NCSA

Automation and Computational Challenges in Image Analyses

Presenter: Peter Bajcsy, PhD

- Research Scientist, NCSA

- Adjunct Assistant Professor ECE & CS at UIUC

- Associate Director Center for Humanities, Social Sciences and Arts (CHASS), Illinois Informatics Institute (I3), UIUC

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

Outline

- Introduction
- Demonstrations of Automation and Computational Challenges in Image Analyses
  - Focus on automation of image analyses – historical map analyses
  - Focus on computational scalability of image analyses – large size image creation and access to remote sensing imagery
  - Focus on automation and computational scalability of image analyses – processing of large volumes of paper scans
- Summary of Open Problems
- Opportunities to Automate and Scale with Mathematica and gridMathematica

Imaging and Image Analyses: Introduction

- Advanced imaging instruments
  - Have become ubiquitous in many scientific areas
  - Need to be connected to significant computational resources to interpret the image content
  - Might be integrated with CPU intensive image content simulations to guide imaging and/or compare results
- Image analyses
  - Have become a part of time critical operations
  - Have increasing requirements for computation and storage due to imaging instrument advancements
  - Have additional computational and storage requirements to support multi-modal/spectral/simulation integration activities, as well as fast access to visualization

Imaging Advancements: Cost Per Pixel

- **Scanners** (~$165 as of 10/10/2010)
  - Canon: CanoScan 8800F Flatbed Scanner; 4800 x 9600 dpi Resolution (46.1 Megapixel), Letter Size Scan Area

- **Cameras** (~$100 as of 10/10/2010)
  - Kodak: EasyShare C195 14.0-Megapixel Digital Camera
  - Samsung: 12.2-Megapixel Digital Camera – Blue

- **Microscopes** (~$100 as of 10/10/2010)
  - Avangard Optics AN-E500 eScope 500x USB 2.0 Mega Pixel Digital Microscope
  - USB Digital Microscope With 1.3 M Pixel Resolution

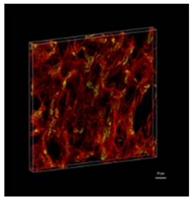
- **Telescopes**
  - COSTAR 8x32 Digital Binoculars Telescope with 2 M Pixel Digital Camera (~$170 as of 10/10/2010)
  - The Panoramic Survey Telescope & Rapid Response System (Pan-STARRS) – 1.4 Giga Pixel camera

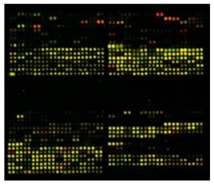
Image Analyses: Grand Challenges

- Our lack of understanding **how to map visual image inspection tasks into a sequence of basic mathematical operations**
  - **Approach:** Formulate sequences of mathematical operations and apply optimization techniques - **Automation** of visual inspections
- Our lack of understanding **how to configure hardware and software in order to perform time-critical image analyses** with the exponentially increasing amount of image data due to advancements in imaging capabilities.
  - **Approach:** Data-driven modeling and support of elastic hardware plus software configurations - **Computational scalability** of image analyses

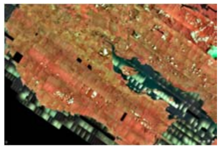
Scientific Applications in Need of Automation and Computational Scalability: Examples

- Bio-medical
  - 3D volume reconstruction from microscopy images and integration with other imaging modalities (MRI, CT, DTI, PET, Ultrasound)
  - cDNA microarray image analyses
  - Cell tracking from video





- Atmospheric science, hydrology and environmental engineering



  - Building and disseminating Giga Pixel images from airborne imaging
  - Predictive modeling from ground and remote sensing
- Humanities, Arts and Social Sciences
  - Imaging and image analyses to preserve and understand historical artifacts

- Astronomy
  - Detection of stars in sky survey images

Demonstrations of Automation and Computational Challenges in Image Analyses

- DISCOVERING KNOWLEDGE OF CARTOGRAPHERS OVER TIME FROM HISTORICAL MAPS
  - Focus on automation of image analyses

- PROVIDING FAST ACCESS TO VERY LARGE SIZE IMAGES
  - Focus on computational scalability of image analyses

- FROM LARGE VOLUMES OF SCANNED LINCOLN PAPERS TO VIRTUAL OBSERVATORIES
  - Focus on automation and computational scalability of image analyses

DISCOVERING KNOWLEDGE OF CARTOGRAPHERS OVER TIME FROM HISTORICAL MAPS

Challenge: How can we automate the discovery process from historical maps to relate the accuracy of earth depiction over time by multiple nations?

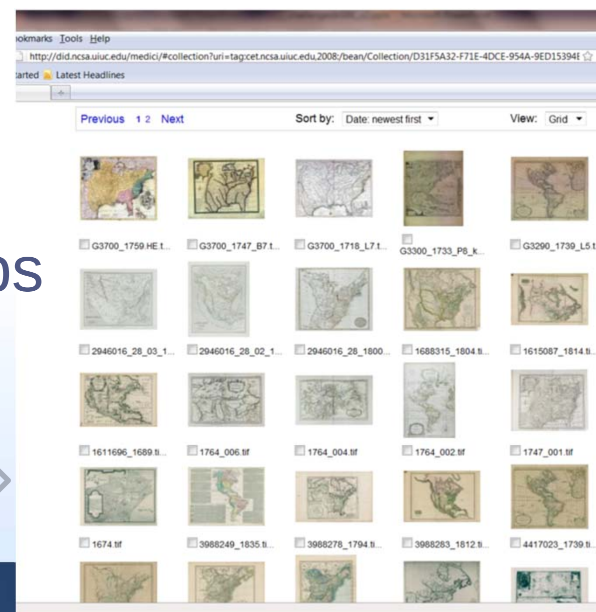
Image Collection

- Data Characteristics
  - UIUC library map collection (44 under investigation)
  - File size ranges a lot ~ average =150MB
  - Test data: 25 maps from UIUC library ~ 4.5GB
  - 150,000 digitized maps in David Rumsey's collection
- Specific image collection of interest
  - French and British maps of the Great Lakes Region from about 1650 to 1850

Target Images:
Contemporary Maps

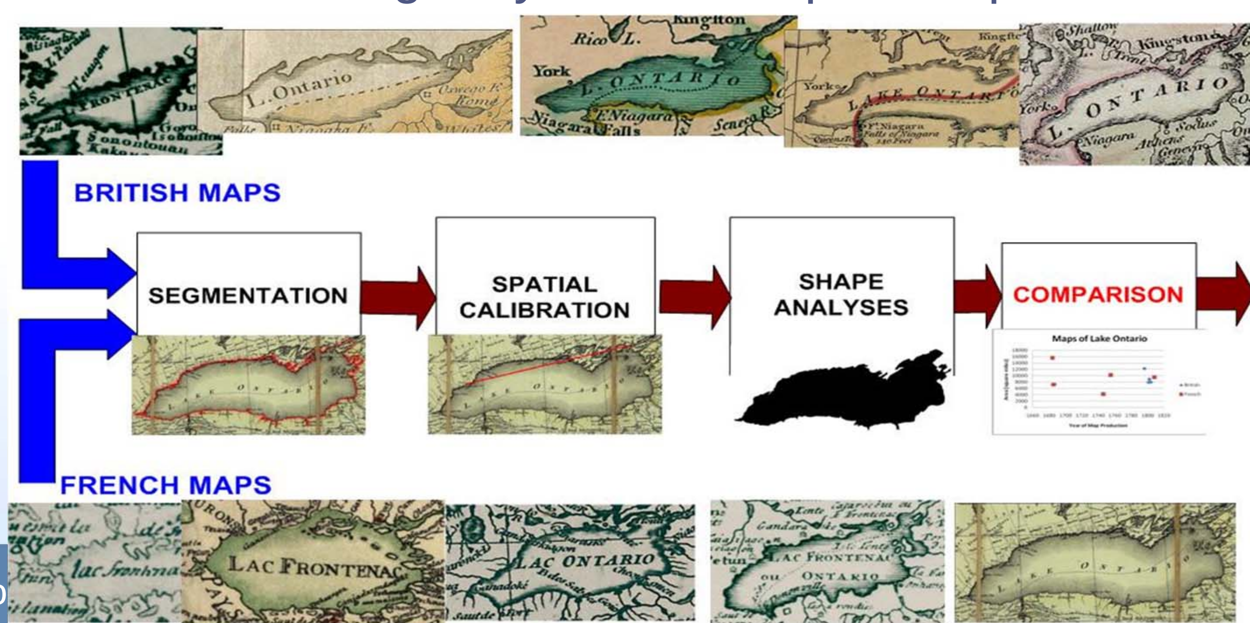
Compare

Historical Maps



Historical Map Analyses

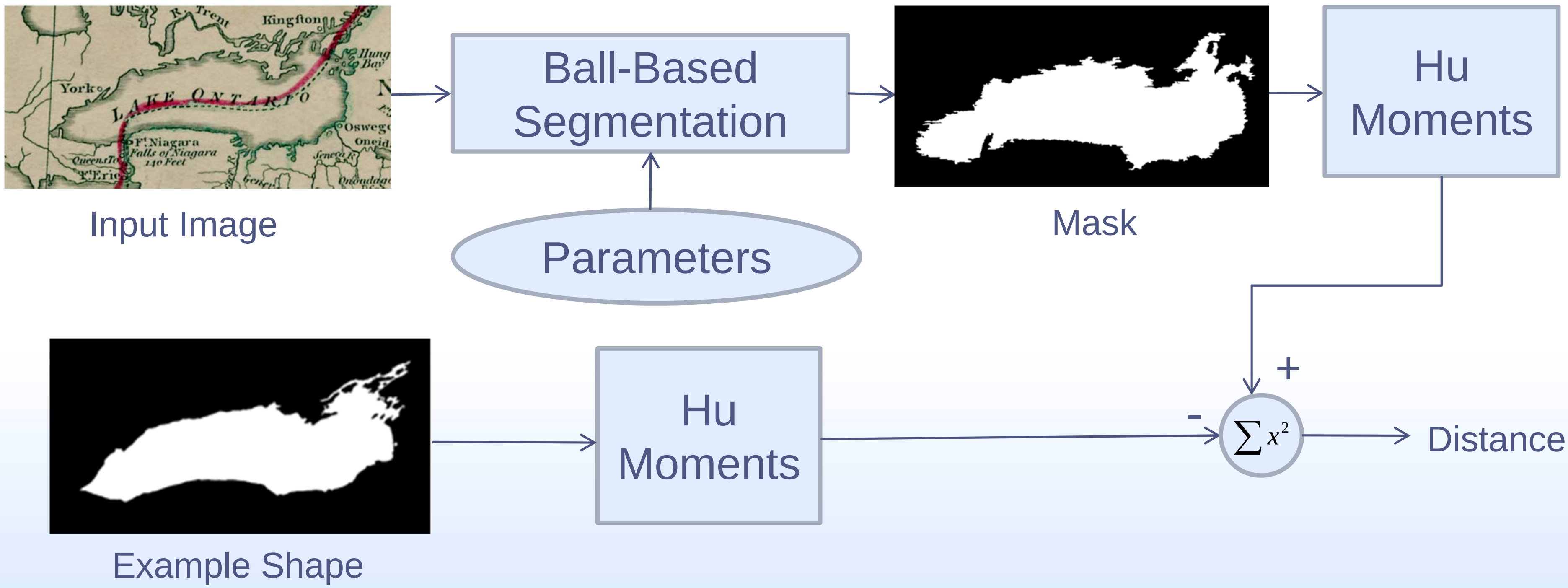
- Analysis workflow
  - Extract the metadata about a map including the map creation year and its place of origin
  - Segment out geographic objects
  - Estimate the map scale based on neatlines
  - Quantify the parameters related to shape, size and color of the segmented objects
  - Deliver chronologically sorted shape/size parameters to end users



NCSA

Segmentation

- Segmentation approach: Template shape-based segmentation using 7 Hu Moments
- Parameters: seed-location, ball size, and threshold
- Goal: Minimize the distance between mask and example shape in space of Hu Moments over a set of parameters



Hu Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Central moments

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

Introduce scale invariance

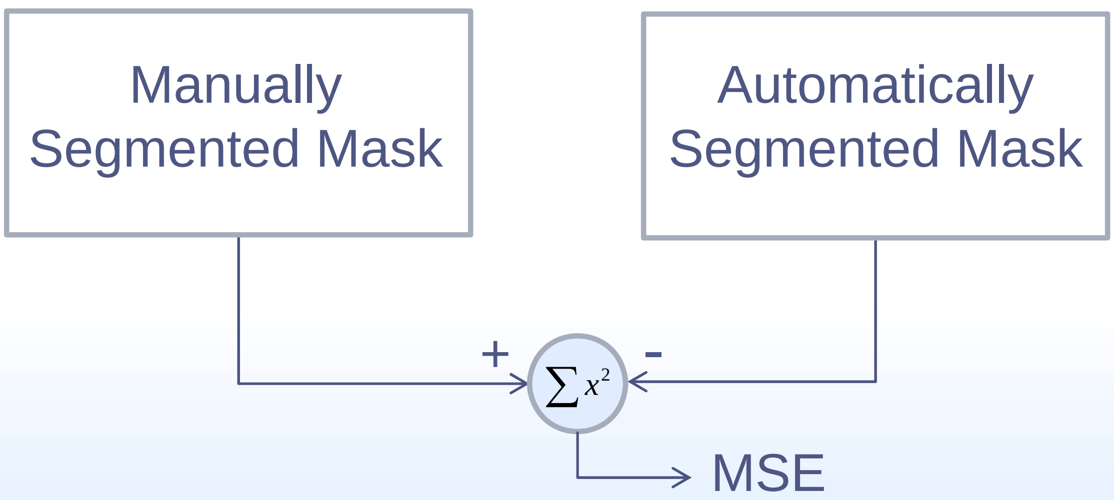
$$I_1 = \eta_{20} + \eta_{02}$$
$$I_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2$$
$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

Hu Moments (M.K. Hu, 1962)

http://en.wikipedia.org/wiki/Image_moments

Template Shape-Based Segmentation

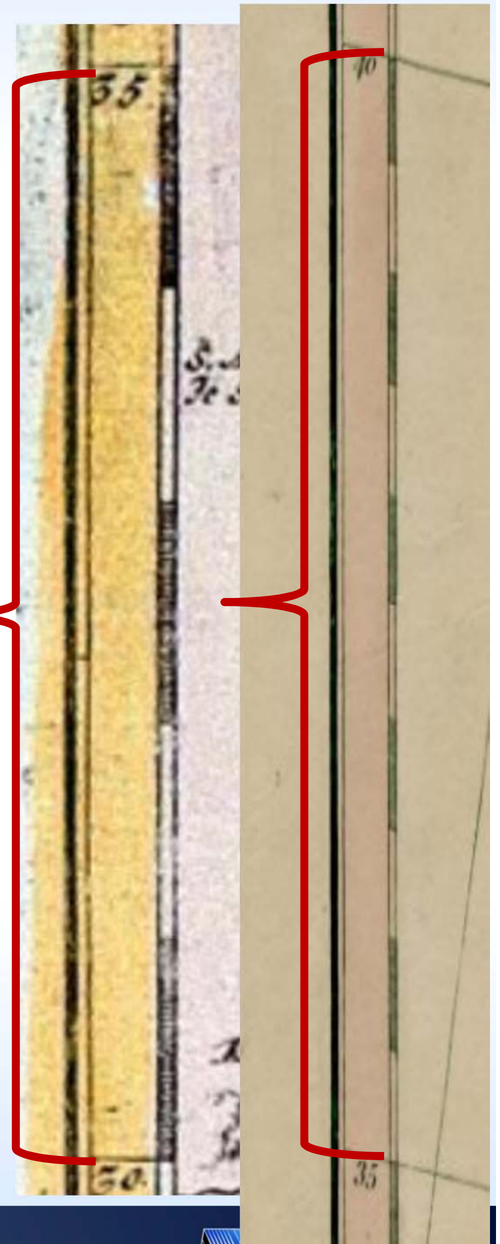
- Motivation: Hu moments are chosen due to invariance under similarity transformations (translation, rotation, scale)
- Accuracy evaluations: Segmentation performance can be evaluated through comparison with manually segmented masks. 28 out of 40 segmentation results were regarded as satisfactory, with <10% MSE.



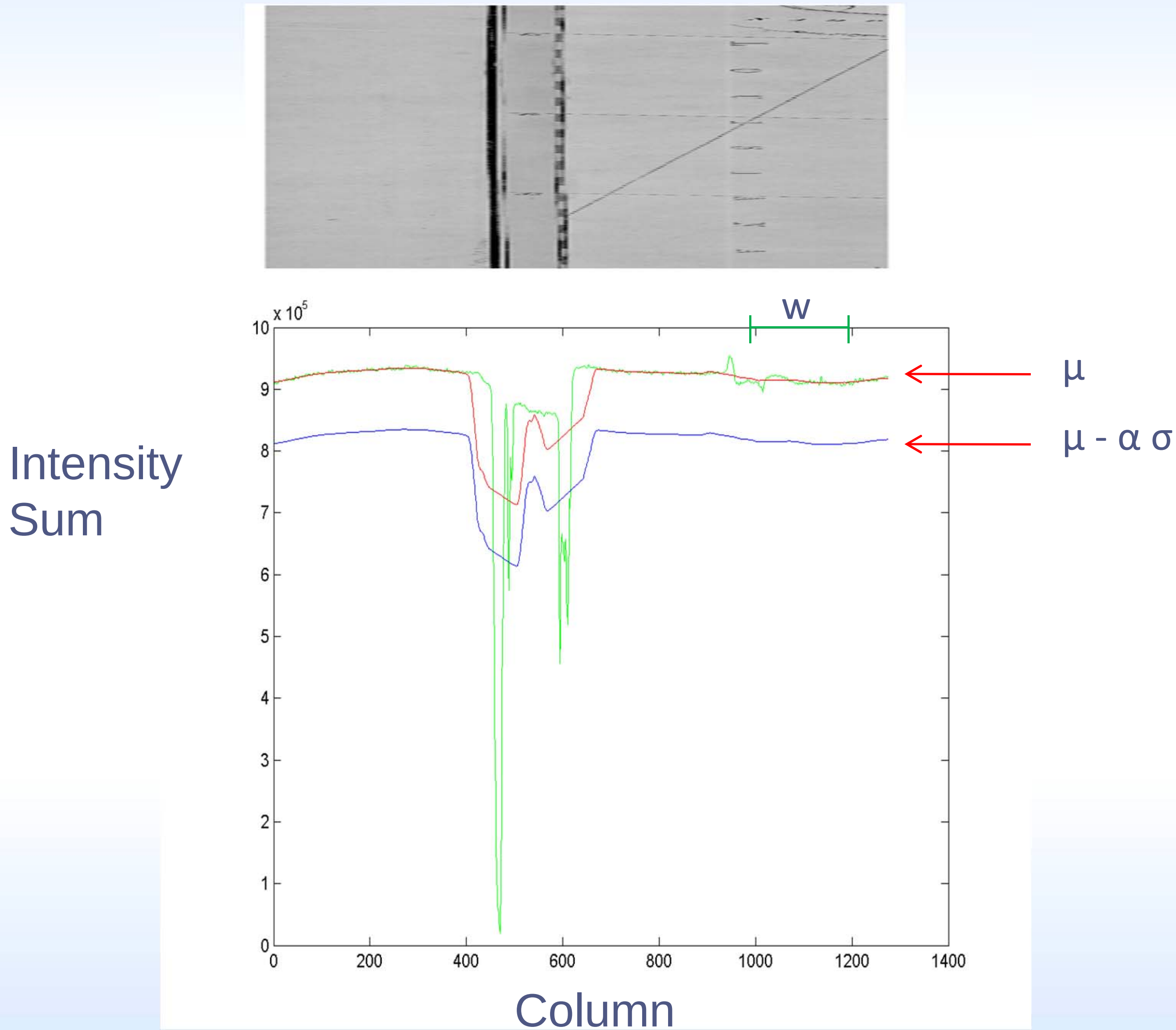
Automatic Spatial Calibration

- **Understanding Content:** There is a many to one mapping between the number of dashes in dashed boundary (neatline) intersected by latitude/longitude lines and the map scale
- **Computational Steps:**
  - Extract and evaluate map borders
  - Identify the dashed line
    - Find all lines
    - Classify: {dashed, solid}
  - Find the dash length in a dashed line
  - Identify the length between two adjacent lat/long intersections with a dashed line
  - Take the ratio and determine the map scale from a lookup table

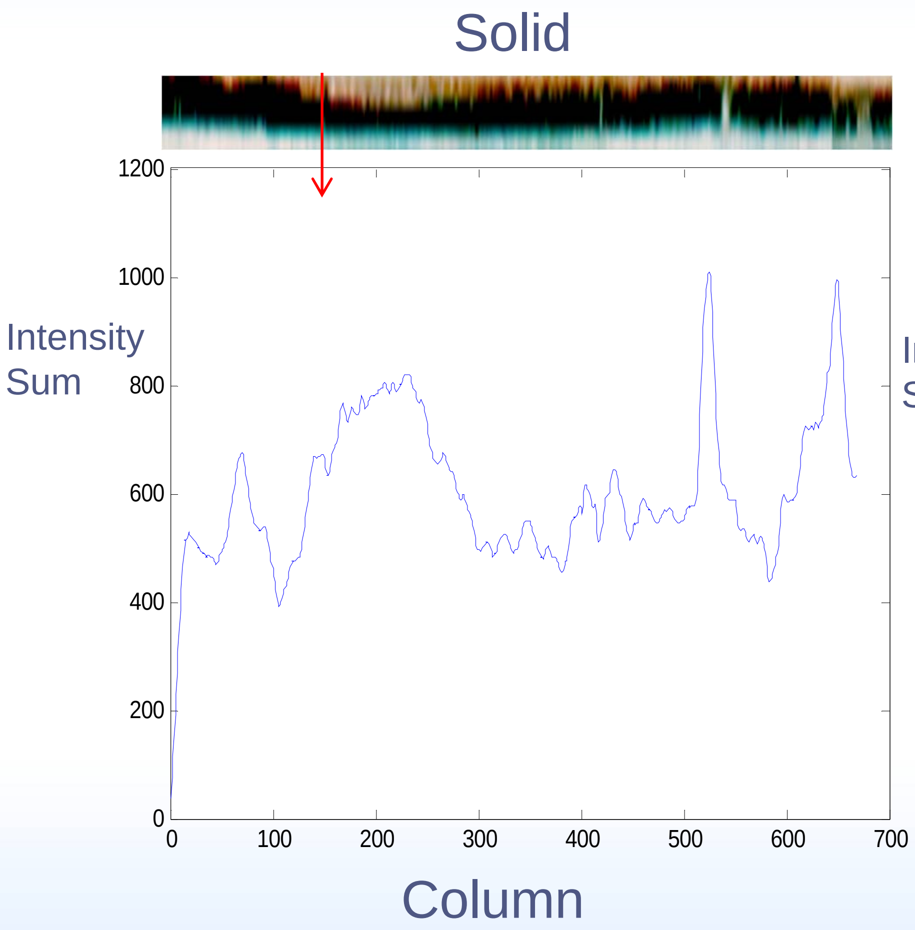
5 or 10 dashes
~ 5 degrees

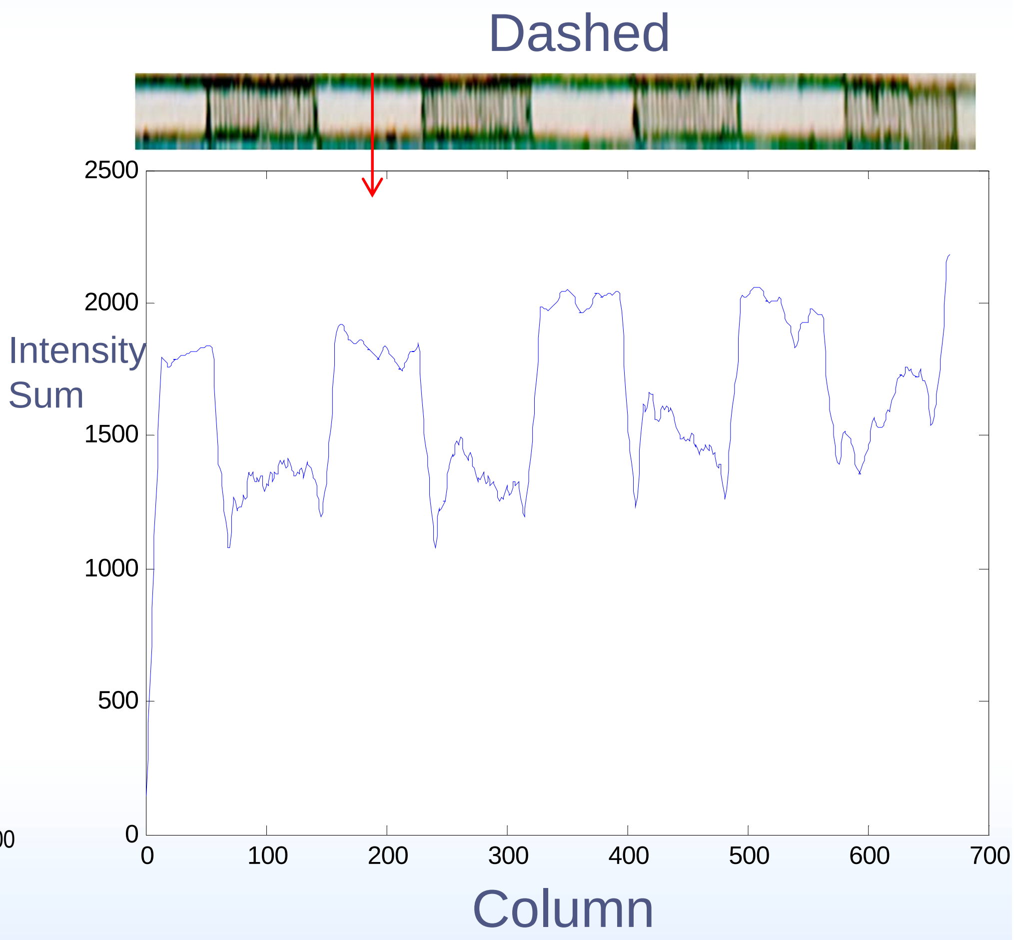


Find All Lines



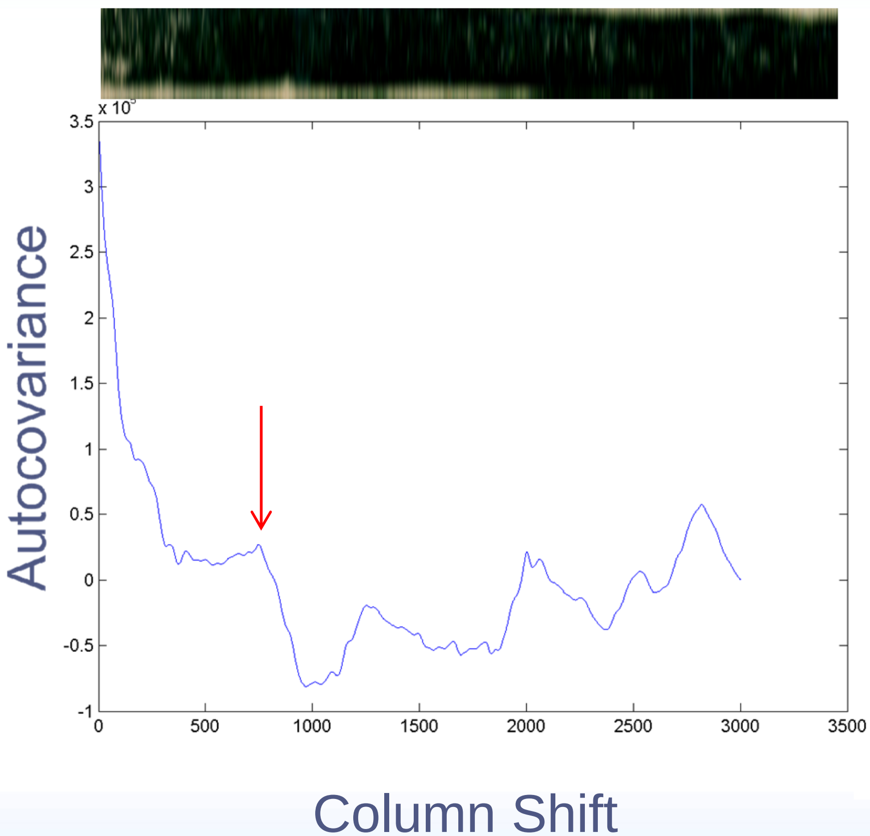
Classification: Solid vs. Dashed Lines



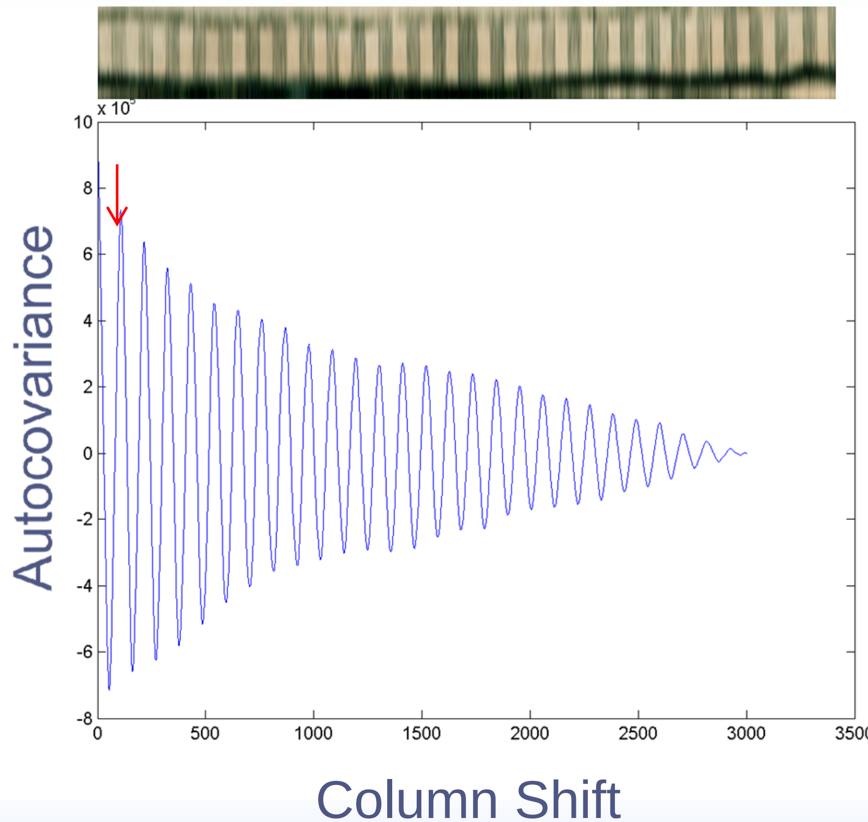


Classification: Autocovariance Features

Solid



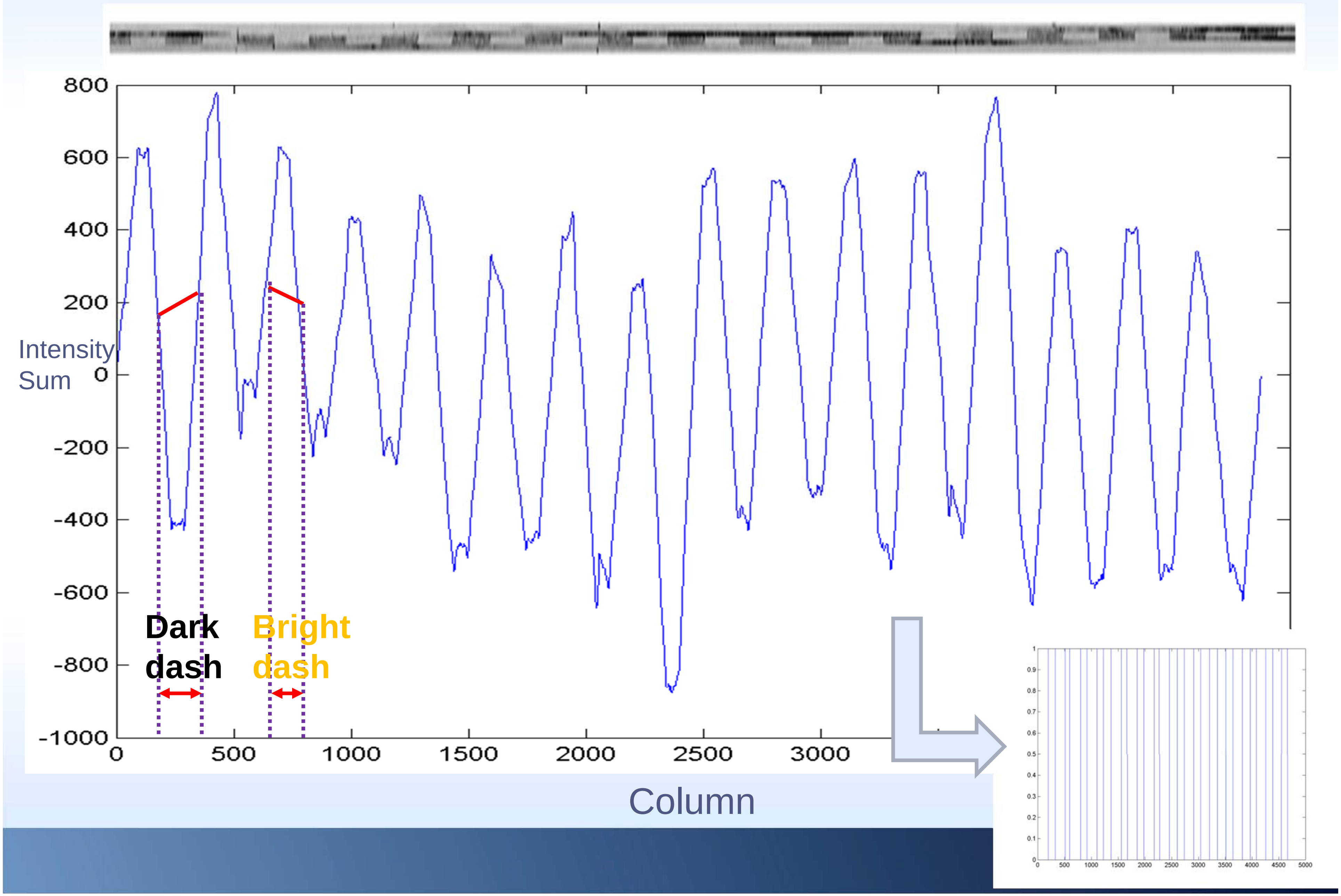
Dashed



Feature: Energy in first peak/total energy

NCSA

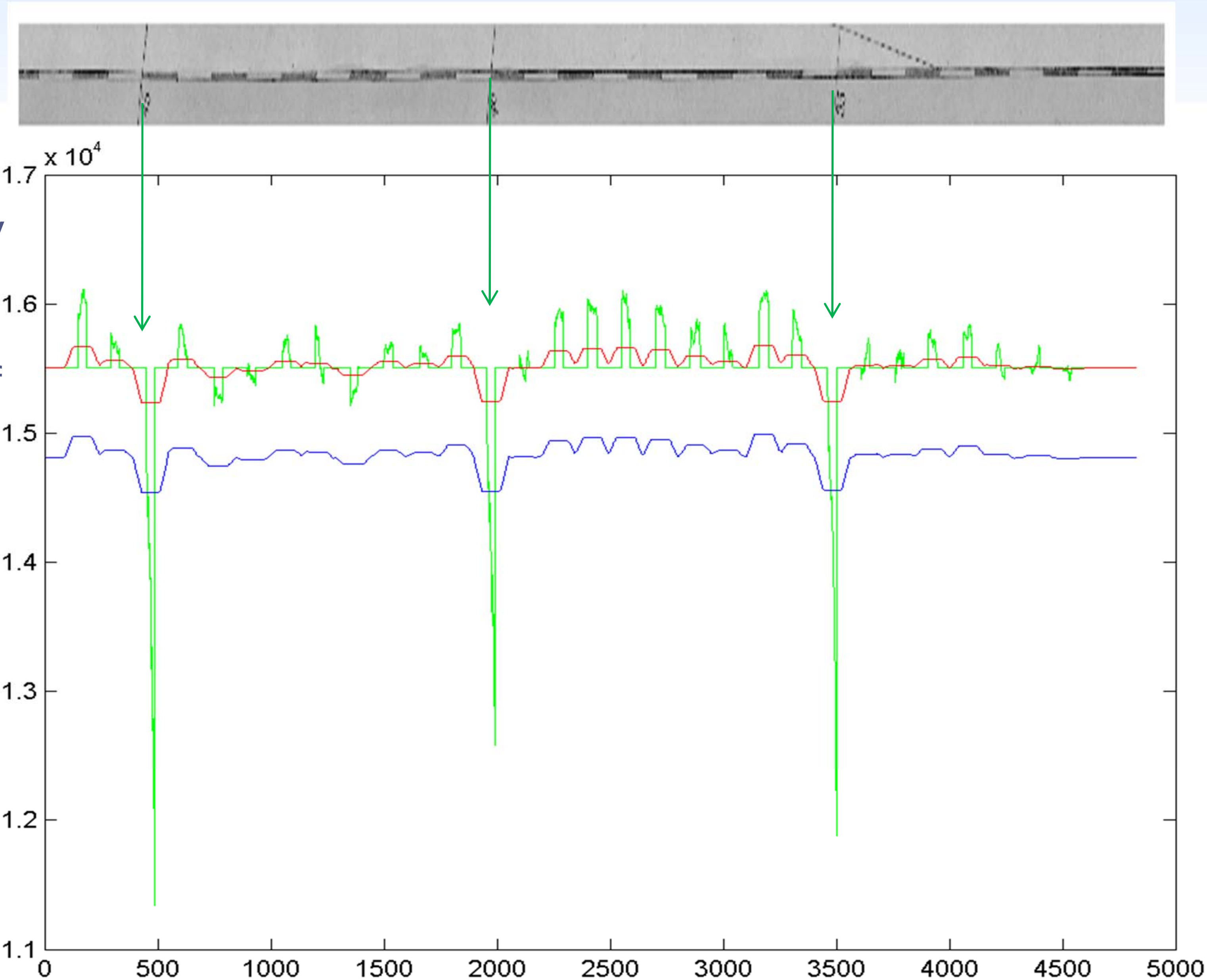
Dash Length Estimation



Detection of Lat/Long Intersections

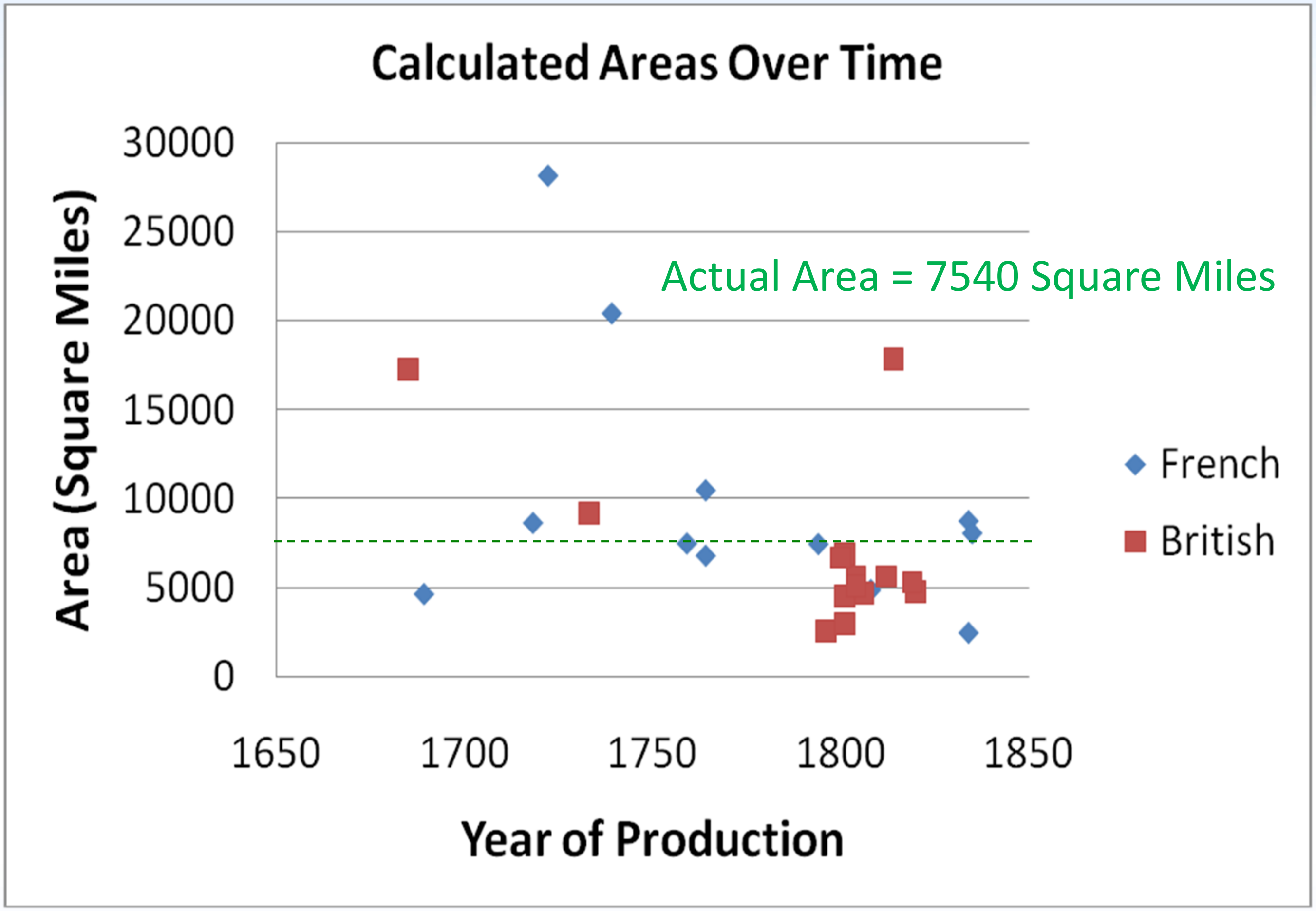
Min Intensity Sum over a range of angles

$$\underset{\theta}{Min}\sum$$



Column

Knowledge From Lake Ontario Results



British:

Mean Year: 1793

Mean Error: 3563 mi^2

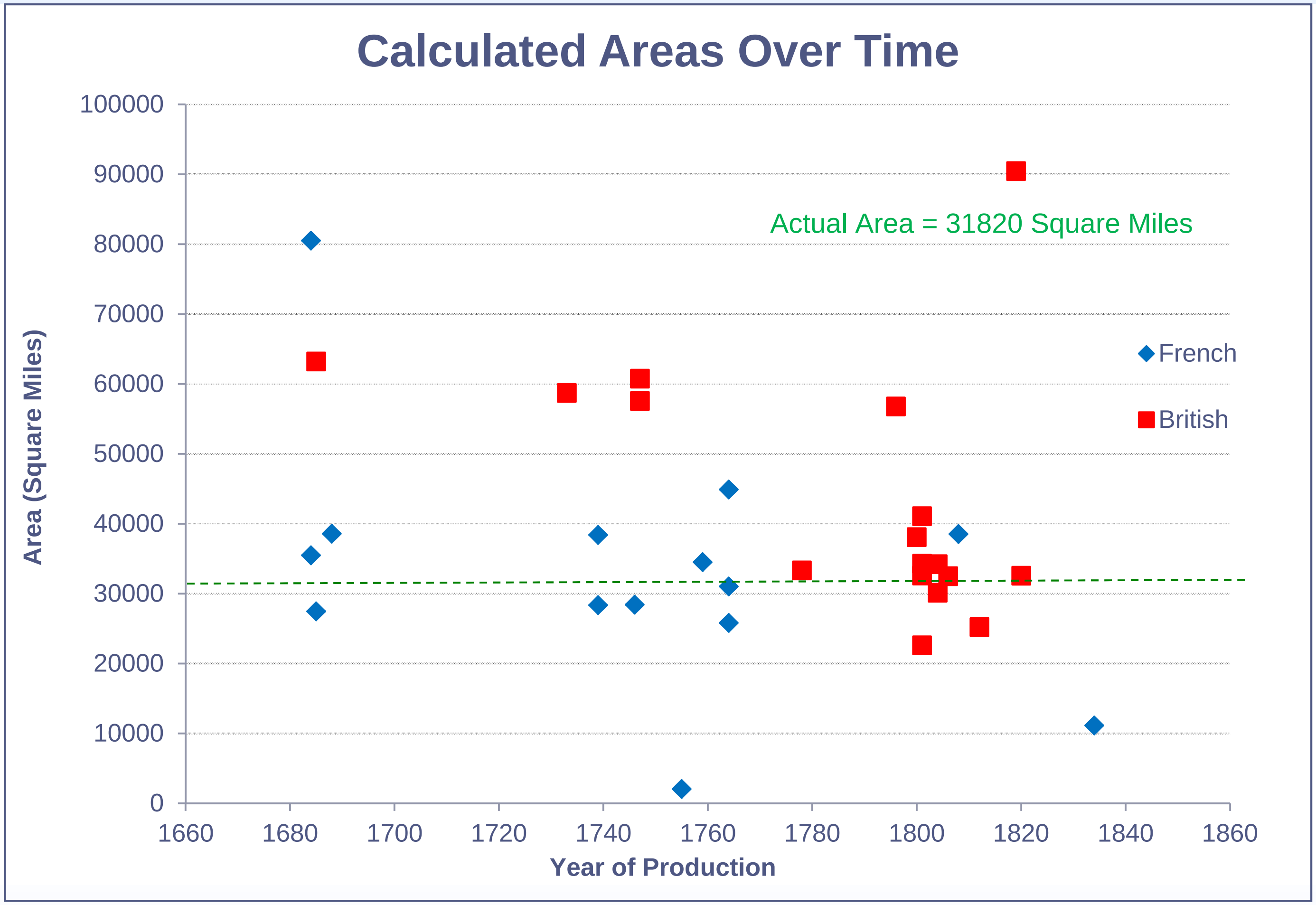
French:

Mean Year: 1772

Mean Error: 4223 mi^2

Qualitative Conclusion: Though the French occupied the Great Lakes region sooner, French maps do not indicate any more accurate depictions of Lake Ontario than British maps.

Knowledge from Lake Superior Results



British:

Mean Year: 1788

Mean Error: 14387 mi^2

French:

Mean Year: 1739

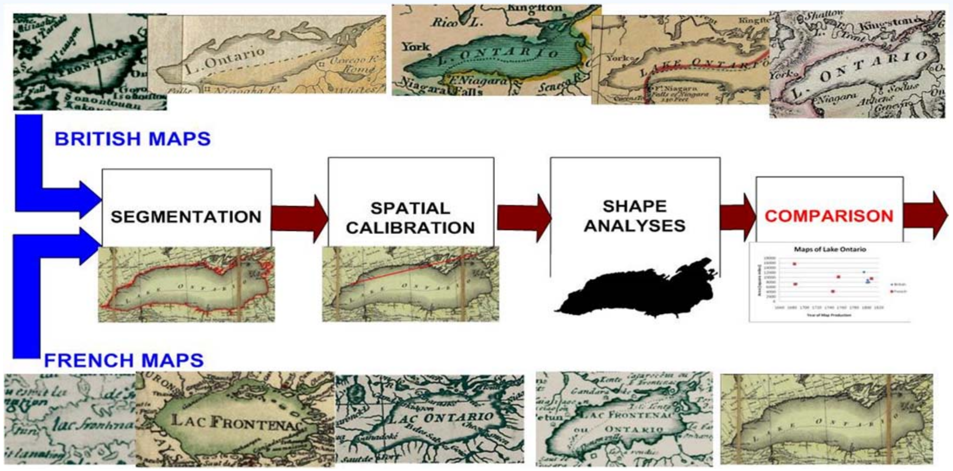
Mean Error: 16276 mi^2

Note: two data points omitted

Qualitative Conclusion: British maps of Lake Superior become more accurate after 1760; French maps do not demonstrate the same change in accuracy. The French maps show accurate knowledge of the lake's size even before it was possible to precisely calculate longitude by ship.

Summary

- Automatically processed:
  - 18 British and 22 French maps of Lake Ontario
  - 18 British and 17 French maps of Lake Superior
- Computation benchmark:
  - map scale estimation step: 2.34 minutes per image on a dual core machine (Matlab code)
  - Rumsey's collection: 150,000 files * 2.34 minutes ~ 5850 hours ~ 244 days
- Mathematical operations
  - Robust first derivatives in 2D
  - Robust periodic signal detection

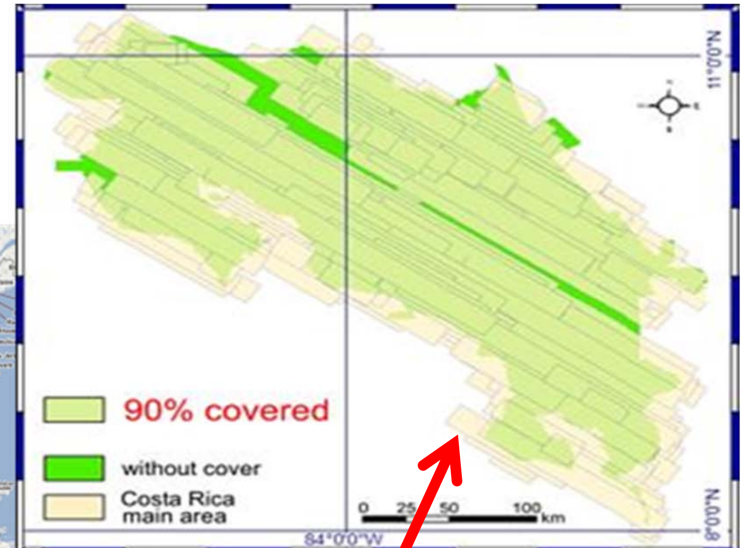


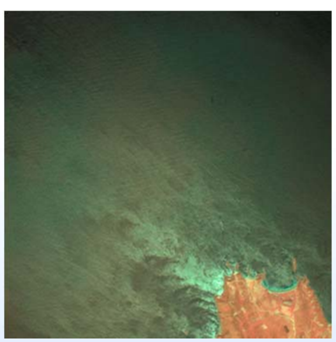
PROVIDING FAST ACCESS TO VERY LARGE SIZE IMAGES

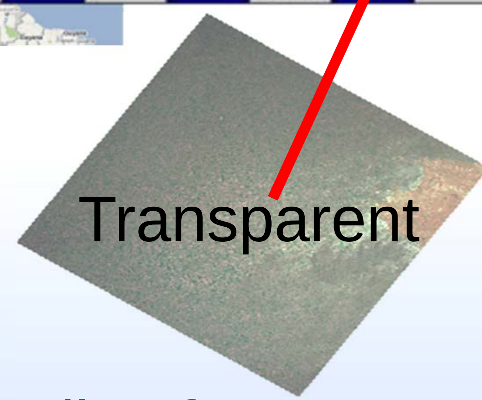
Challenge: How can we automate (a) stitching of a large number of small images and (b) building of an image pyramid to support fast access to very large size images?

Data Characteristics

- Input: 10,158 airborne images (16.6M Pixel each) ~ 0.5TB







An original tile

The same tile after pre-processing

Data and Computation Description

- Data volume:
  - Costa Rica airborne imagery: 10,158 tiles ~ 0.5TB plus additional 0.5TB of hyperspectral imagery
- Stitching
  - Input: georeferencing information about each tile center plus corner points
  - Output: The stitched image is 294,847x269,195 (WxH) pixels ~ 79.4 Giga Pixel color image
- Image pyramid
  - Input: 79.4 Giga Pixel image
  - Output: 19 levels ($\log_2(\max\{W,H\}) = 18.17$);
    - ~1,616,015 tiles of up to 258x258 pixels (1 pixel overlap on each side)

Memory and Storage Challenges

- Stitching: 79.4 Giga Pixel color image ~ 238.2 GigaBytes (raw data)
- Image Pyramid: pixel size (1+1/4+1/16+….1/(2^19)) * 79.4 Giga Pixels ~ (4/3) * 79.4 Giga Pixels = 105.87 Giga Pixels ~ 317.6 GB (raw data)
- Memory: 4 bytes per pixel (integer ~ RGB+transparency)
- Storage: PNG file format (3 bytes per pixel + header and then compressed)

Storage: Image Pyramid Size

$$\text{Pyramid Size/Pixel Size} = 1 + \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^8} + \cdots \frac{1}{2^{MaxLevel}} =$$

$$= \frac{2^{MaxLevel} + 2^{MaxLevel-2} + 2^{MaxLevel-4} \ldots + 4 + 1}{2^{MaxLevel}} =$$

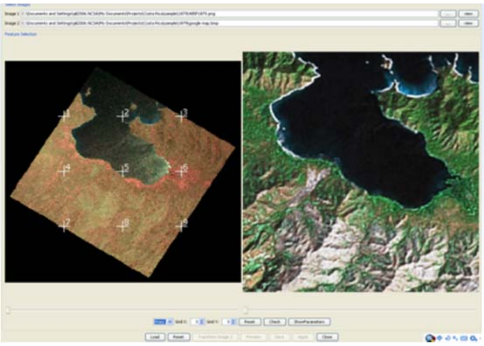
$$= \frac{\frac{1}{3}(2^{MaxLevel+2}-1)}{2^{MaxLevel}} = \frac{1}{3}(4 - \frac{1}{2^{MaxLevel}}) < \frac{4}{3}$$

$$x = 2^{MaxLevel} + 2^{MaxLevel-2} + 2^{MaxLevel-4} \ldots + 4 + 1$$

$$x + 2*x = 2^{MaxLevel+2} - 1$$

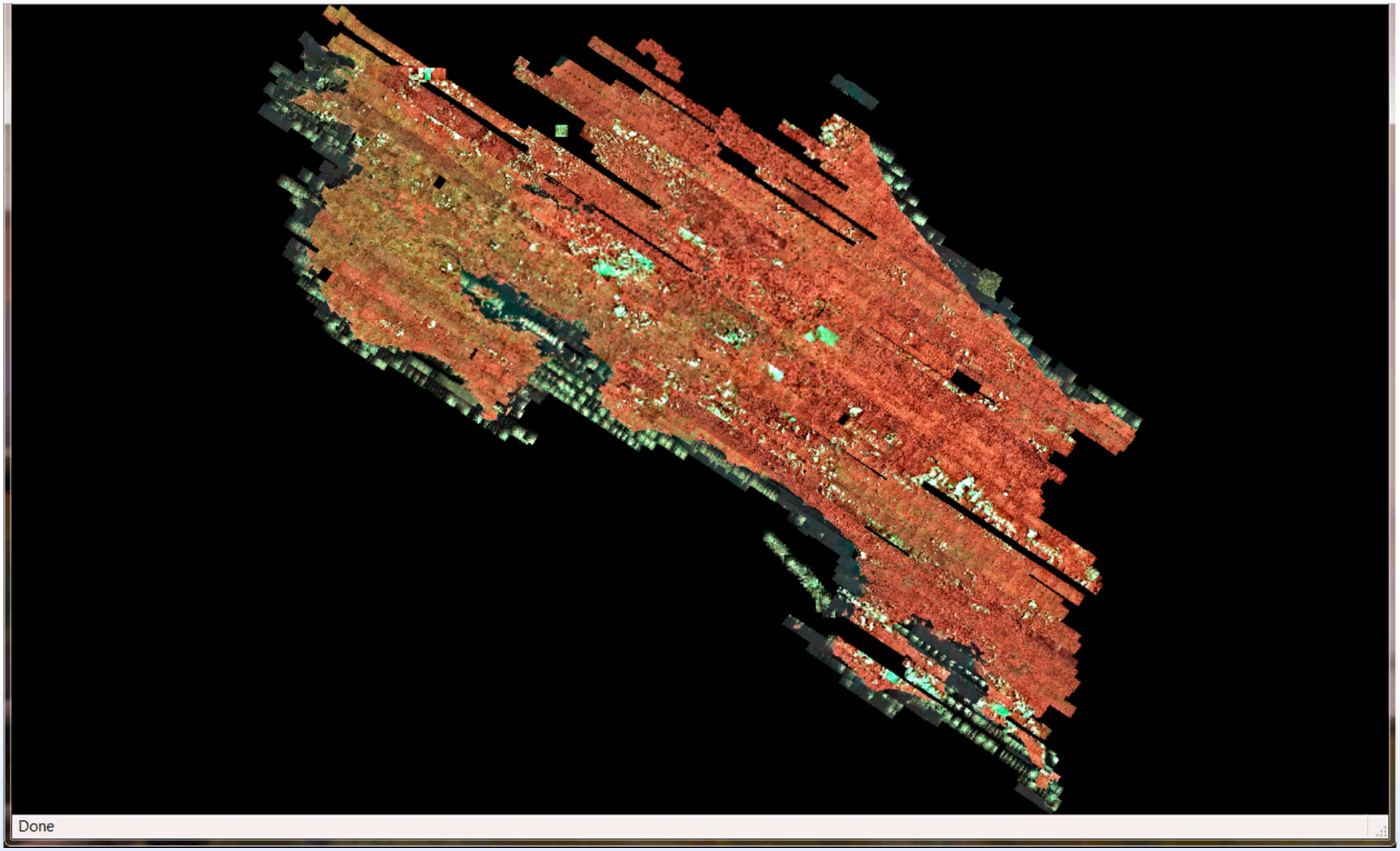
$$x = \frac{1}{3}(2^{MaxLevel+2} - 1)$$

Computational Process



- Information available for stitching
  - Georeferencing information about the tile center plus corner points (Excel spreadsheet with latitude/longitude of each image)
  - Text document with flight paths and rotation of plane (image)
- Computational steps to build an image pyramid from image tiles:
  - Convert image location to geospatial location
  - Find all images that are inside geospatial area of a pyramid tile
  - Scale/Rotate images to match flight path and scale to the pyramid tile level
  - Combine all images to create a pyramid tile

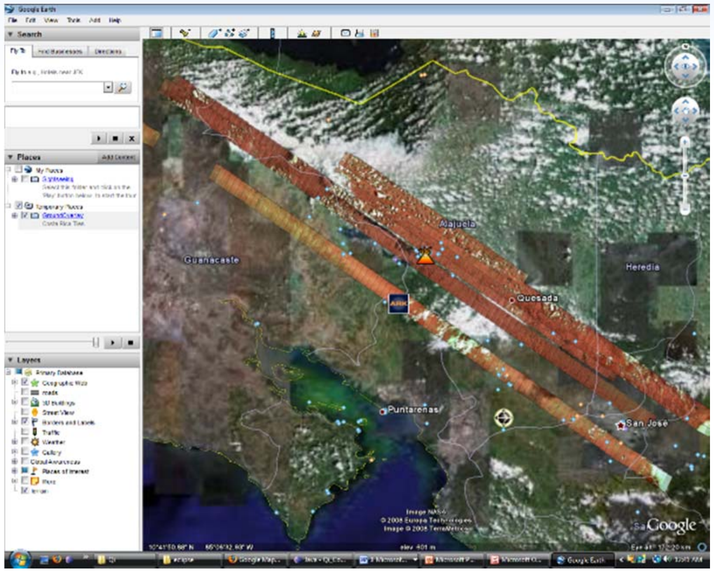
Demo



Overlay with Google Map/Earth

Accessing imagery using Google Map

We generated programmatically a HTML file to overlay tiles onto Google Map via an internet



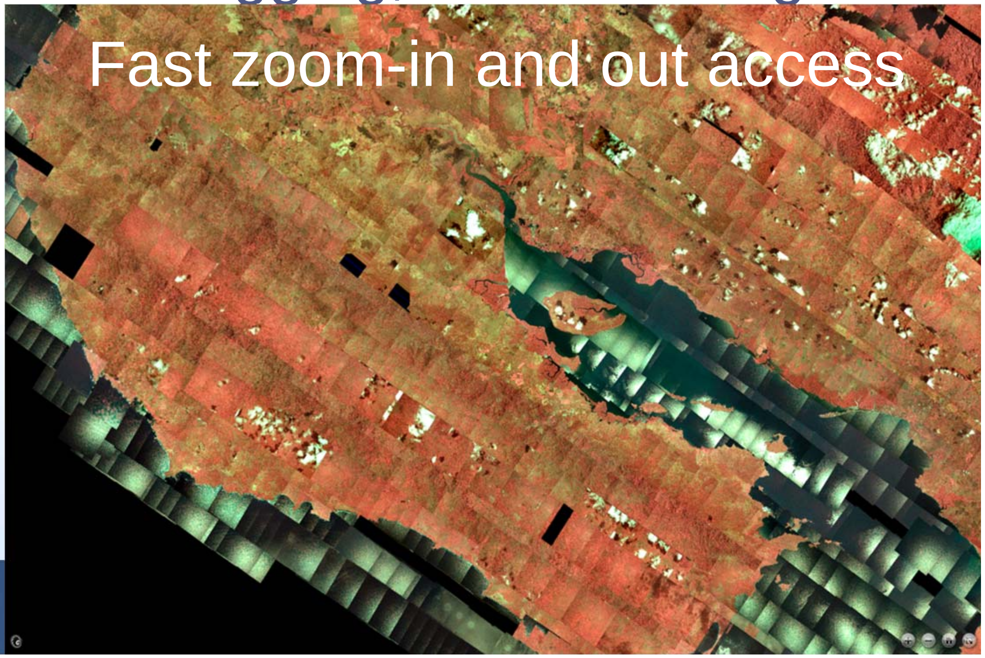
Accessing imagery using Google Earth

We generated a KML file to overlay tiles onto Google Earth.

Observation: It took two times more time to retrieve and render image tiles in Google Map than in Google Earth. The limitation of Google Earth is that it loads all images to RAM first and image volumes cannot exceed the maximum allowed cache memory allocations.

Summary: Computational Challenges

- Building and Disseminating 79.4 Giga Pixel Color Image from image tiles:
  - Building an image pyramid per input image tile ~ **12 hours on 120 cores** (MS ABE ~ ABE= NCSA Intel 64 Cluster with 9600 cores, 2.33GHz processor, 8GB RAM)
  - Building the final pyramid consisting of **19 levels and 1,616,015 pyramid tiles ~ 30 days on 120 cores MSABE**
- Overall time including software debugging, data cleaning and hardware failures?



FROM LARGE VOLUMES OF SCANNED LINCOLN PAPERS TO VIRTUAL OBSERVATORIES

Challenge: How can we automate image preprocessing (e.g., removal of color scale bars) and transcriptions of image scans of handwritten Lincoln Papers to deliver viewable and searchable digital information?

Input: Paper Copies of Docs & Metadata

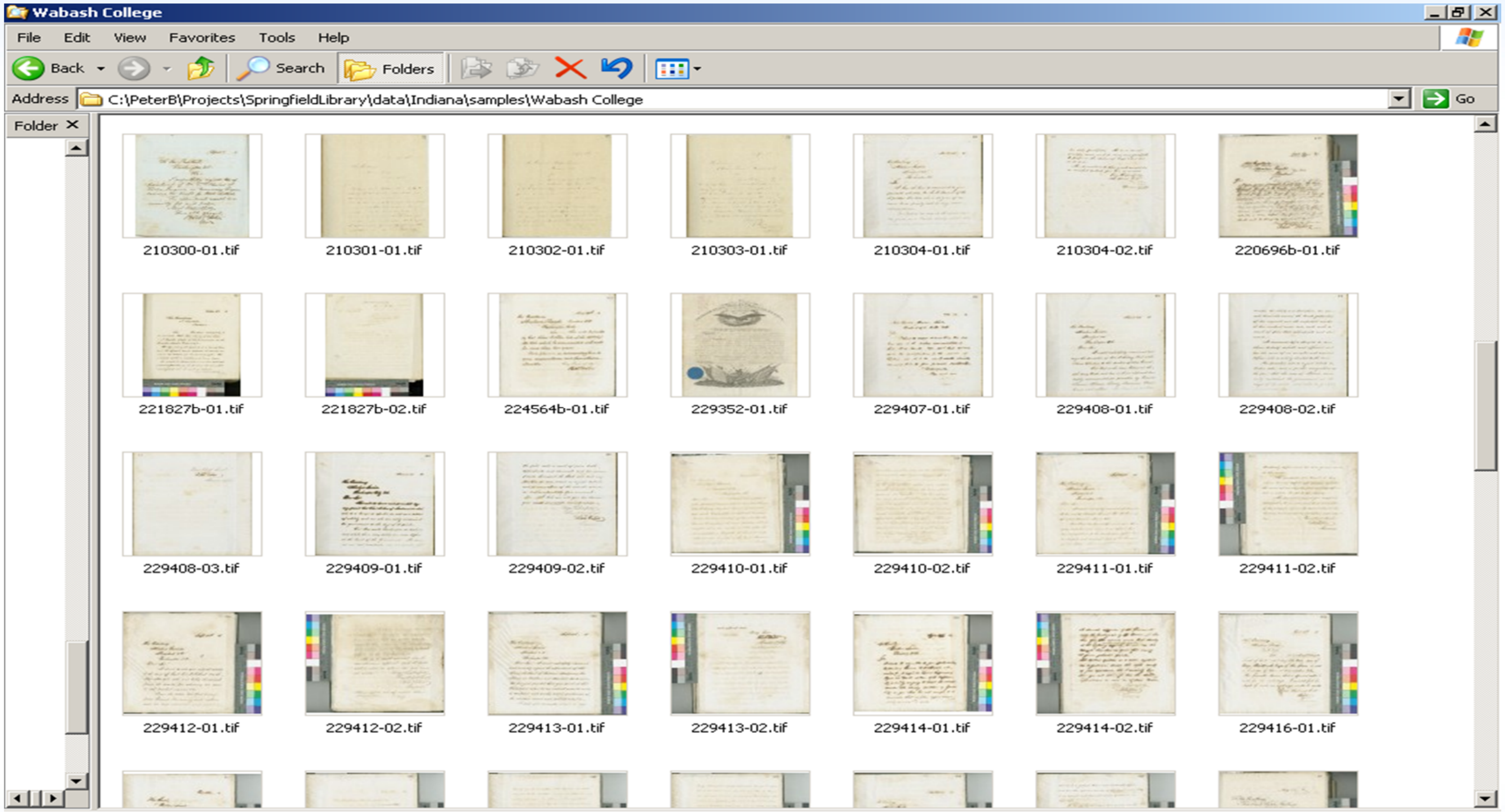
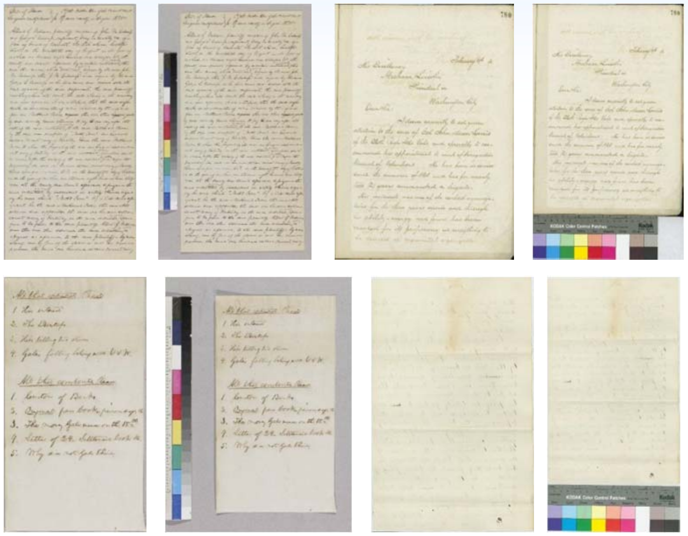


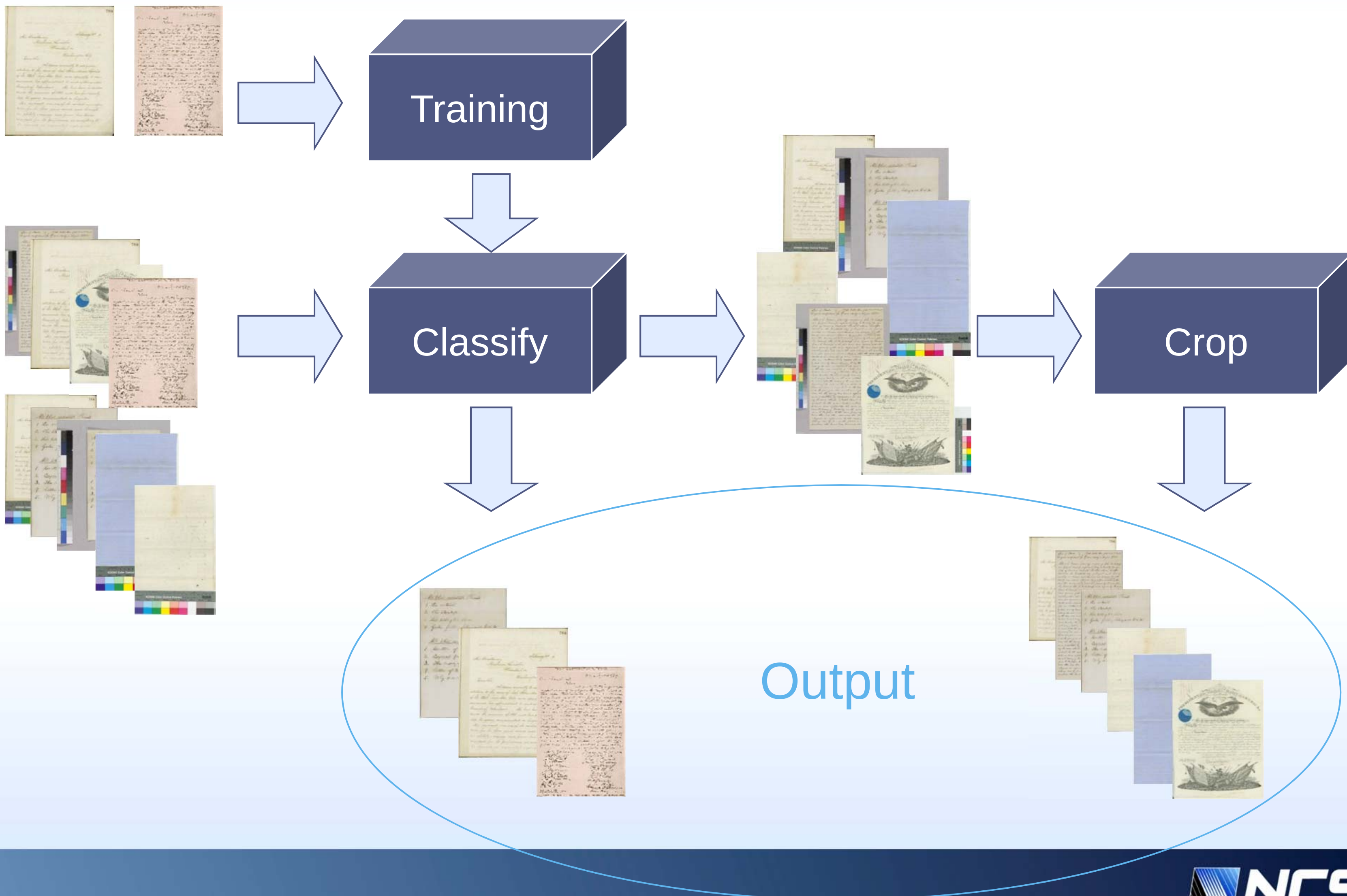
Image Scan Characteristics



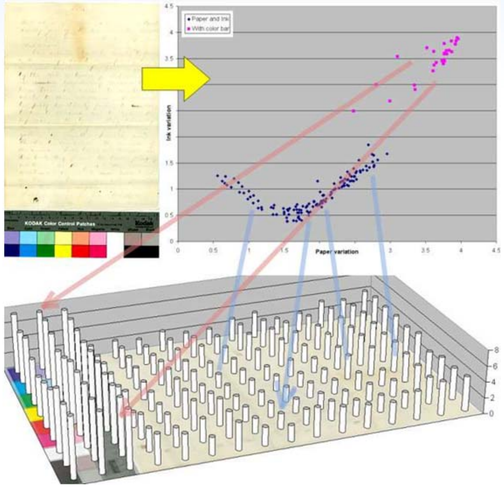
- Lincoln Papers: Characteristics
  - Background paper color and intensity
  - Ink color and intensity
  - Density of writing
- Scanning Process
  - Color scale bar position
- Image scan file characteristics
  - The average image size is about 150 MB.
  - Currently, there are about 39,000 scanned images (5.9 TB),
  - The anticipated number is 200,000 - 300,000 images (34 TB).

Task: Automatically classify images and remove the Kodak color scale bar if needed

Image Cropping



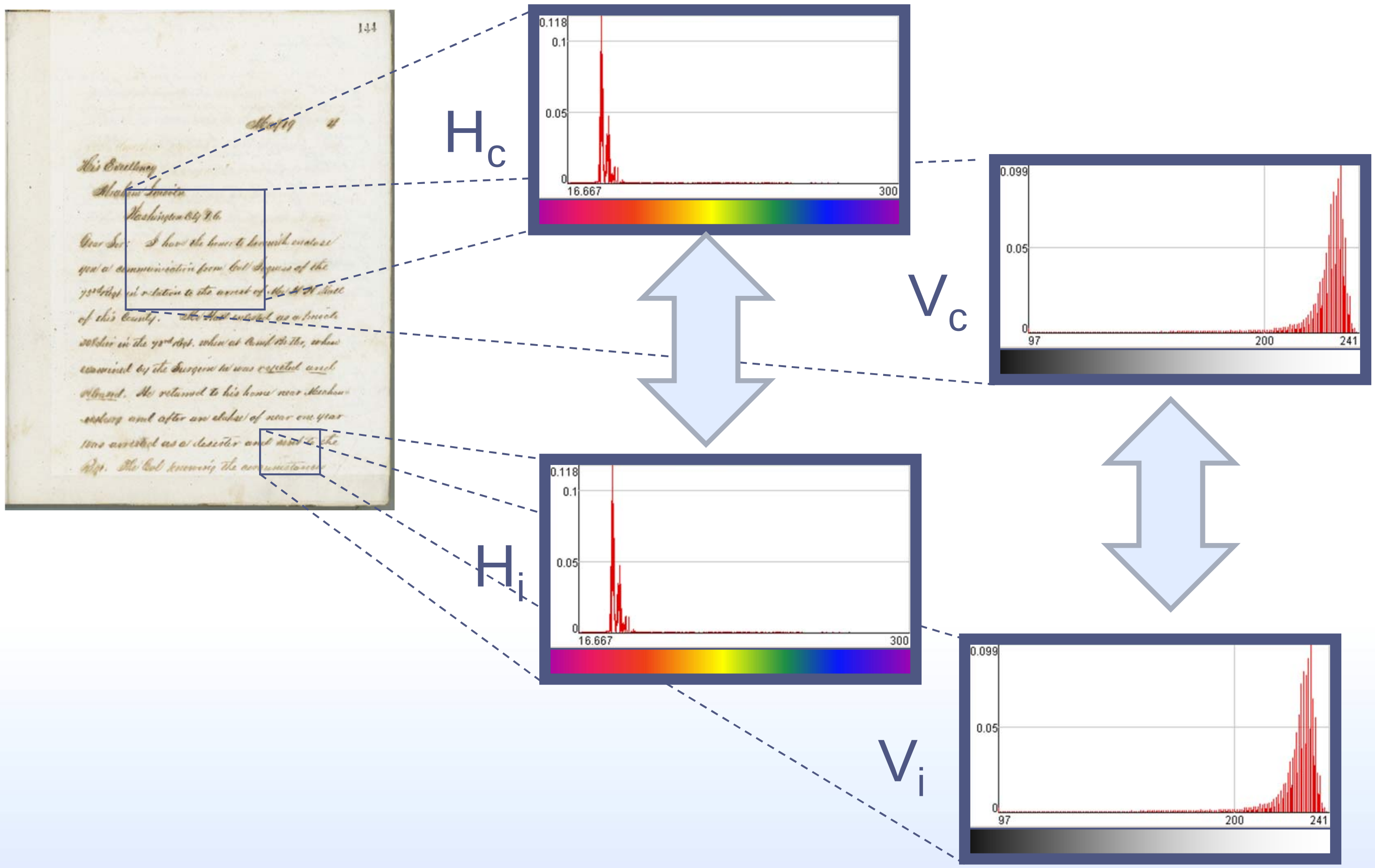
Automation of Image Cropping



- Pre-processing:
  - convert image from RGB to HSB color space
  - create a grid of tiles from each image
- Training:
  - select training tile (auto or semi-auto methods)
  - compute histogram of hue for the training tile
  - select similarity threshold for comparing histograms (auto or semi-auto methods)
- Testing:
  - compute histogram of every tile
  - compare histogram of every tile with the histogram of a training tile and assign “Kodak color bar” or “doc” label
- Post-processsing:
  - Determine crop area based on tile labels and crop image

Histogram of Hue-Saturation-Value (HSV)

Spatial characteristics of HSV Histograms

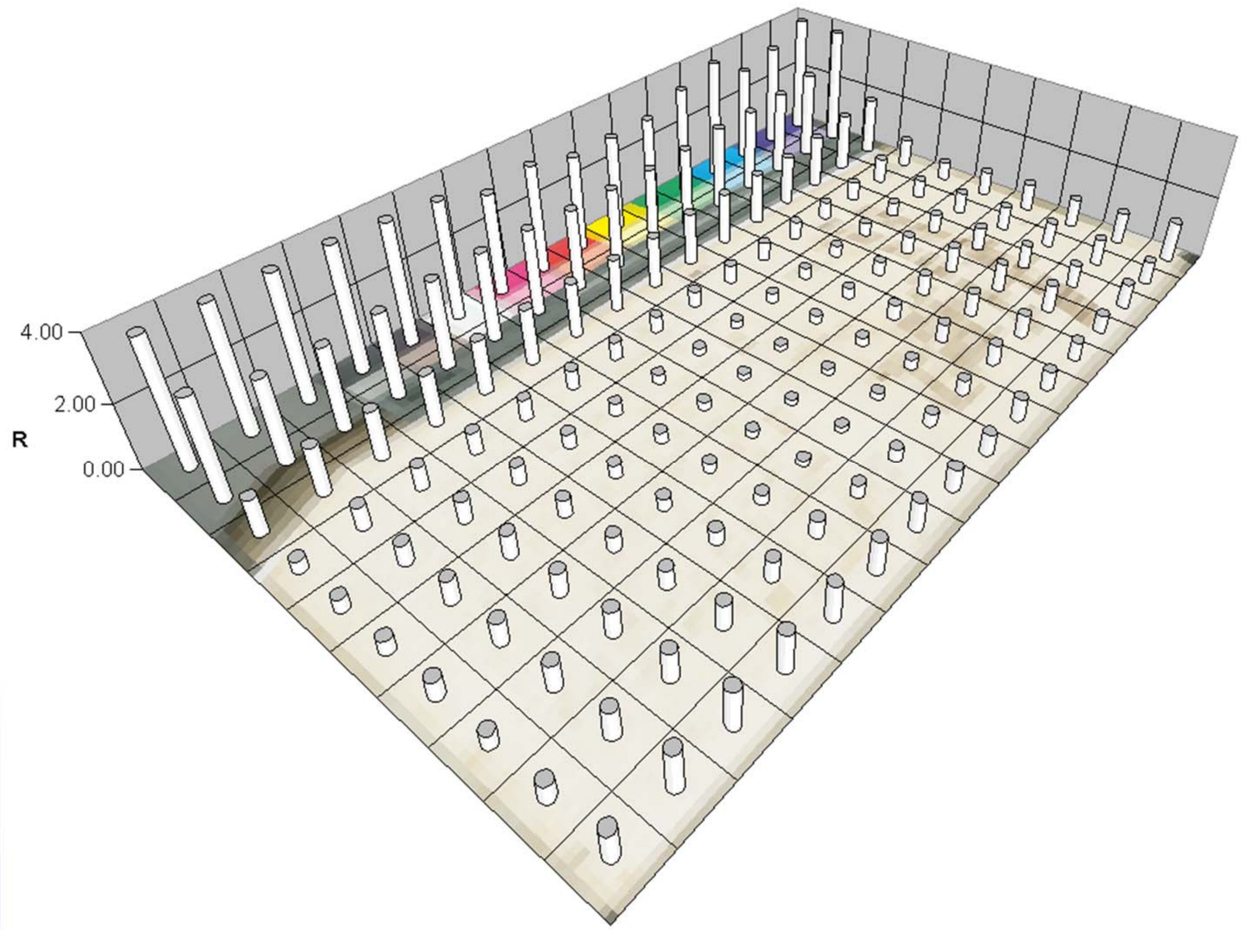


Comparison of HSV Histograms

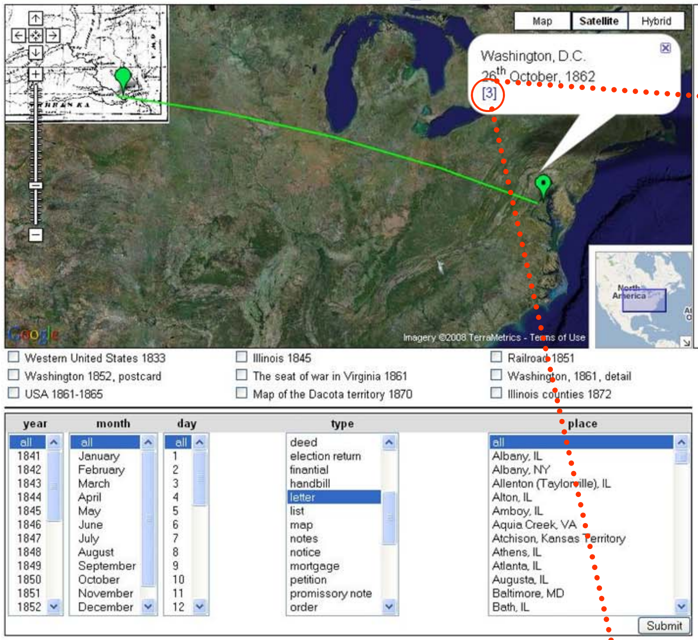
- The difference between two normalized histograms h_a and h_b is defined as:
  - $d: H \times H \rightarrow [0,1]$
  - $d(h_a, h_b) = \Sigma_k |h_a(k) - h_b(k)|$
- The difference between the center of and image and a piece of the image is denoted Residual and can be defined as:
  - $R = d(H_i, H_c) + d(V_i, V_c)$
- <u>Assumption:</u> The central area of every image contains only paper and ink

Spatial Distribution of HSV Histogram Residual

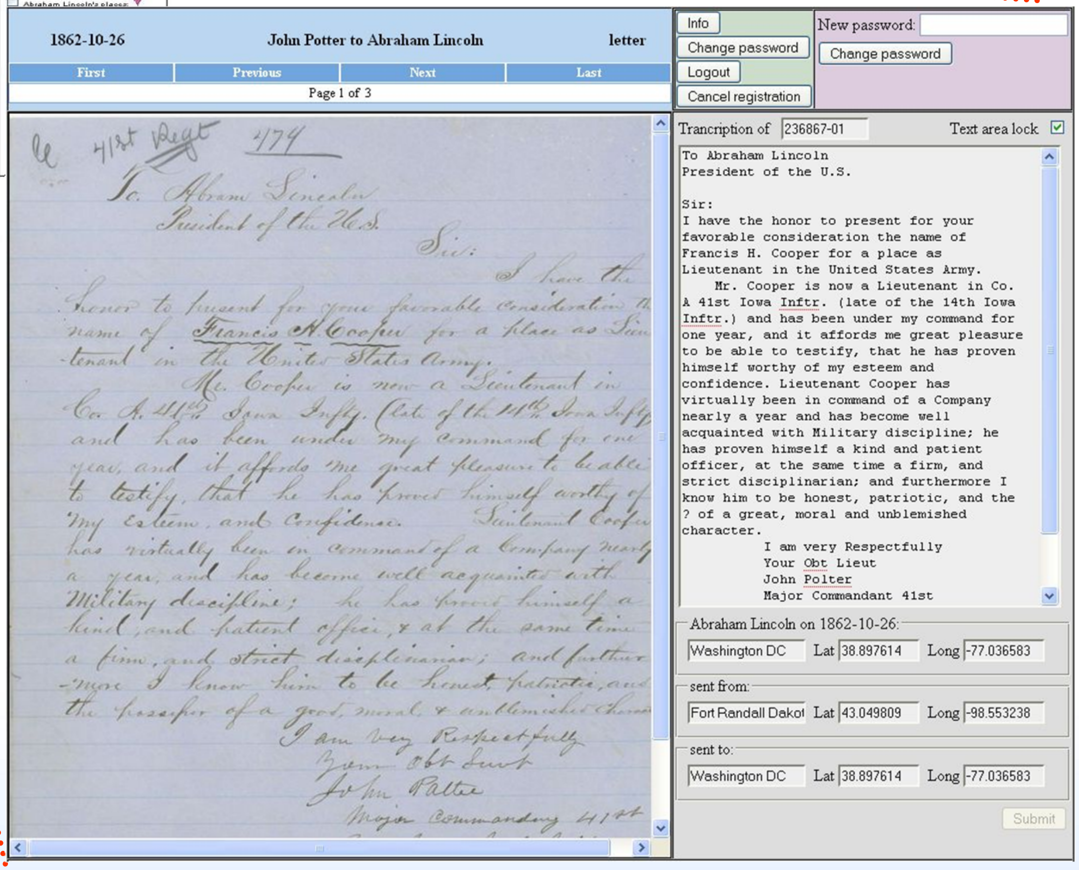
- $R = d(H_i, H_c) + d(V_i, V_c)$



Lincoln Papers Virtual Observatory



Handwritten Character Recognition

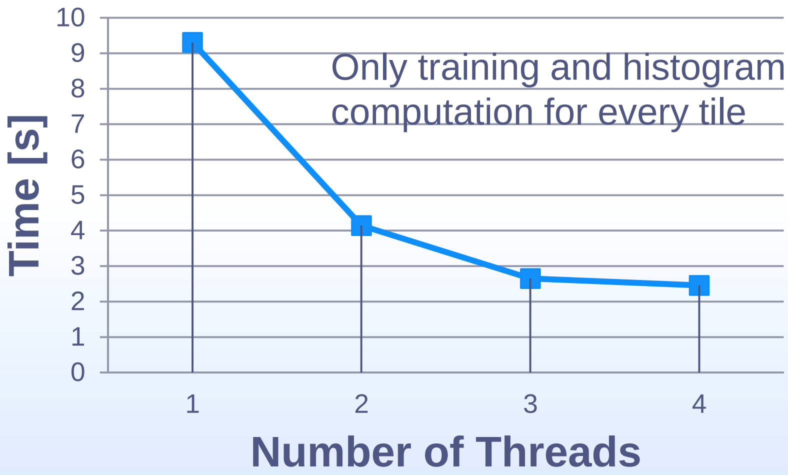


In this example a letter was sent from Fort Randall to President Abraham Lincoln on October 26, 1862.

Summary

- Processing challenges
  - Image cropping: 300,000 files times 60 seconds per file on a single core machine = 5,000 hours = 208.3 days
  - File format conversions (TIFF->PDF/A)
  - Image pyramid construction for web deployment
  - (Handwritten character recognition)
- Mathematical operations
  - Color space representations
  - 2D Histogram
  - Histogram comparison
  - Supervised or unsupervised classification

Parallel implementation of the image cropping algorithm using Parallel Colt and Java Parallel IO libraries
Hardware:
SunFire V40z, 4 Dual Core AMD Opterons, 32GB RAM,
Linux, Java 1.6.0_05, NFS (no parallel I/O)



Summary

Focused on Open Problems in Humanities

- Open problems related to foreground detection
  - Image cropping to remove color scale bars (Lincoln Papers)
  - Template form based matching (1940 US Census)
- Open problems related to text recognition in images
  - Recognition of handwritten characters (Lincoln Papers)
  - Recognition of handwritten form entries (1940 US Census)
  - Recognition of mixed printed and handwritten form entries (Architectural drawings)
  - Recognition of printed text before 1820 with unknown characters to the current OCR packages (18thConnect)
- Open problems related to image content understanding
  - Authorship discovery (Digging Into Data: illustrations in manuscripts, historical maps, photographs of quilts)
  - Content-based search (Compound Media: Adobe PDF and mash-ups)

Many Sources of Images in Humanities

- 2D scanners of planar artifacts:
  - **Paper:** handwritten documents, printed manuscripts, illustrations, historical maps, architectural drawings, …
  - **Vellum:** handwritten documents, maps, illustrations, ..
- 3D scanners of static 3D artifacts:
  - sculptures, archeological findings, historical buildings, furniture, …
- Cameras (visible spectrum) to obtain photographs of 2D and 3D artifacts:
  - **2D objects:** paintings, quilts, coats of arms, murals, …
  - **Static 3D objects**: cylinder seals, clay tablets, historical sites, …
  - **Dynamic 3D objects:** unique 3D activities
- Specialized instruments:
  - X-ray machines, microscopes, infrared cameras, hyperspectral cameras applied to cylinder seals, sarcophagi, palimpsests, …

Opportunities to Automate and Scale with Mathematica and gridMathematica

- Image segmentation (local comparisons of pixels and image features)
  - Supervised (template shape-based map and illustration segmentation)
  - Unsupervised (similarity-based, ball-based)
- Image object recognition (model-based, correlation-based)
  - Line detection and classification (historical maps)
  - Color calibration object detection and recognition (Lincoln Papers)
  - Character recognition (handwritten or printed manuscripts)
- Image comparison (global comparisons)
  - Image representation, feature extraction, and feature based proximity metrics
- Image registration (feature-based, correlation-based)
  - Control point detection, selection, matching and image transformation image (single and multi-modal, stitching and alignment)

Acknowledgement

- **Collaborators on Historical Map project:**
    - Tenzing Shaw, ECE UIUC; Michael Simeone & Robert Markley, English Department, UIUC
- **Collaborators on Costa Rica 2015 Project:**
    - Rob Kooper, NCSA; Costa Rica Alvaro de la Ossa & Jose Castro & Santiago Nuñez-Corrales from National Center for Advanced Technology Studies - Instituto Tecnologico de Costa Rica (ITCR), National Center for Advanced Technology Studies - Centro Nacional de Alta Tecnología (CeNAT), Costa Rica; Qi Li, student in CS UIUC, Kevin Franklin, I-CHASS, UIUC
- **Collaborators on Lincoln Project:**
    - Daniel Stowell & Stacy McDermott, Lincoln Library in Springfield, IL; Vernon Burton & Kevin Franklin, I-CHASS, UIUC; Melvin Casares, and Jose Castro from Instituto Tecnológico de Costa Rica (ITCR); Piotr Wendykier and James Nagy from Emory University Atlanta

Image Spatial Data Analysis Group

iSDA

Acknowledgement

- **Funding Agencies:**
  - NSF, NARA, NCSA International Partners, Google Summer Code
- **Full Time Employees:**
  - Peter Bajcsy, Rob Kooper, Michal Ondrejcek, Kenton McHenry, and Luigi Marini
- **Students:**
  - Tenzing Shaw, Victoria Winner, Monica Czarny, Alex Lewis, Rachel Samaniego