Georegistration of Earth Observing-1 (EO-1) Data Using Global Land Survey (GLS) Maps

Jacqueline Le Moigne[1], Patricia Sazama[1&2],
Steve Swanson[3], Vuong Ly[1] and Dan Mandl[1]

1. *NASA Goddard Space Flight Center, Software Engineering Division*
2. *University of Maryland, Computer Science Department*
3. *Princeton University, Computer Science Department*

Background – Image Registration

- What is Image Registration?
  "Exact pixel-to-pixel matching of two different images or matching of one image to a map"
- Navigation or Model-Based Systematic Correction
  - Orbital, Attitude, Platform/Sensor Geometric Relationship, Sensor Characteristics, Earth Model, etc.
- Image Registration/Feature-Based Precision Correction
  - Navigation within a Few Pixels Accuracy
  - Image Registration Using Selected Features (or Control Points) to Refine Geo-Location Accuracy
- Image Registration as a Post-Processing or as a Feedback to Navigation Model

Image Registration Frameworks

- Mathematical Framework
  - I1(x,y) and I2(x,y): images or image/map
    - find the mapping (f,g) which transforms I1 into I2:

      I2(x,y) = g(I1(fx(x,y),fy(x,y))

      » f : spatial mapping

      » g: radiometric mapping
  - Spatial Transformations "f"
    - Translation, Rigid, Affine, Projective, Perspective, Polynomial, …
  - Radiometric Transformations "g" (Resampling)
    - Nearest Neighbor, Bilinear, Cubic Convolution, ...

- Algorithmic Framework (Brown, 1992)
  1. Feature Extraction
  2. Feature Matching (Similarity Metrics & Matching Strategy)
  3. Image Resampling (if needed)

Image Registration Components

0 Pre-Processing

- Cloud Detection, Region of Interest Masking, …

1 Feature Extraction ("Control Points")

- Gray Levels, Salient Points (e.g., Edges, Edge-like such as Wavelet Coefficients, Corners), Lines, Contours, Regions, Scale Invariant Feature Transform (SIFT), etc.

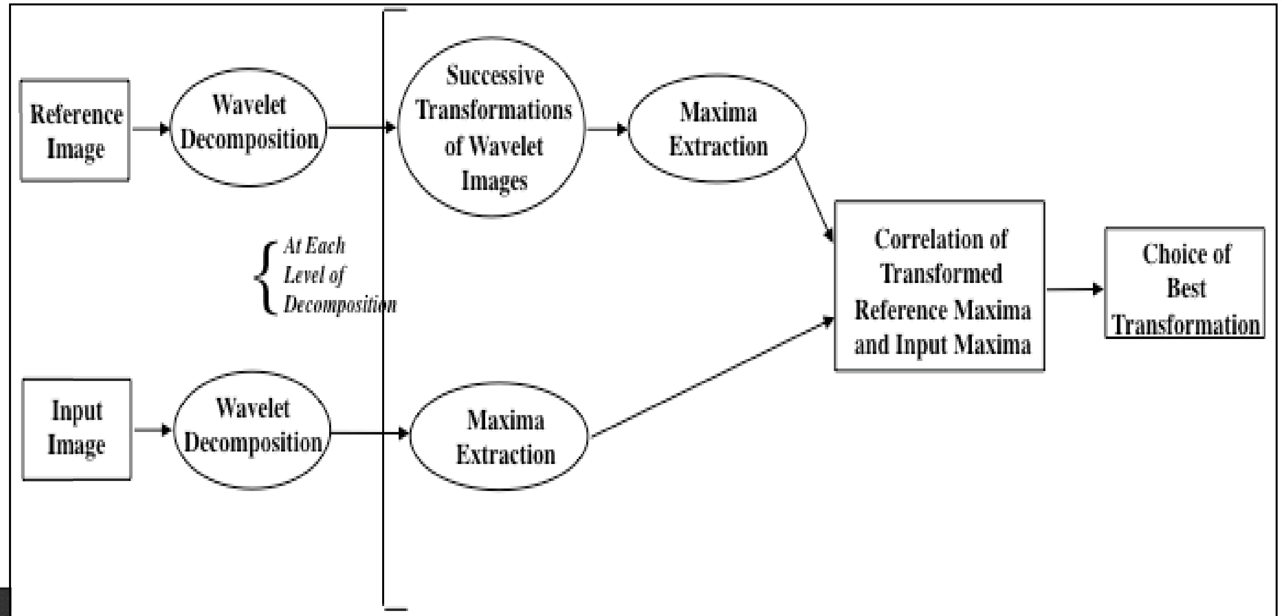
2 Feature Matching

- Choice of Spatial Transformation (function f: a-priori knowledge)
- Choice of Search Strategy :
  - Global vs Local, Multi-Resolution, Optimization, ...
- Choice of Similarity Metrics
  - L2-Norm, Normalized Cross-Correlation, Mutual Information, Hausdorff Distance, …

3 Remapping/Resampling (function g: if necessary)

Wavelets and Wavelet-Like Features for Image Registration

Figure 1
Original Image



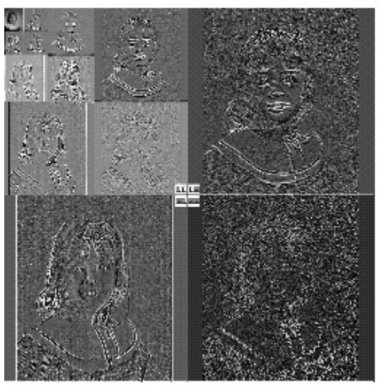


Figure 2
Wavelet Coefficients Corresponding to Figure 1

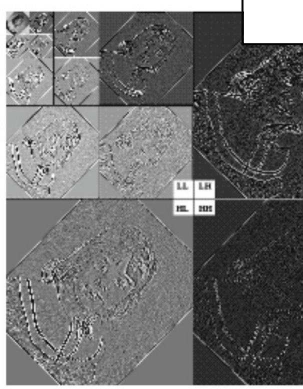
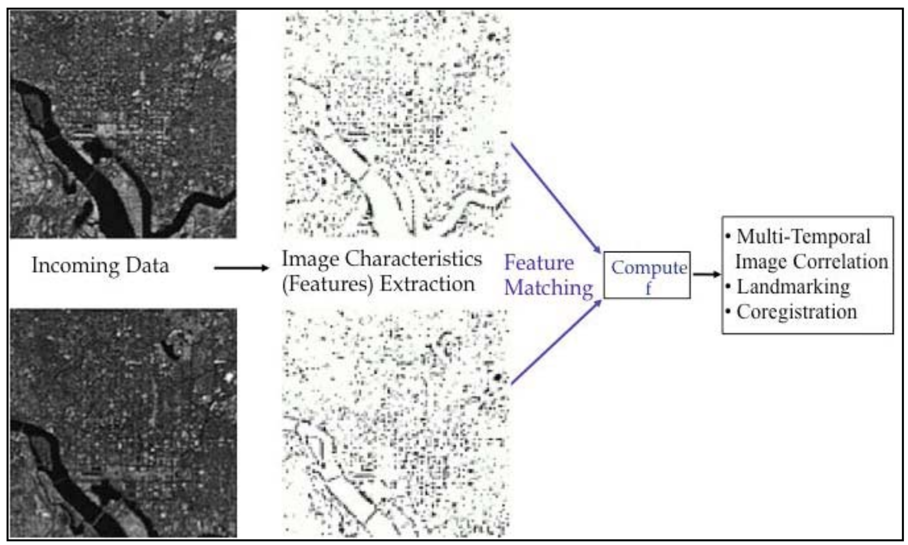


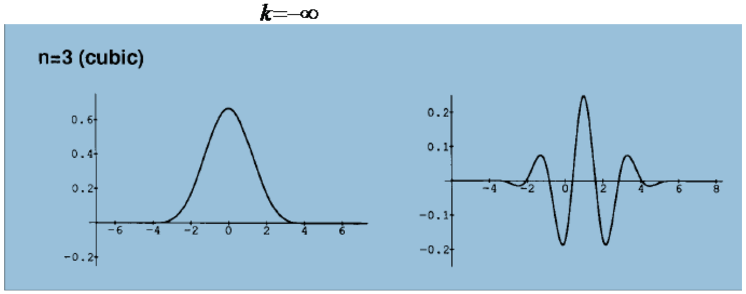
Figure 3
Wavelet Coefficients Correspond to Figure 1 rotated 44 degrees



Rotation- and Translation-Invariant Representations

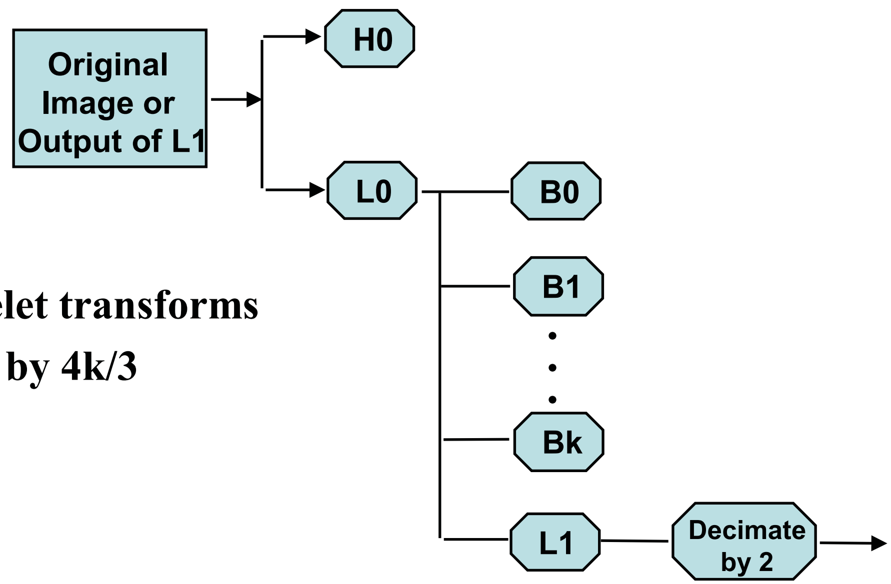
- **Spline Wavelets [Battle & Lemarié; Unser et al]**

$$V_i^n = \{g_i^n(x) = \sum_{k=-\infty}^{+\infty} c_i(k)\varphi^n(2^{-i}x - k), x \in \Re, c_i \in l_2\}$$



with scaling function $\varphi^n(x) = \sum_{k=-\infty}^{+\infty} p(k)\beta^n(x-k)$ p arbitrary invertible convolution operator or filter, and $\beta^n(x)$ is a B-spline of order n (can be constructed by repeated convolution of B-Spline of order 0)

Example of B-Spline Scaling Function and Associated Wavelet



- **Simoncelli et al**
  - **Relax critical sampling condition of wavelet transforms**
  - **Provides an overcomplete representation by 4k/3**

Matching Strategies

- Exhaustive Search
- Fast Fourier Transform
- Optimizations:
  - Gradient Descent $\begin{bmatrix} \sum f_x^2 & \sum f_x f_y & \sum R f_x \\ \sum f_x f_y & \sum f_x^2 & \sum R f_y \\ \sum R f_x & \sum R f_y & \sum R^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \sum (f-g) f_x \\ \sum (f-g) f_y \\ \sum (f-g) R \end{bmatrix}$
  - Modified Marquart-Levenberg: hybrid optimization approach between a pure gradient-descent approach and a more powerful but less robust Gauss-Newton method, implemented in a multi-resolution fashion
  - Spall's Simultaneous Perturbation Stocchastic Approximation (SPSA): based on gradient approximation computed from objective function (200 iterations)
- Robust Feature Matching
  - Hierarchical Subdivisions of Search Space
  - Pruning of Search Space

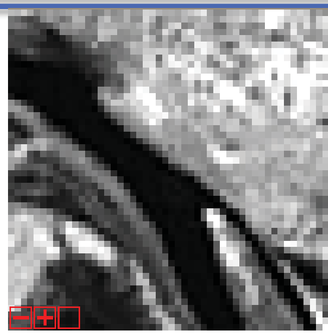
Matching Strategies

- Exhaustive Search
- Fast Fourier Transform
- Optimizations:
  - Gradient Descent
    $$\begin{bmatrix} \sum f_x^2 & \sum f_x f_y & \sum R f_x \\ \sum f_x f_y & \sum f_x^2 & \sum R f_y \\ \sum R f_x & \sum R f_y & \sum R^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \sum (f-g) f_x \\ \sum (f-g) f_y \\ \sum (f-g) R \end{bmatrix}$$
  - Modified Marquart-Levenberg: hybrid optimization approach between a pure gradient-descent approach and a more powerful but less robust Gauss-Newton method, implemented in a multi-resolution fashion
  - Spall's Simultaneous Perturbation Stocchastic Approximation (SPSA): based on gradient approximation computed from objective function (200 iterations)
- Robust Feature Matching
  - Hierarchical Subdivisions of Search Space
  - Pruning of Search Space

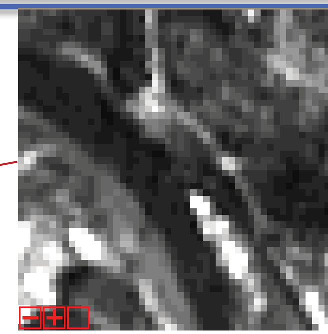
Global Land Survey (GLS) Maps

- 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
- Current Ground Truth or "Reference Chips" extracted from the GLS 2000 (can be updated when the GLS 2010 is completed)
- Reference Chips of size 256 X 256
- http://landsat.usgs.gov/science_GLS.php

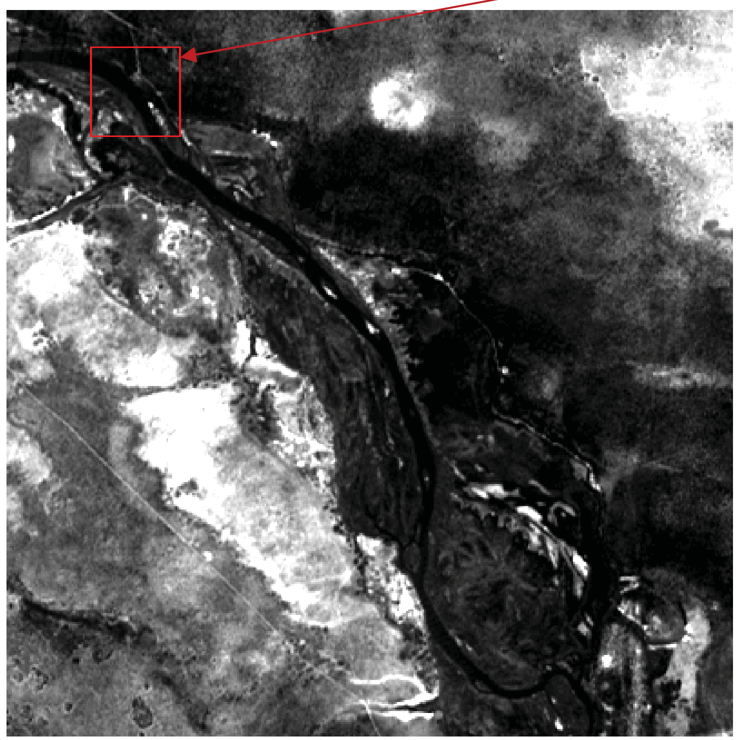
Chip Registration



Overlapping chip from database



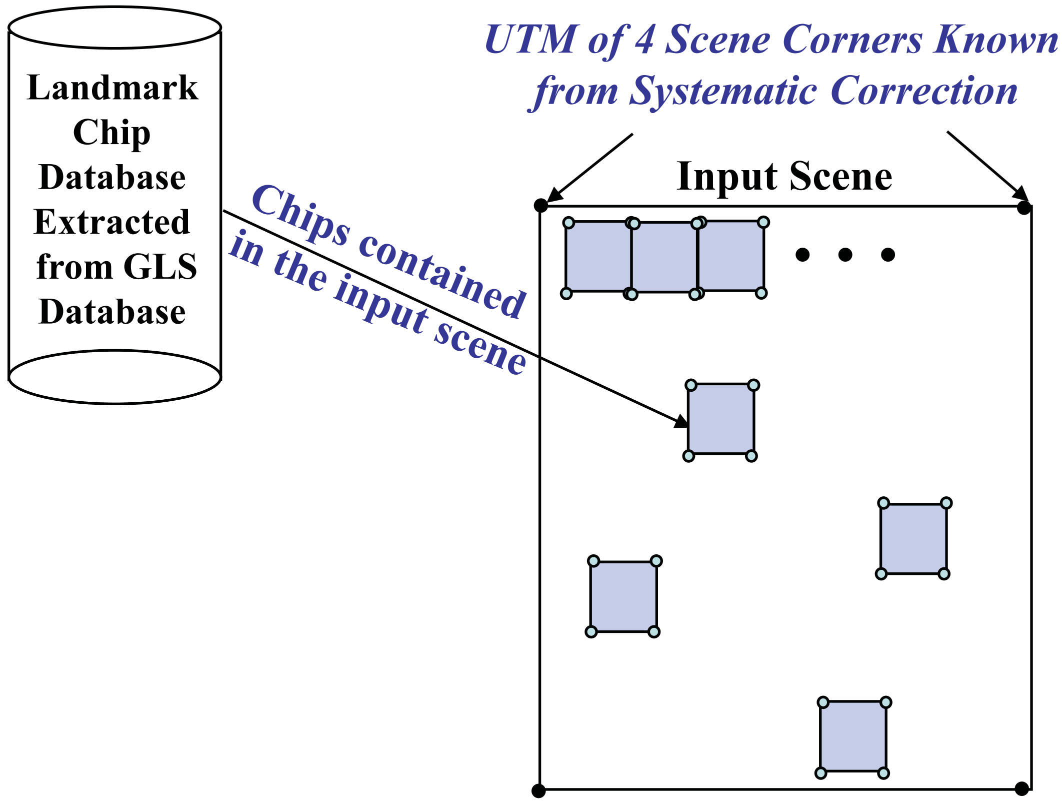
Chip extracted from EO1 scene



Area in EO1 scene where chip was extracted

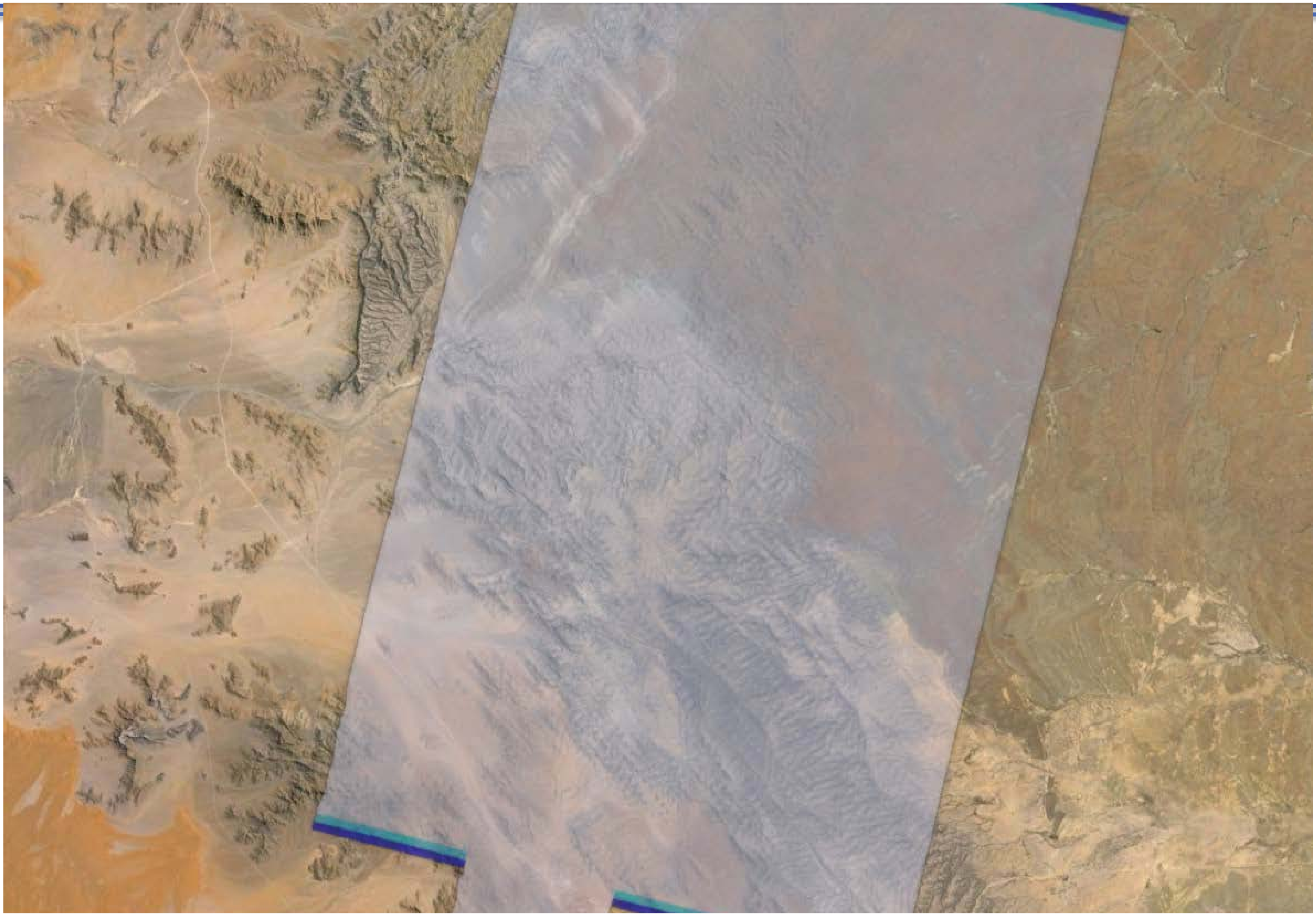
Currently "chip database" created (in a brute-force fashion) by extracting successive 256x256 sub-images of all GLS scenes and storing them according to path and row

Automatic Registration of EO-1 Scenes Using Global Land Survey (GLS) Database

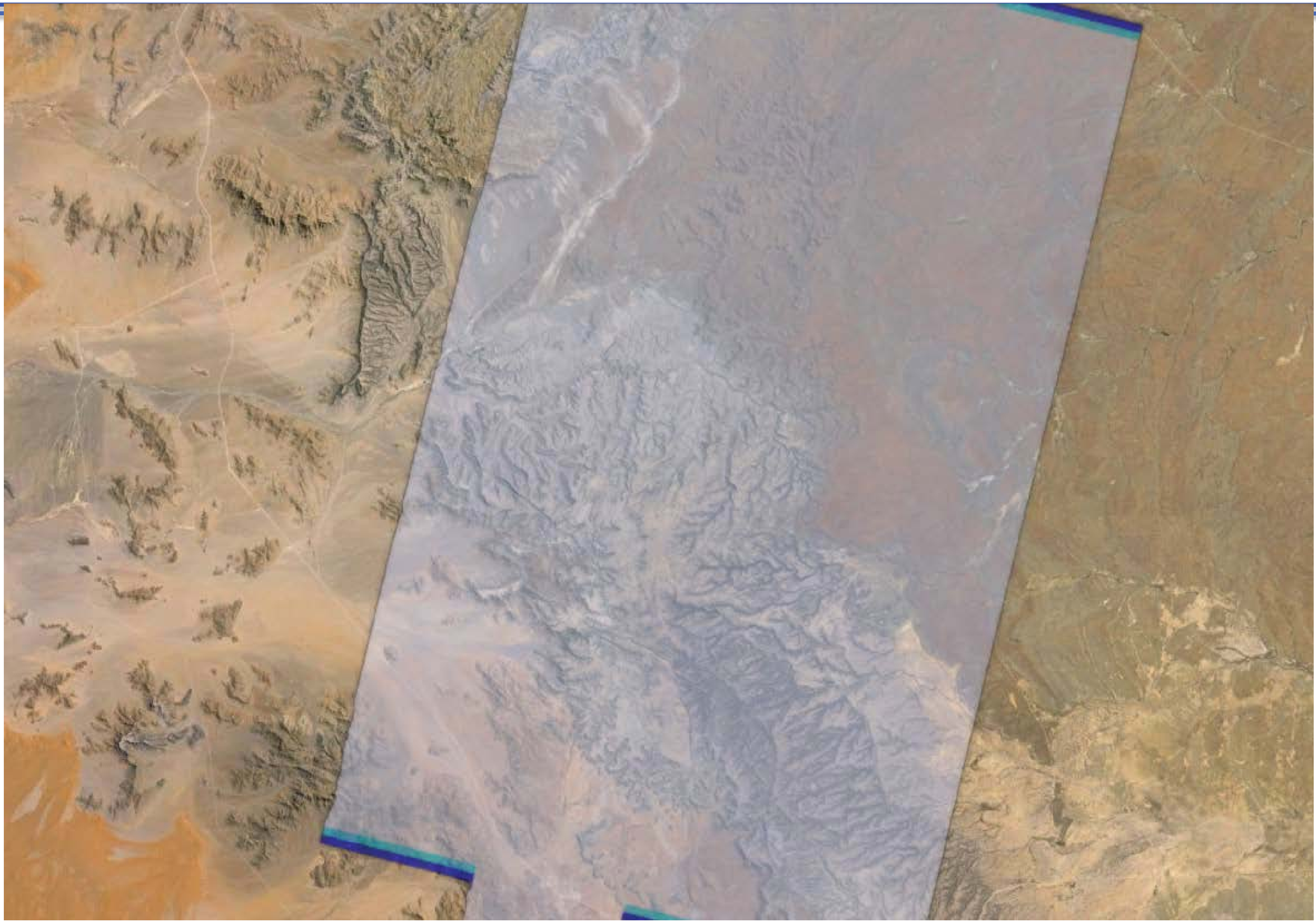


1. *Find Chips that correspond* to the Incoming Scene
2. *For Each Chip, Extract Window* from input scene using UTM coordinates
3. *Eliminate Windows* with insufficient information
4. *Smooth and Normalize* gray values of both Chip and Window using a Median Filter
5. *Register each (Chip,Window) Pair* 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

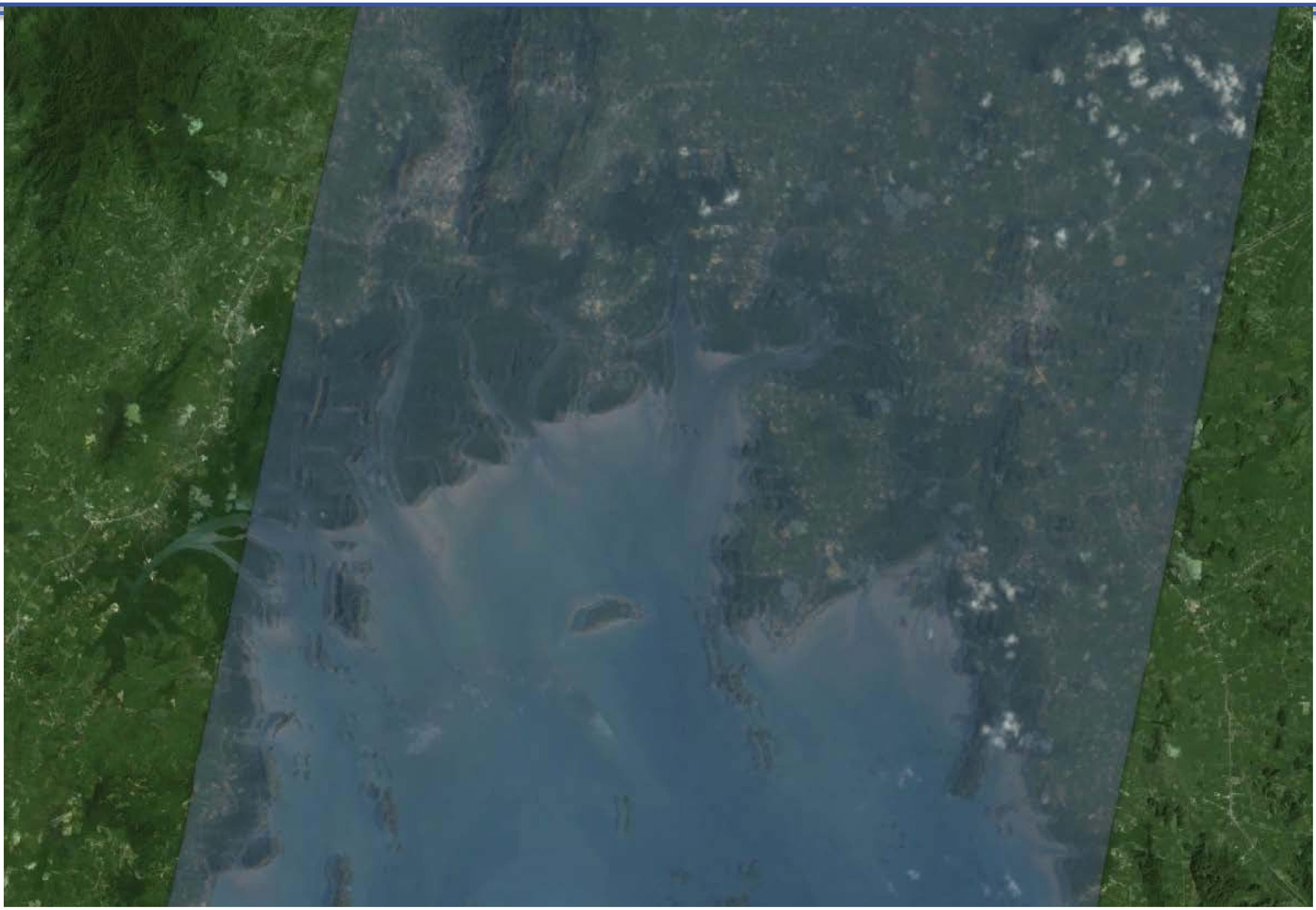
Scene 1 Before Automatic Registration Superimposed onto Google Earth



Scene 1 *After* Automatic Registration Superimposed onto Google Earth



Scene 2 Before Automatic Registration Superimposed onto Google Earth



Scene 2 After Automatic Registration Superimposed onto Google Earth

Quantitative Results With All Chips (“Wall-to-Wall)

- ***Scene 1 (EO1A1780772013325110KF)***
  - Wavelet Registration (Median Global Transformation, after outlier elimination)

    Tx = -15.84, Ty = -18.17, Theta = -0.0083, Scale = 1.0
  - Manual registration (using ENVI):

    Tx = -15.99, Ty = -20.49, Theta = 0.0224, Scale = 1.0
  - **Error in (Tx,Ty,Theta) = (0.15, 2.32, 0.03)**

- ***Scene 2 (EO1A1300542014053110PZ)***
  - Wavelet Registration (Median Global Transformation, after outlier elimination)

    Tx = -14.32, Ty = -3.12, Theta = -0.0211, Scale = 1.0
  - Manual registration (using ENVI):

    Tx = -16.45, Ty = -4.99, Theta = 0.0218, Scale = 1.0
  - **Error in (Tx,Ty,Theta) = (2.13, 1.87, 0.04)**

TIMING – Running Python Script : 19.36s

Chips Selection Using Entropy

- If Chips pre-selected based on the information content (e.g., using an entropy measure)
  - ⇒ Registration may be more accurate because transformation only computed on pairs that have a significant amount of features
  - ⇒ Registration faster because less local registrations
  - ⇒ Chip database smaller to be stored onboard

- Compute Entropy of all Chips Using Histogram:

$$H = -\sum_{i=0}^{255} p_i \log p_i$$

where p_i is the value of the histogram for gray value i

- Keep only Chips with Entropy Above Threshold

- Number of Chips Scene 1/Scene 2:
  - Before Selection:
  - After Entropy Selection:

Quantitative Results
Only Keeping Chips with High Entropy

- ***Scene 1 (EO1A1780772013325110KF)***
  - Wavelet Registration (Median Global Transformation, after outlier elimination)

    Tx = , Ty = , Theta = , Scale = 1.0
  - Manual registration (using ENVI):

    Tx = -15.99, Ty = -20.49, Theta = 0.0224, Scale = 1.0
  - **Error in (Tx,Ty,Theta) = (, ,)**

- ***Scene 2 (EO1A1300542014053110PZ)***
  - Wavelet Registration (Median Global Transformation, after outlier elimination)

    Tx = , Ty = , Theta = , Scale = 1.0
  - Manual registration (using ENVI):

    Tx = -16.45, Ty = -4.99, Theta = 0.0218, Scale = 1.0
  - **Error in (Tx,Ty,Theta) = (, ,)**

TIMING – Running Python Script : s

Conclusions and Future Work

- Results visually acceptable with fast and real-time computations
- Applicable on the ground or on-board
- Computations can be made more accurate and faster by pre-selecting the chips for information content:
  - Initial experiments using entropy => better accuracy and faster computations
  - Potential future improvements:
    - Investigate other chip pre-selection methods, e.g., edgeness count, land cover classification, etc.
    - Use information content method also on extracted windows to only register pairs with sufficient information content
- Other Improvements:
  - Compute global transformation from the list of corners coordinates (GP's)

    => after outlier elimination, compute rigid, affine or polynomial transformation
  - Include cloud and water masks
  - Implement automatic chip registration on SpaceCube or hybrid processor
  - If no database onboard, incorporate automatic "region of interest extraction"

    => change detection can be performed onboard without chip database