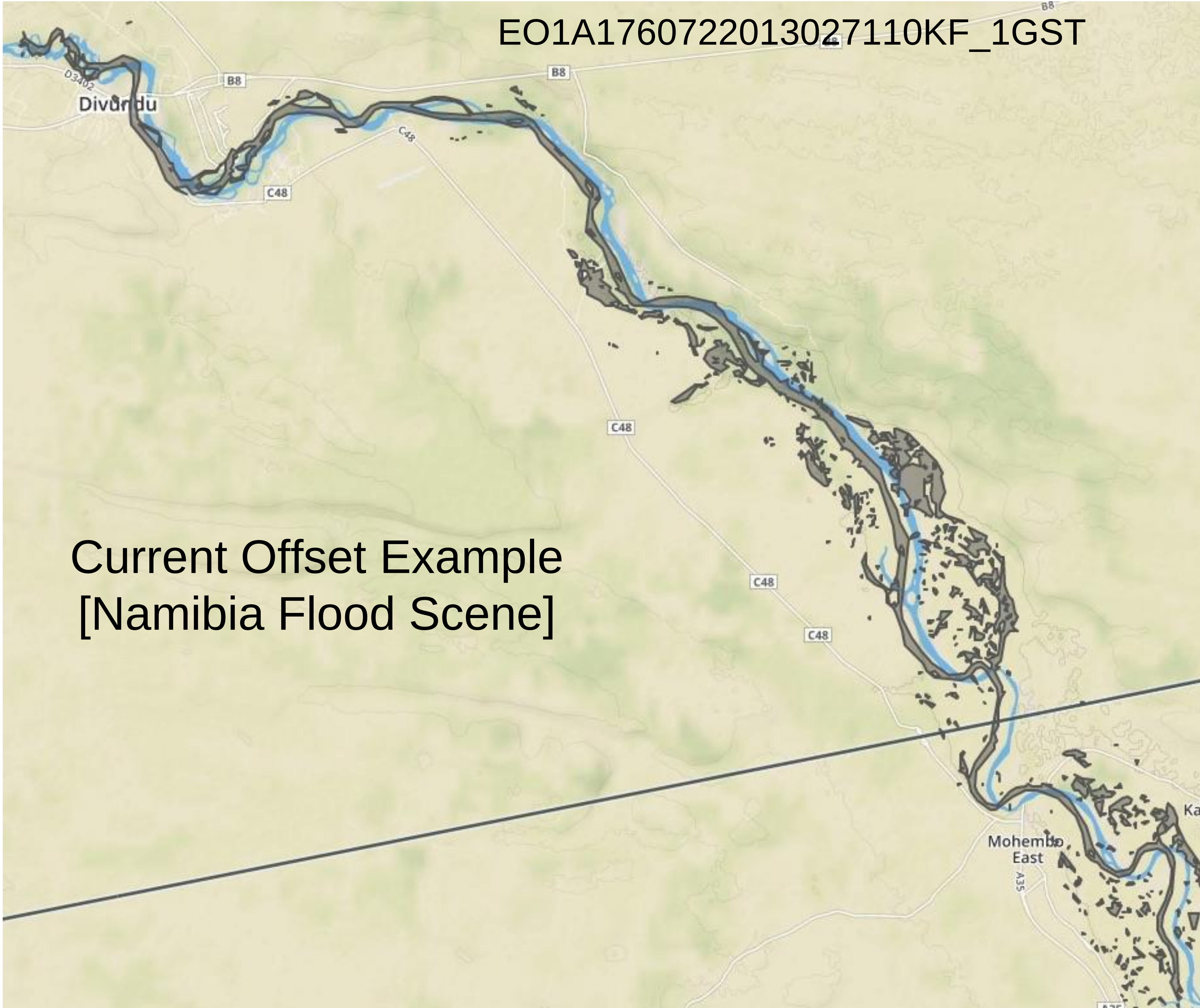
Image Co-Registration For Onboard Low Latency Products

Pat Cappelaere
Vuong Ly
Dan Mandl
Jacqueline LeMoigne

Goals

- **Co-register** hyperspectral/multispectral bands **onboard** within a **few seconds**
- Proof-of-Concept using EO-1 ALI L1GST
- 9 Bands @30m resolution 16.4MB TIF Int16 ~(2231x3671)
- **Rationale**: Current EO-1 L1GST Data Product Offset Not Good for Low Latency Products

EO1A1760722013027110KF_1GST



Current Offset Example
[Namibia Flood Scene]

Previous Workflow

- For One EO-1 ALI L1GST Scene:
  - Find/Make [30-150] Chips from Landsat GLS2000 (Landsat Band 7-- 2.1 to 2.3 microns)
  - Subset Matching Tile(s) from EO-1 ALI Scene (EO-1 Band 8 --- 1.2 to 1.3 microns)
    - ✓ Issue was whether we are detecting the same features using these bands
  - Apply Filter, Linear Stretch and Convert to Byte
  - Apply co-registration (ureg) -> Tx,Ty [,Theta,Scale] for each tile
  - Using Median RST to Determine Average Transformation
  - Apply Transformation To Band File(s)
  - Visual Assessment

Current Workflow

- Make 1 Best Chip from Best Available Landsat L8 scene from the Amazon Cloud (latest, least cloudy one) (Landsat 8 Band 7 --- 2.107 to 2.294 microns)
- Subset Matching Tile from EO-1 ALI Scene (EO-1 Band 10-- 2.08 to 2.35 microns)
- Apply co-registration (ureg) -> Tx, Ty [do not use Theta, Scale]
  - Ignoring theta because theta is small, would normally be a expensive transform to apply
  - Ignoring scale because it is always one
  - Translation is easy, only change origin in header thus is can be very fast
- Generate Automated Metrics (to assess the results of the transformation)
- Ported code to various flight testbeds

Timing Metrics

- Assumptions:
  - Only timing the coregistration portion, not acquisition of data
  - EO-1 scene(16.4 Mbytes per band) with loaded chip
- Results to subset an EO-1 tile and determine transformation parameters
  - 0.38 s (Mac OSX)
  - 3.44 s (Flight Testbed – 667MHz ARM ZC702 (1 Core)
  - 6.78 s (Flight Testbed – 1GHz TILE-Gx36 (1 Core)
  - 33.18 s (Flight Testbed – 864MHz TILEPro64 (1 Core)
- Time to Apply RST Transformation to One TIF Band (x 9 bands)
  - < 0.01s

Quantifying Registration Errors

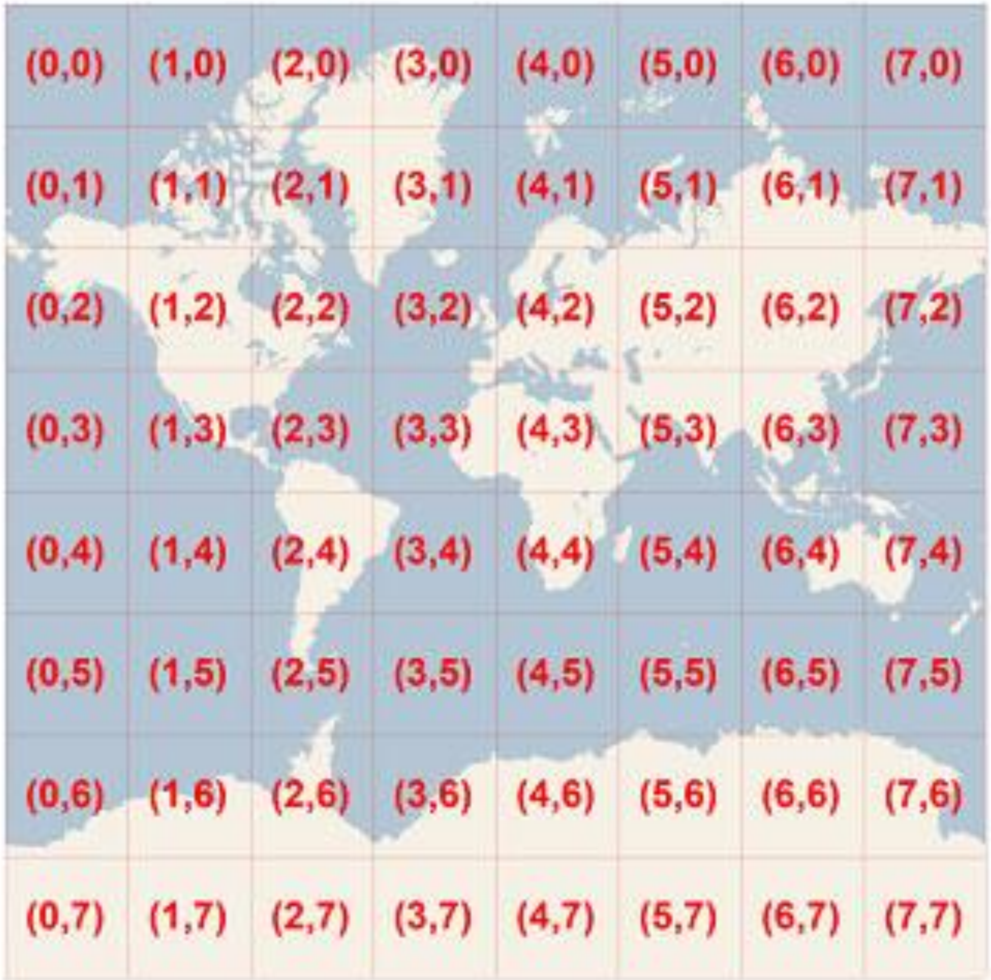
- **Problem**: How To Automatically Quantify The Mis-registration Before and After Registration?
- Root Mean Square Error (RMSE) (Coulter & Stow 2012)
  - RMSE = $\sqrt{(\sum^{N}_{1} ((x_1-x_2)^2 + (y_1-y_2)^2)/N)}$
  - Using Automatically Detected Top 10 Features between Chip and Matching Tile
  - **How**: using SURF (Speeded-Up Robust Features), Brute Force Matcher, and error catching using RANSAC
  - Compute RMSE on Matched Features
- Use Several Algorithms and Average Results (WORK IN PROGRESS)

Chip Determination

- **Rationale**:
  - Avoid Onboard Database (210GB)
  - Avoid Real-time Search for Chip(s)
  - Use Direct Retrieval of Chip if Stored Onboard or Load it With Command Load With Option to store or not

QuadKey Tiles
(similar algorithm as used by Bing/Google)

- Used to identify chip subset for the specified Landsat 8 scene and the matching EO-1 scene
- Square Tiles
- Every Target Belongs to a QuadTile
- QuadTile Easy To Find (Algorithm)
- QuadTiles have QuadKeys (aka Name)
- Quad-Neighbors Easy To Find



Example: Target: latitude, longitude [8.01,98.56]
-> Quadkey12 Bounding box =“132221300201” [7.88,98.44,8.06,98.66]

Quad-Tile Size Trade

Hash len	Size km	Res m/px	GLS Pixels
11	19.4	75.7	648x648
12	9.7	37.85	324x324
13	4.8	18.9	162x162

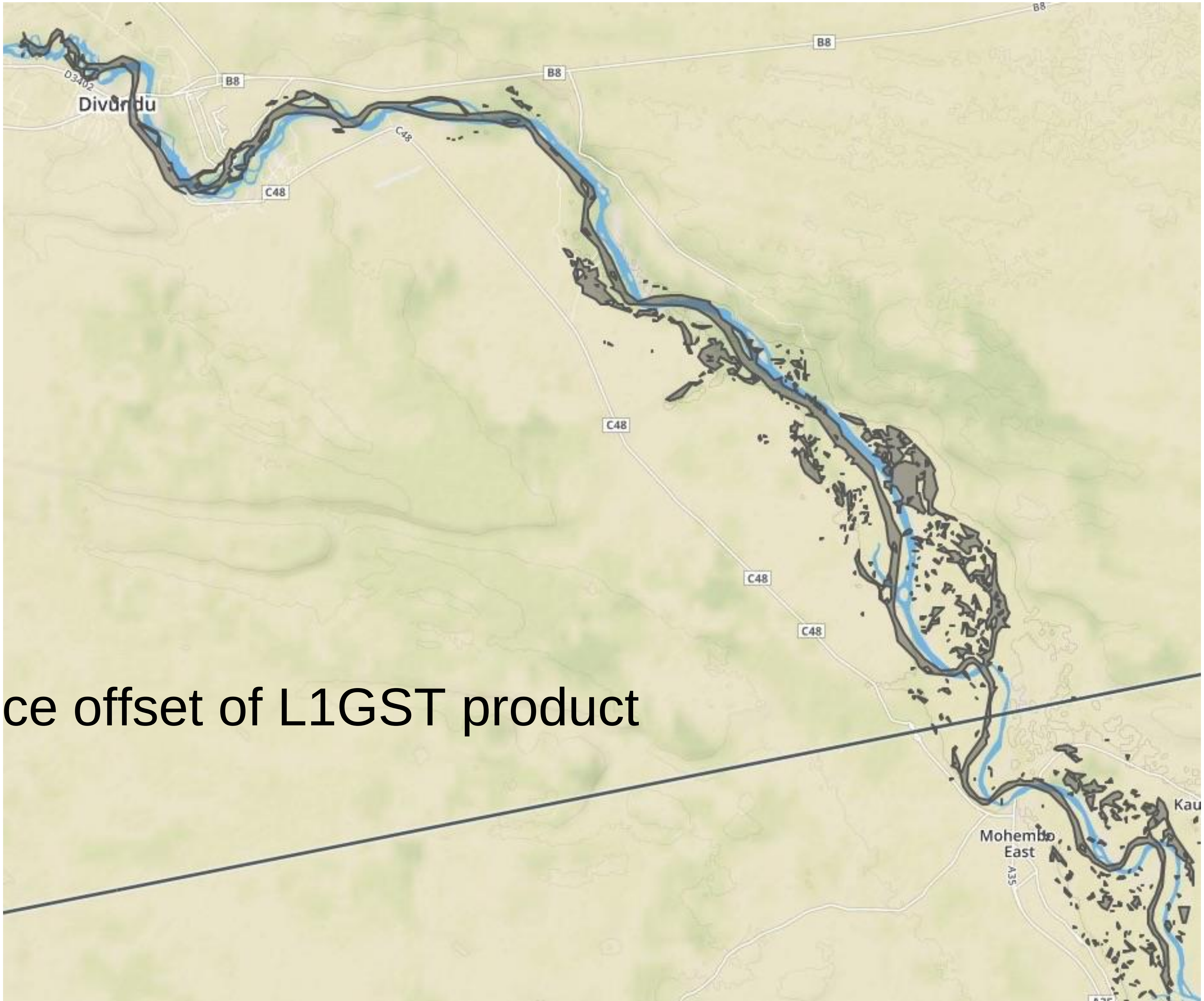
Co-registration Namibia Example

Flood Map Workflow

- EO-1 Scene from USGS (10 bands - **92MB compressed**)
  - ➢ Generate Water Surface Extent
  - ➢ Apply HAND mask
  - ➢ Apply Cloud mask
  - ➢ Generate Final Raster
  - ➢ Generate Compressed Topojson Vector Product **(24KB**)
    - ✓ Aka low latency product
- Display on Browser using geojson.io

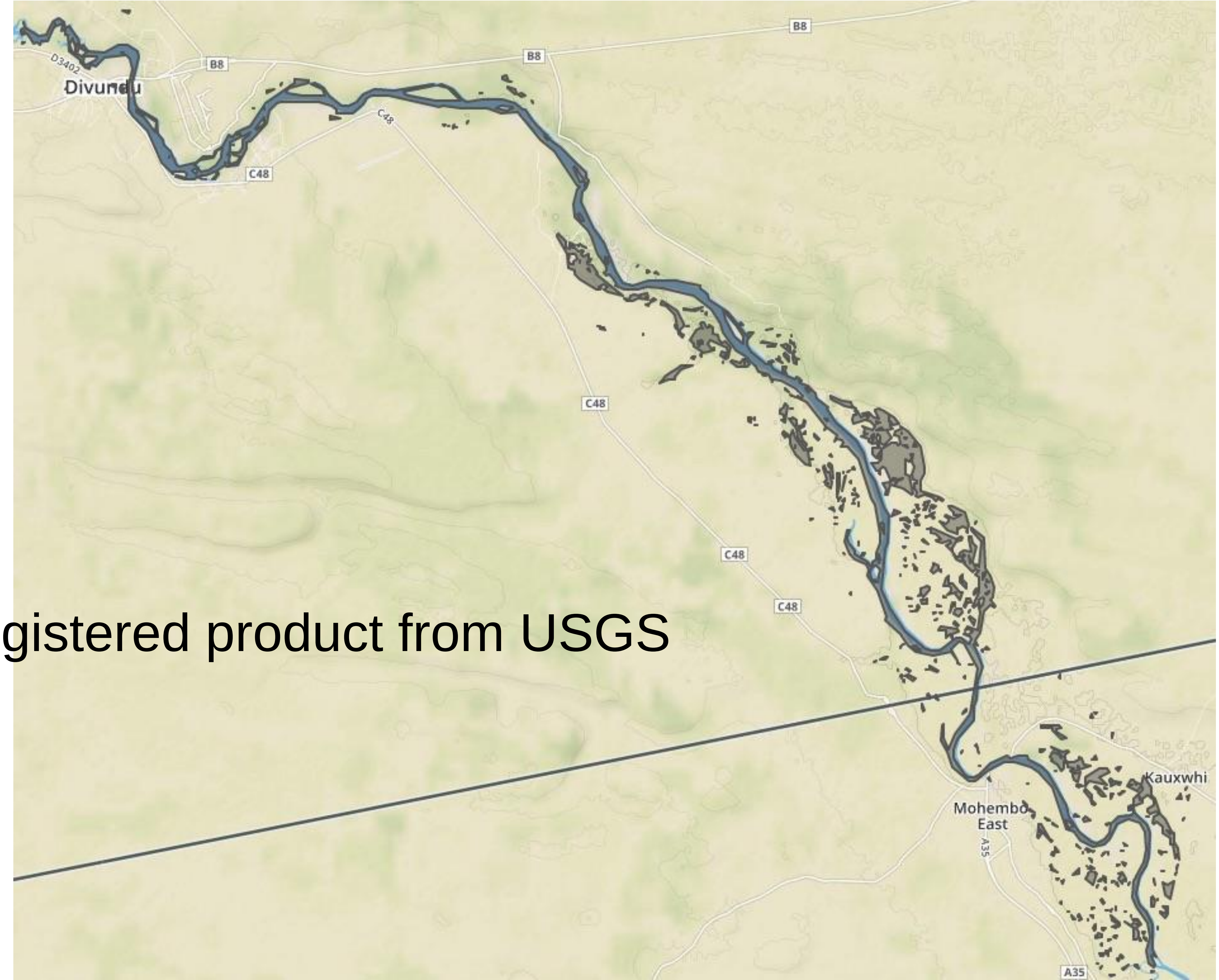
USGS EO-1 L1GST :

EO1A1760722013027110KF_1GST



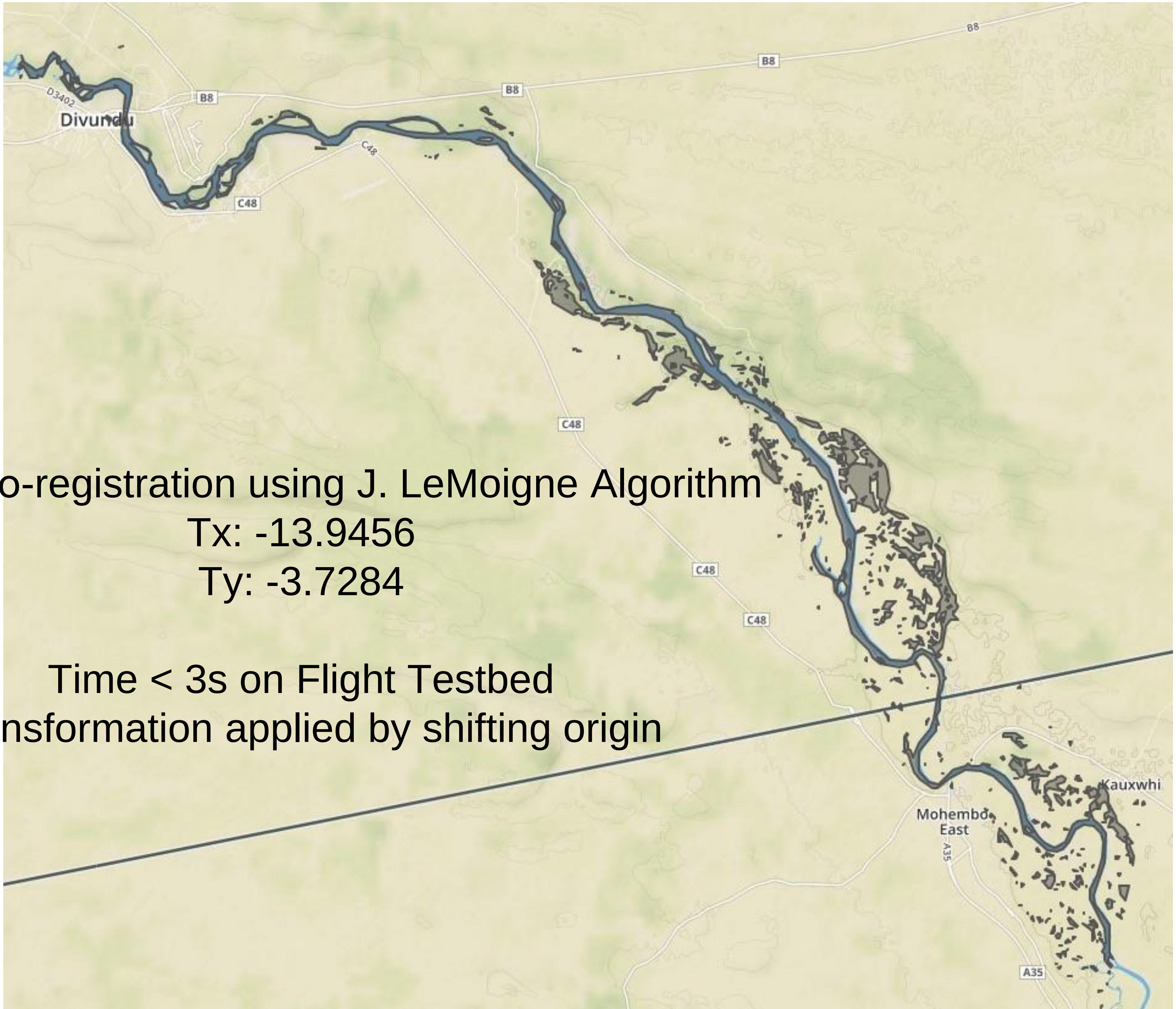
Notice offset of L1GST product

USGS EO-1 L1T : EO1A1760722013027110KF_1T



Co-registered product from USGS

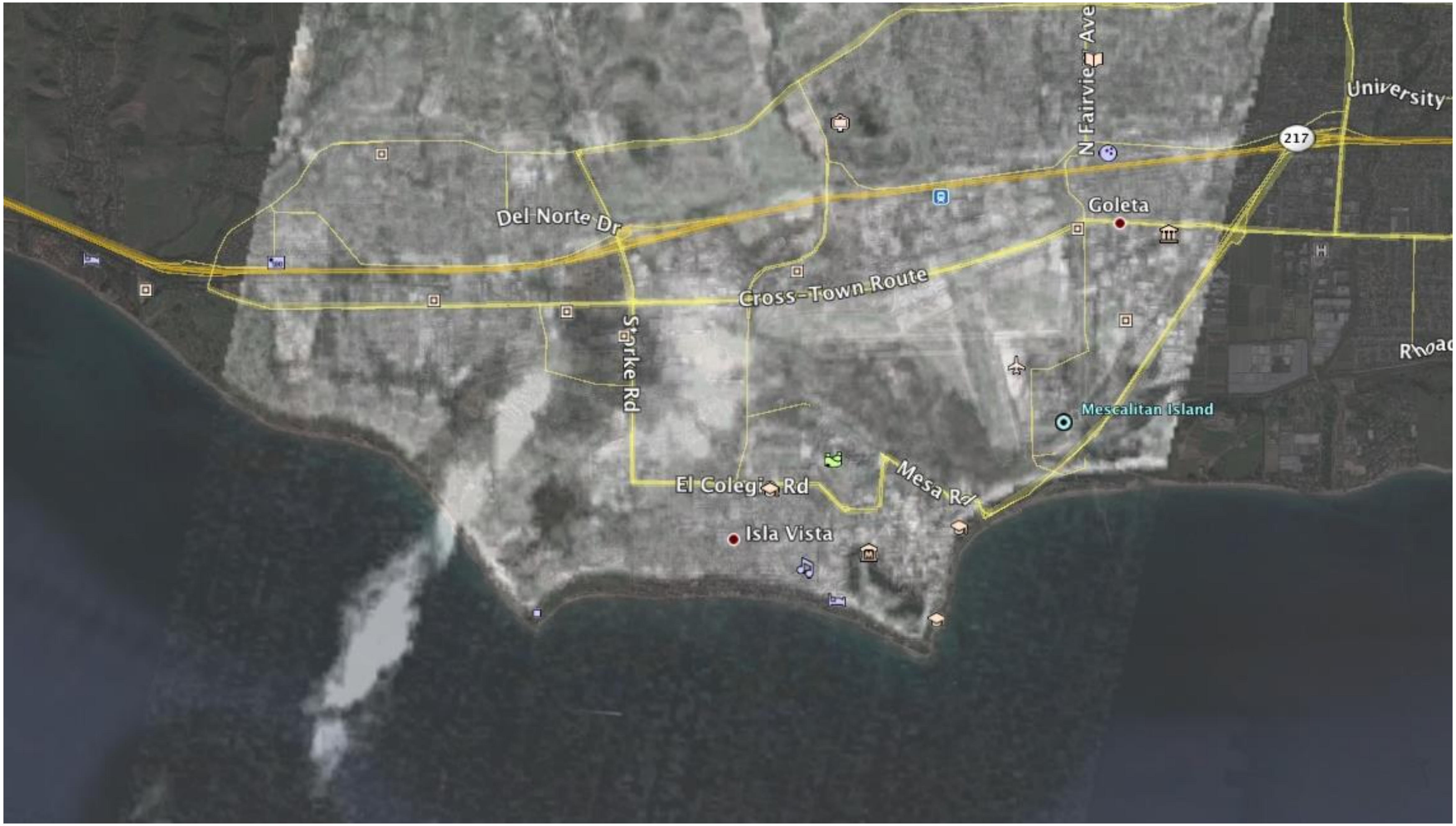
L1GST After Co-Registration (one L8 Tile)



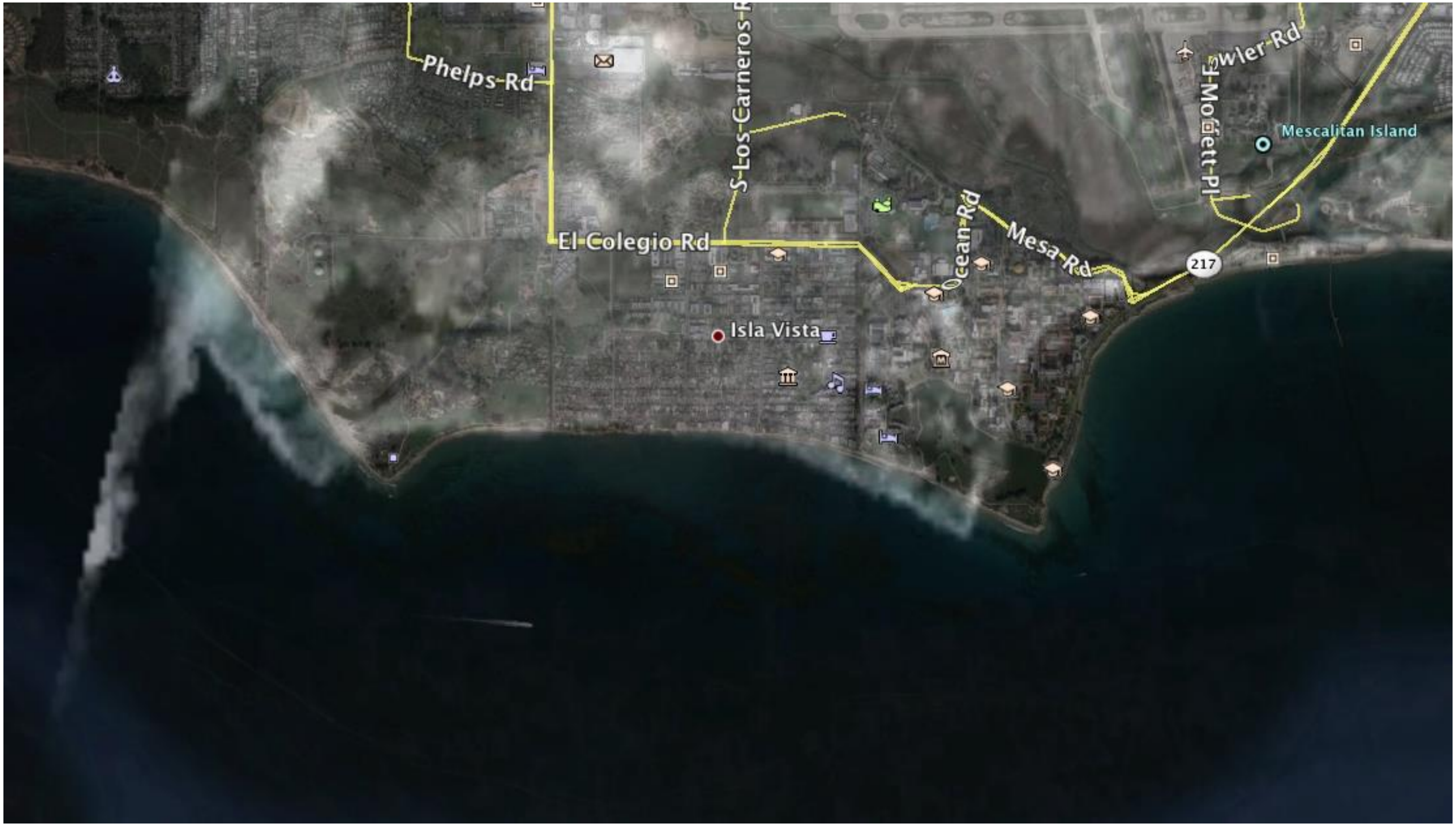
“Flight” co-registration using J. LeMoigne Algorithm
Tx: -13.9456
Ty: -3.7284

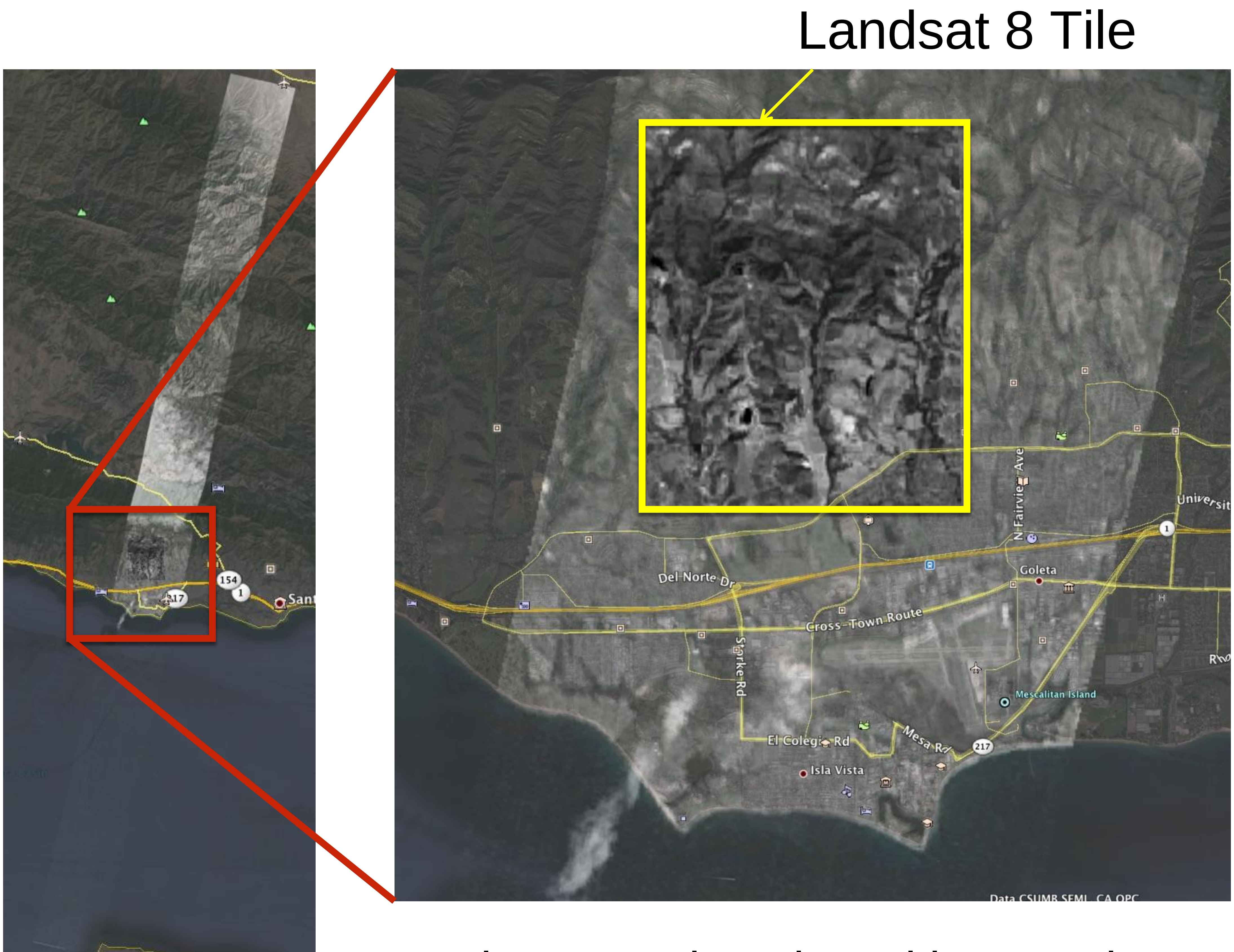
Time < 3s on Flight Testbed
Transformation applied by shifting origin

Example of Hyperion L1G Misregistration on Google Earth



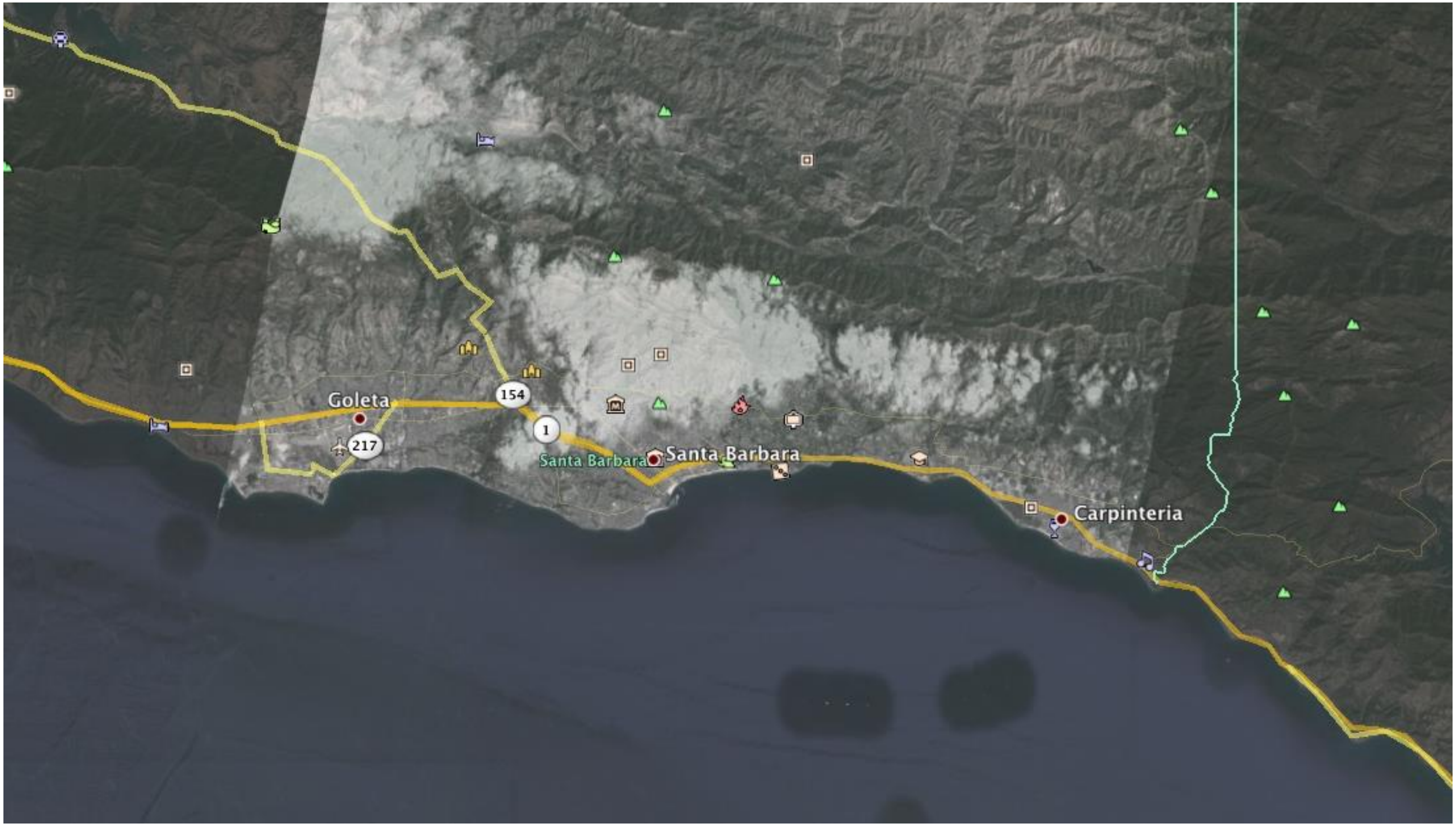
ALI L1GST Misregistration on Google Earth





Hyperion Coregistration with 1 Landsat 8 Tile
Tx: -2.743618 Ty -7.184056

ALI L1G Coregistered (Same Scene)



Tx: -13.772506 Ty: -0.815789

Possible Future Work

- Finish/Validate Automated Metrics
- Improve Tile Selection Algorithm
  - *Shannon Entropy (Image Histogram) to use Wavelet Entropy*
- Automate EO-1 Processing Flow On Matsu Cloud
- Port to CHREC Space Processor (Xilinx MicroZed)

pat@cappelaere.com

(410) 340-4868