Data Products on Clou

Vuong Ly

HysplRI Symposium
Ground Data Processing and Distribution sion
June 5, 2014

EO1 Cloud Computing

Technologists
NASA Investigators
Disaster Responders

Matsu Cloud

http/ftp server

Level 1R and Level 1G Processing for ALI & Hyperion

Co-registration with Landsat GLS

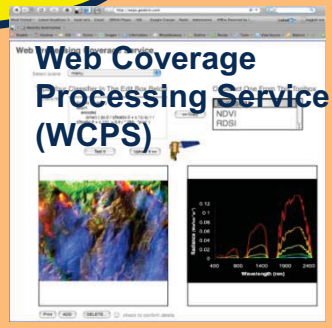
Atmospheric Correction for Hyperion

- Open Stack-based Elastic Cloud SW
- 300+ core processors
- 500+ TB of storage
- 10 Gbps connection to GSFC
  - Being upgraded to 100 Gbps
- Hadoop Tiling/MapReduce/Accumulo
- Supplied by Open Cloud Consortium
- Open Science Data Cloud Virtual Machines & HTTP server to VM's

Multi year data product archive

Starlight 100 Gigabit Ethernet Exchange

Hyperion and ALI Level 0 Processed data from GSFC, EO1 MOC



Web Coverage Processing Service (WCPS)

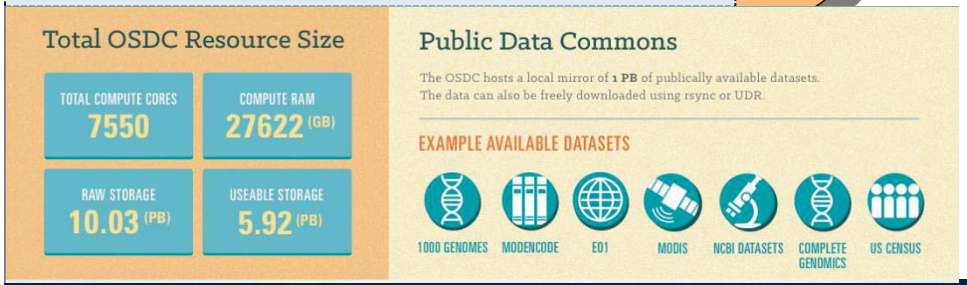
Namibia Flood Dashboard

Joyent Cloud

EO-1 GeoBliki

EO-1 GeoBPMS

- Ruby on Rails
- 3 processors
- 3TB of storage



EO1 Cloud Computing

- Data is available publicly and instantaneously at ftp://matsu.opencloudconsortium.org
- Namibia Flood Dashboard http://matsu.opencloudconsortium.org/namibiaflood
- Web Coverage Processing Service http://matsu.opencloudconsortium.org/wcps

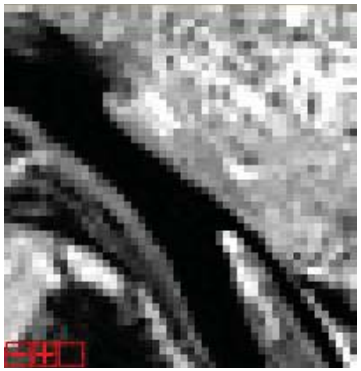
Co-registration with Landsat GLS

- Global Land Survey Maps - A collection of Landsat-type satellite images from USGS
  - Near complete global coverage
  - Orthorectified
  - Each image has cloud cover of less than 10%
  - Four versions: 1970, 1990, 2000, and 2005

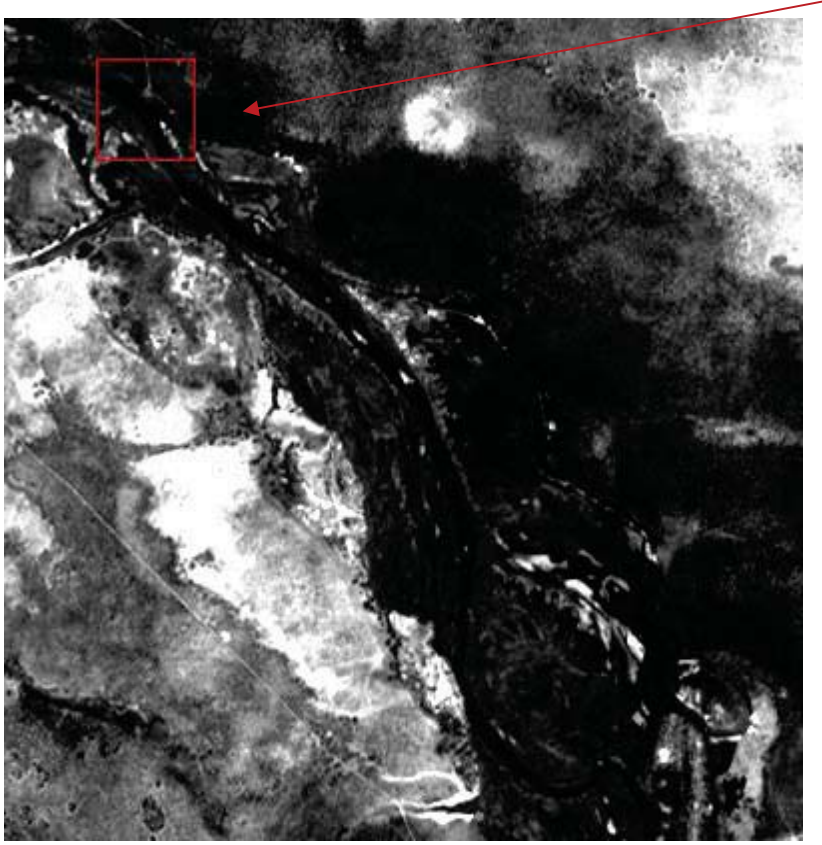
- Ground truth for the registration programs was drawn from the GLS 2000 and can be updated when the GLS 2010 is completed

- http://landsat.usgs.gov/science_GLS.php

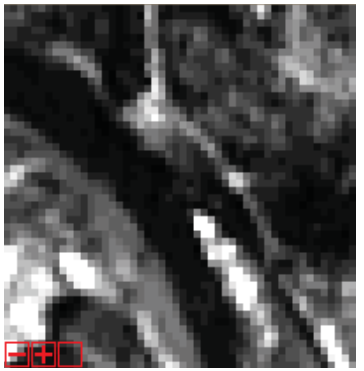
Chip Registration



Overlapping chip
from database



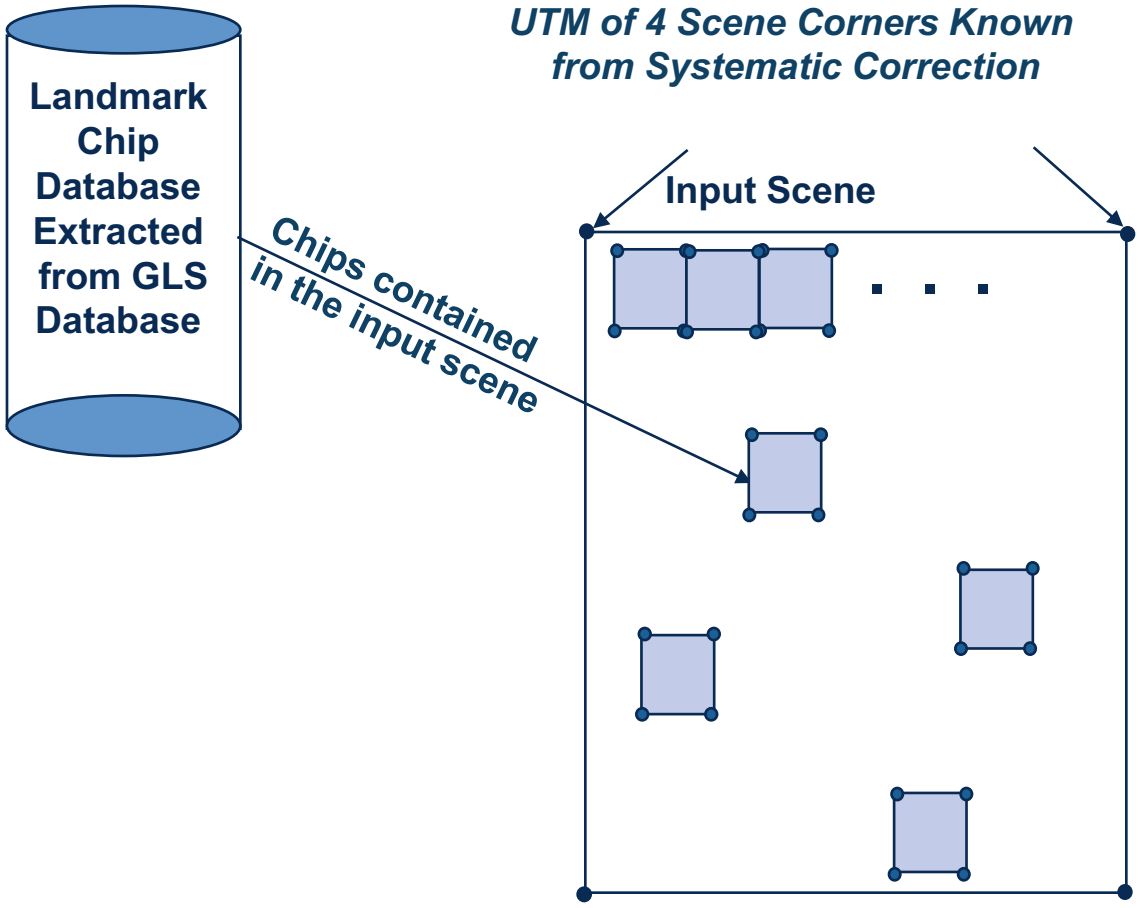
Area in EO1 scene where chip was extracted



Chip extracted
from EO1 scene

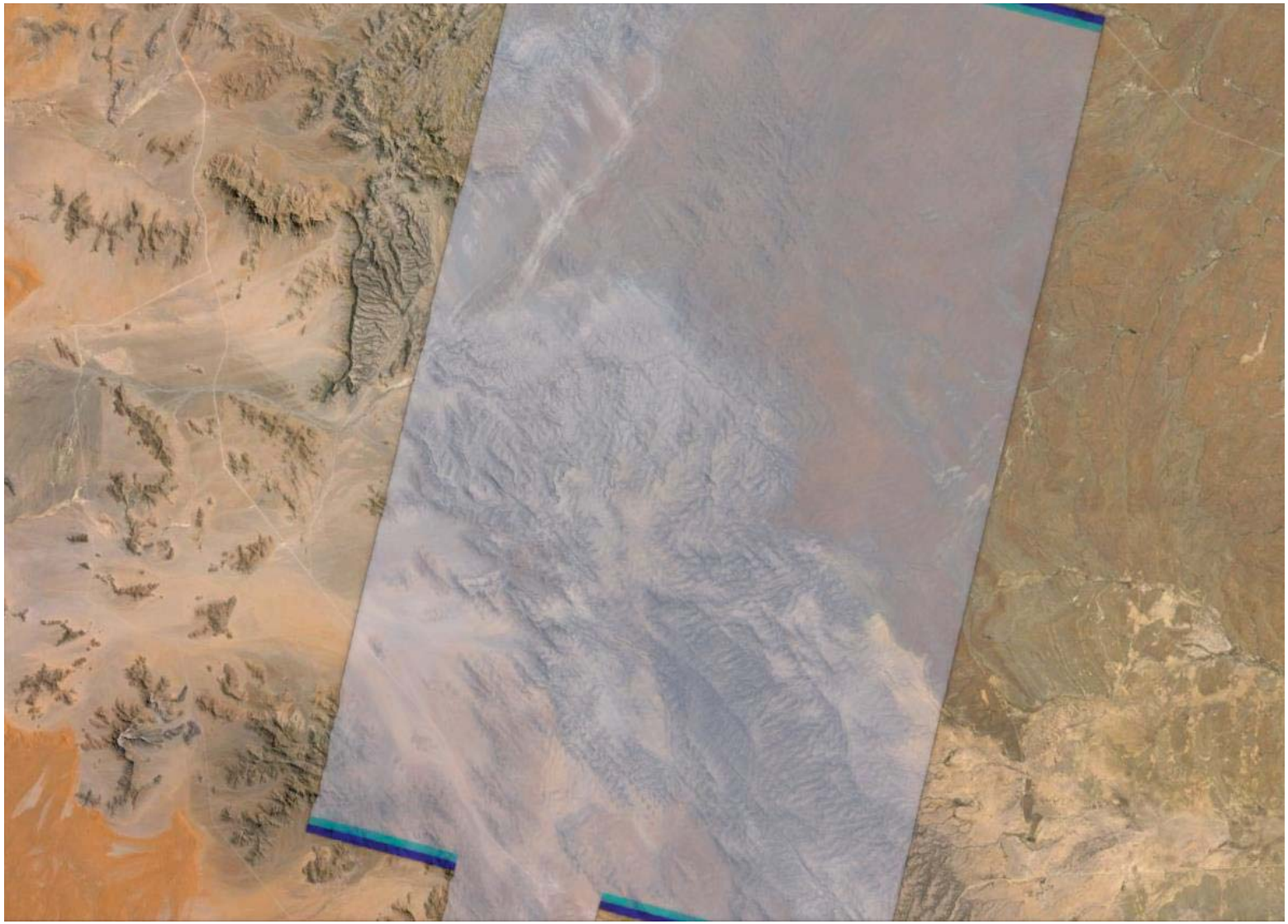
Currently "chip database" created (in a brute-force fashion) by extracting successive 256x256 sub-images of all GLS scenes and storing them according to path and row

Automatic Registration of EO1 Scenes Using Global Land Survey (GLS) Database

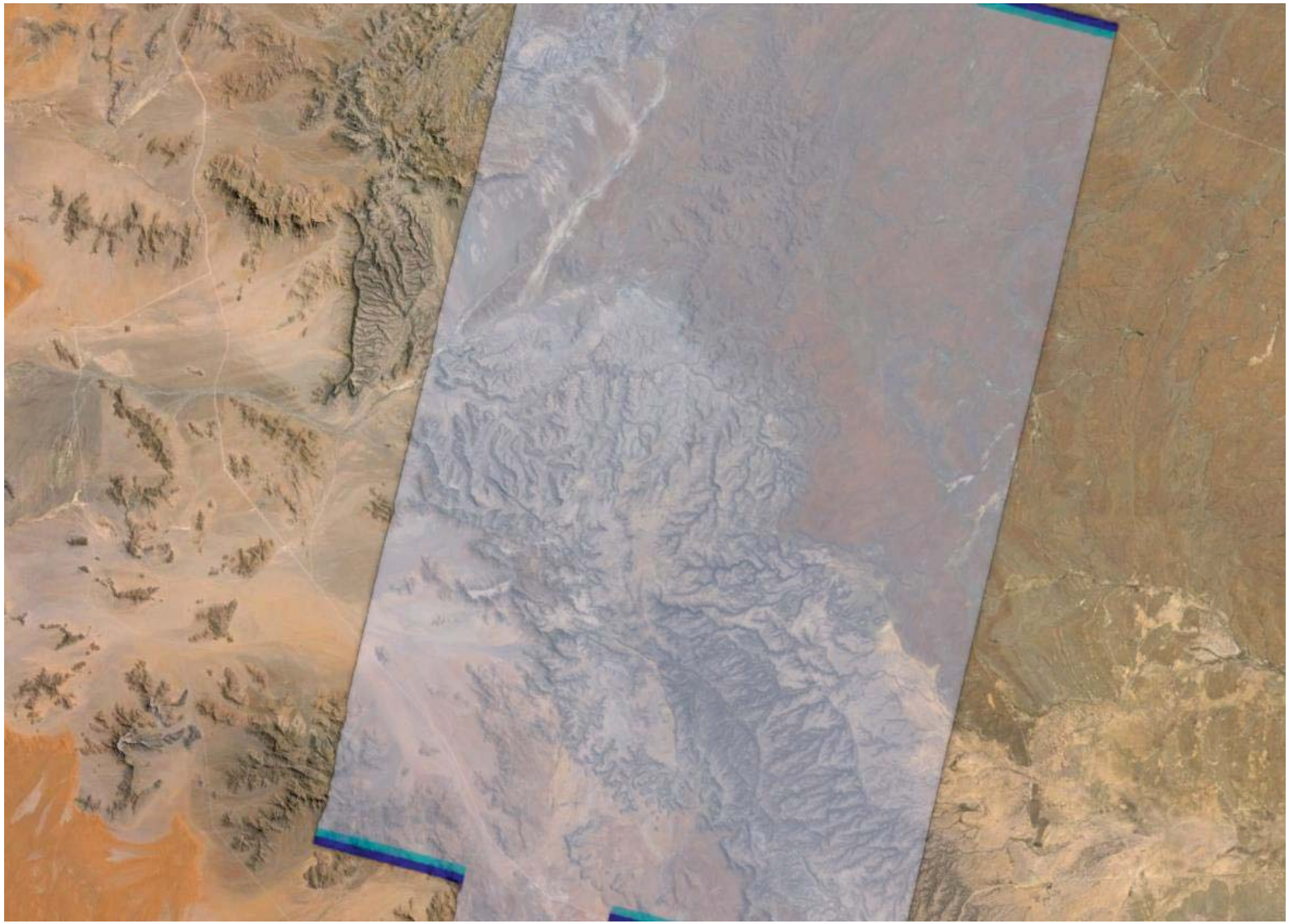


1. *Find Chips that correspond* to the Incoming Scene
2. *For Each Chip, Extract Window* from input scene using UTM coordinates
3. *Eliminate Windows* with insufficient information
4. *Smooth and Normalize* gray values of both Chip and Window using a Median Filter
5. *Register each (Chip,Window) Pair* using a wavelet-based automatic registration: get a local rigid transformation for each pair
6. *Eliminate Outliers*
7. *Compute Global Rigid Transformation* as the median transformation of all local ones
8. *Compute Correct UTM* of 4 Scene Corners of input scene
9. *If desired, Resample* the input scene according to the global transformation

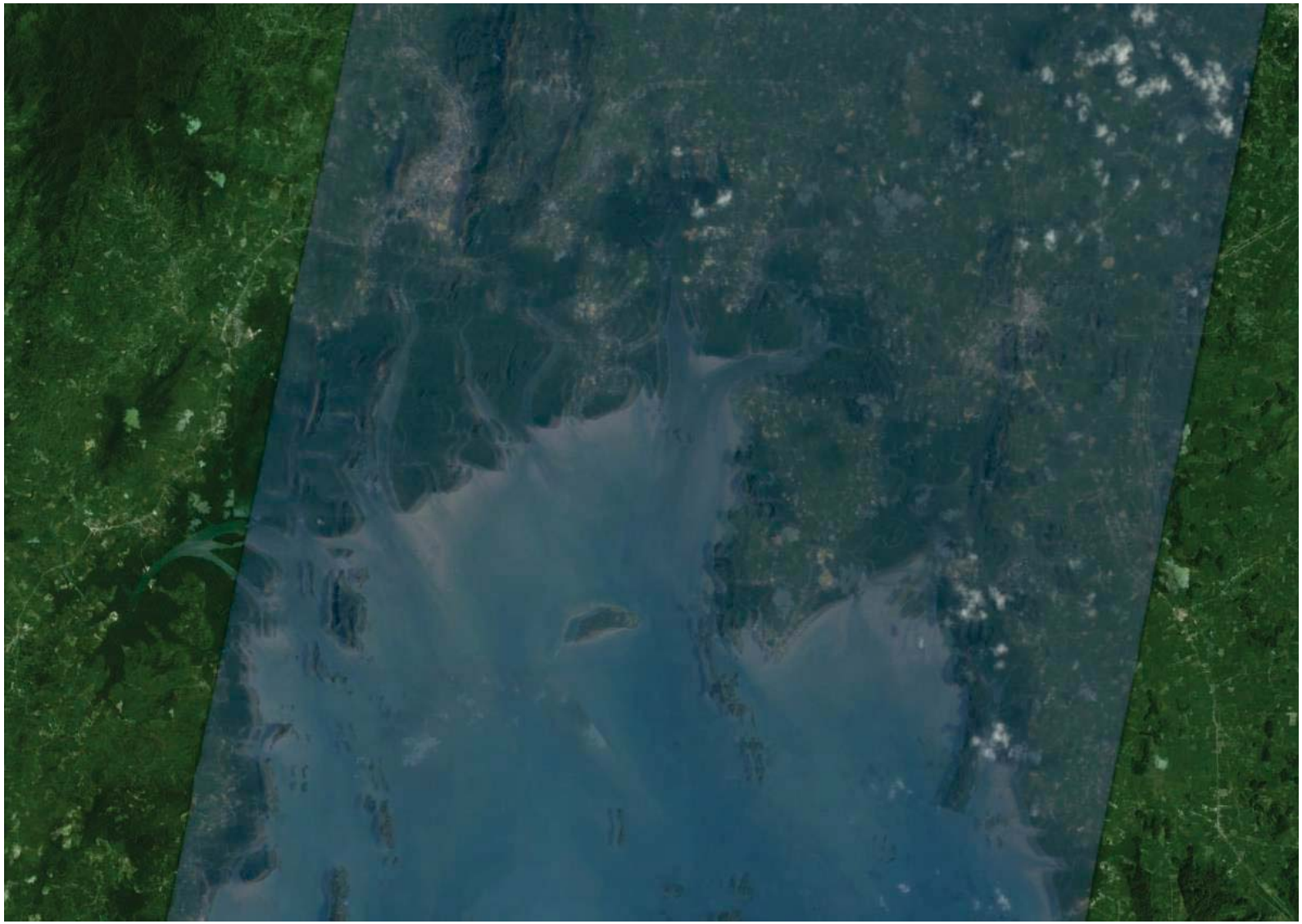
Scene 1 Before Automatic Registration Superimposed onto Goggle Earth



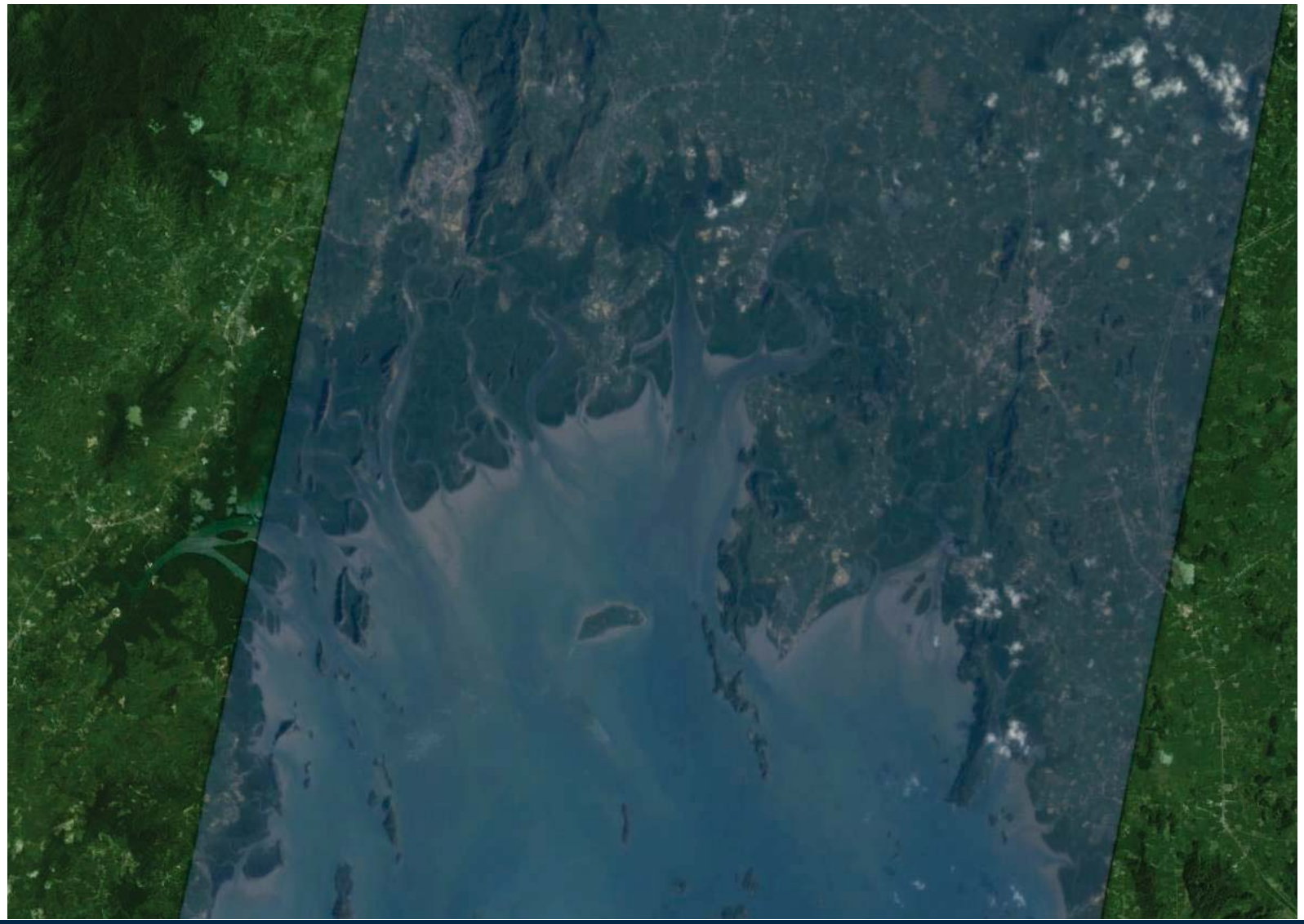
Scene 1 After Automatic Registration Superimposed onto Goggle Earth



Scene 2 Before Automatic Registration Superimposed onto Goggle Earth



Scene 2 After Automatic Registration Superimposed onto Goggle Earth



Conclusions and Future Work

- Results visually acceptable
- Computations very fast and real-time
- RMS still too high (Translation errors between 0.4 and 2.5 pixels) because:
  1. Chips and windows need to be pre-selected based on the information content (e.g., using an entropy measure)
     - Registration would be more accurate because transformation would only be computed on pairs that have a significant amount of features
     - Registration would be faster because less local registrations
     - Chip database would be smaller to be stored onboard
  2. Global transformation should be computed by taking the list of original corners coordinates of each window and their corresponding corrected coordinates, and treat them as a list of ground control points and their corresponding points => after outlier elimination, global transformation can be computed using a rigid, an affine or a polynomial transformation.
  3. Masks for clouds and water should be included, so registration would not use cloud or water features that are often unreliable
- Onboard, computations can be performed on SpaceCube or hybrid processor