MAP MOSAICKING WITH DISSIMILAR PROJECTIONS, SPATIAL RESOLUTIONS, DATA TYPES AND NUMBER OF BANDS

Tyler J. Alumbaugh and Peter Bajcsy
National Center for Supercomputing Applications
605 East Springfield Avenue, Champaign, IL 61820
talumbau@ncsa.uiuc.edu, pbajcsy@ncsa.uiuc.edu

ABSTRACT

When researchers are interested in multiple geographic datasets over similar geographic areas, it is oftentimes necessary to mosaic the digital images. The process becomes increasingly difficult as the images vary in projection, spatial resolution, and data representation. We first present a framework to describe this mosaicking process in a formal setting. We resolve the issues of varying projection, spatial resolution, and data representation by means of mappings between sets. We also introduce a method within the framework to mosaic spatially localized digital maps that are otherwise highly heterogeneous. The motivation for our work comes from applications that require computing statistics of map attributes over a set of closed boundaries, e.g., counties. Given our motivation, the objective is to minimize the information lost from the mosaicking process while computing spatial statistics. We strive to meet the objective by proposing a new mosaicking method, defining an error metric and comparing our proposed method with the two mosaicking methods available in the ArcGIS commercial software package. Our obtained results for multiple mosaicking methods show that, for our data sets, the proposed method falls in between the two ArcGIS methods in terms of accuracy and computational requirements, and exceeds both methods in terms of easy usability.

1. INTRODUCTION

A photographic mosaic can be defined as "an assemblage of photographs, each of which shows part of a region..., put together in such a way that each point in the individual photographs appears once and only once... and variation of scale from part to part... is minimized." Mosaicking is then defined as "the process of constructing a mosaic from [these] photographs," (ASCE et al. 1994). Often, mosaicking is used in Geographic Information Systems (GIS) to "stitch together" maps of similar scale and projection type. However, generalized map mosaicking can be a difficult task and may involve combining map images of differing spatial resolution (spatial scale), spectral resolution, number of features per map location, and geographic projection. Furthermore, there are map differences resulting from the digital nature of the map pieces, such as the data type and data structures used to store map information. It is desirable to form a resulting mosaicked image that maintains a consistent geographic projection and resolution throughout the image. Thus, the task of mosaicking is about resolving all map differences automatically to produce one consistent map. In practice, the mosaicking task occurs in those application domains that involve visualization, geographic planning, (e.g., environmental restoration, battlefield preparation, precision farming) or statistical processing and modeling of spatial regions. These differences in data sets arise from multiple types of camera-based aerial or satellite sensors and their associated measurement accuracy, as well as preferred data representations of the sensor-operating agency. Other factors contributing to data heterogeneity include temporally changing data gathering and storage techniques and the application/sponsor specific requirements for every particular data collection.

In general, mosaicking can be done either before or after georegistration. If the mosaicking is done before georegistration, the mosaic is formed based on a selected set of map features and a chosen map transformation model. In this case, we rely on some number of overlapping map features, also called tie points or control points, that are either manually found or algorithmically found by feature extraction routines. A general description of mosaicking before georegistration can be found in (Shao et al. 200). In this work, we focus on mosaicking after georegistration. Our current task of map mosaicking is about automatically resolving dissimilar geographic projections, spatial resolutions, data types and number of bands while assuming that map information is stored in the same data structures. We also assume that while the number of map bands (attributes, features or channels) varies

and is adjusted during mosaicking, the physical meanings of bands in multiple maps do not necessarily correlate. This assumption would not hold in the case of spectral resolution matching (for example, mosaicking maps with hyperspectral bands of varying wavelengths). During the mosaicking process, we rely heavily on the ability to calculate the latitude and longitude of any point on any of the map pieces. In general, it would not be effective to simply tile the images together based on the tie points. Each image may have its own projection type and thus we would simply produce an image where we georeference with one set of equations in one section and with a different set of equations in another part of the image. Also, we would be unable to follow consistent lines of latitude and/or longitude with such a system.

When dealing with images that differ in so many aspects, it can become difficult to decide which mosaicking technique is "best" or if even such a determination is possible. Our proposed theoretical model captures several aspects of the problem by introducing (1) indexed sets of images, (2) data processing constraints, and (3) mappings and operators to resolve map dissimilarities. Based on our model, we define a means by which to determine a "best" mosaicking technique, but other choices are clearly possible. We also attempt to analytically model the effect of various parameter changes on an image, which can only be known through direct calculation.

Given the proposed theoretical model, we present issues related to a map mosaicking implementation and its trade-offs. We show that the method is mostly determined by simply making choices among the various mappings outlined in the next section. The portion that relies on georeferencing capabilities is a straightforward 'tiling' approach. Our proposed method can be described as follows. We resolve dissimilar spatial resolutions, data types and number of bands by adjusting (a) the final spatial resolution to the coarsest resolution, (b) the final data type to the most accurate type and (c) the final number of bands to the maximum number of bands. Resolving dissimilar geographic projections is addressed by choosing one of the images as the projection type for the mosaic image, slicing up the remaining images into small pieces, and inserting each small piece into the mosaic at a location that maintains correct georeferencing with respect to the originally chosen image.

In order to quantify and compare error associated with multiple map mosaicking techniques, we define an error metric and focus on comparison of the proposed mosaicking method to the two methods available in the ArcGIS commercial software package. Specifically, we compare our method to results obtained with ArcMap and its Spatial Analyst extension. ArcMap is capable of performing mosaicking and reprojecting tasks in two ways. The first is its "on the fly" reprojection which occurs when a user loads two layers that are spatially related but in different coordinate systems. Every data layer loaded after the first is reprojected into the projection system of the first layers using this technique. The second method is its resampling method. This method is quite computationally intensive and can take a long time for files of the size used in this work. We access this functionality by calling a method from the Visual Basic Editor in ArcMap (ESRI, 2003). We conclude by discussing the average obtained error for each of the methods as well as some mention of performance issues

2. THEORETICAL MODEL FOR RESOLVING MAP DISSIMILARITIES

To begin, we formally introduce an indexed set *A*, to hold our universe of possible image sets, a set of map mosaicking constraints, and a formal problem statement. We then propose a theoretical solution for resolving map dissimilarities.

Indexed Set of Images: We are given a set of *n* images, each of which we denote as i_j, $j = 1,...,n$. We call the set of all *n* images ***I***. Every image in ***I*** has a set of parameters: resolution (r), number of bands (b), geographic projection (p), and sample type (s) defined as the number of bits per sample. For all the images, we collect all of the unique parameters into sets ***R, B, P***, and ***S***. (e.g. $S = \{s_1, s_2, \ldots, s_m\}$). Every image can be transformed into another image with any collection of values from any of the sets. We then consider the indexed family of sets, A, which contains sets of images indexed by elements from the Cartesian product $R \times S \times B \times P$. Formally, we write

$$A = \{A_\omega\} = \{i_{j,\omega}\}, \omega \in R \times S \times B \times P, j = 1,...,n.$$

For example, for $\omega = (r_1, s_2, b_1, p_3)$, A_ω is a possible set in A which contains all the images transformed to that set of parameters. We then define $f = \{f_\omega : I \to A_\omega\}, \omega \in R \times S \times B \times P$ to be the indexed family of functions from our original set of images to an element in A.

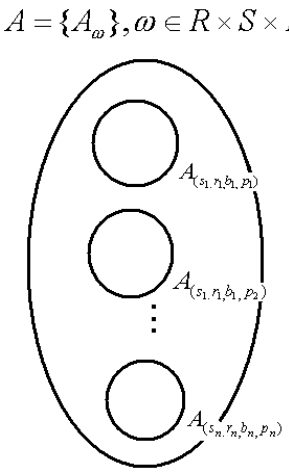

Figure 1. A set of all transformed images based on all combinations of parameters from $R \times S \times B \times P$.

Map (image) mosaicking constraints: In general, it is desirable in any data processing operation to (1) preserve information content, (2) minimize computational requirements for processing and (3) minimize information uncertainty (or maximize data accuracy) of the final data product. We mapped these general requirements for any data processing into a set of constraints imposed on (1) information entropy H that models information content, (2) data size in memory M that corresponds to computational requirements, and (3) spatial error from geographic re-projection E that represents information uncertainty.

Problem statement: Find the 'best' f_ω that will maximize the information entropy (H) and minimize both the size in memory (M) of an image and any spatial error from geographic re-projection (E). Thus, for an indexed set of images i_j, and a function f_ω, we find the optimum choice of $\omega_{optimal}$ as:

$$\omega_{optimal} = \max_{\omega} \sum_{j=1}^{n} \frac{H(i_{j,\omega})}{M(i_{j,\omega})E(i_{j,\omega})} \tag{1}$$

where $i_{j,\omega}$ in A_ω indicates the image i_j and its parameters ω that are used for transforming the image by using the function $f_\omega(i_j)$. The three variables H, M and E can be viewed as separate dimensions of our data with always non-negative quantities (see Figure 2).

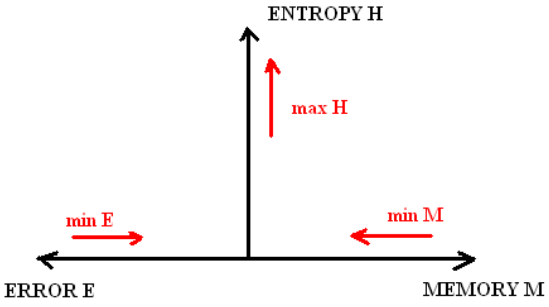

Figure 2. Optimization of entropy, memory and spatial error.

Theoretical Solution: In order to find $\omega_{optimal}$ in the equation above, we develop models for entropy H, memory M and spatial error E, and their changes as a function of $\omega \in R \times S \times B \times P$. Every operation for resolving map dissimilarities might change one or more of the optimized variables H, M and E. We now introduce the list of all considered image operators in Table 1 and show the effect of image operators on the values of

$h = H(i_{j,\omega})$, $e = E(i_{j,\omega})$ and $m = M(i_{j,\omega})$ in Table 2:

Table 1. Definition of operators that changes values of entropy, spatial error and memory size.

Symbols for Image Operators	Denoted Image Operations
$\Delta_{R\uparrow}$	Spatial upsampling.
$\Delta_{R\downarrow}$	Spatial subsampling
$\Delta_{S\uparrow}$	Changing the sample type up (more bits per pixel).
$\Delta_{S\downarrow}$	Changing the sample type down (fewer bits per el).
$\Delta_{P\downarrow}$	Changing the projection type
$\Delta_{B\uparrow}$	Spectral upsampling (larger number of bands)
$\Delta_{B\downarrow}$	Spectral subsampling (smaller number of bands)

Table 2. Models for changes of entropy h, spatial error e and memory size m due to individual image operators described in Table 1. The g entries denote a model that has to be obtained either experimentally or analytically using a known image content.

	$\Delta_{R\uparrow}$	$\Delta_{R\downarrow}$	$\Delta_{S\uparrow}$	$\Delta_{S\downarrow}$	$\Delta_{P\downarrow}$	$\Delta_{B\uparrow}$	$\Delta_{B\downarrow}$
m	Linearly reasing	Linearly reasing	Linearly reasing	Linearly reasing	Approximately Constant	Linearly reasing	Linearly reasing
h	Constant	g_1	Constant	g_2	Approximately Constant	Constant	g_3
e	Constant	g_4	Constant	Constant	g_5	Constant	Constant

To clarify Table 2, the term "constant" refers to no changes due to an operator. For instance, we cannot affect the amount of information in an image by upsampling the image, thus $\Delta_{R\uparrow} h = h$. The term "linearly increasing or decreasing" refers to a linear model with a multiplicative factor larger or smaller than one. For example, the effect of $\Delta_{S\uparrow}$ (or $\Delta_{S\downarrow}$) on m is modeled as $\Delta_{S\uparrow} m = \frac{\#bits\ old\ sample\ type}{\#bits\ new\ sample\ type} \cdot m$ with the ratio greater than one for $\Delta_{S\uparrow}$ and smaller than one for $\Delta_{S\downarrow}$. The term " approximately constant" refers to slight changes that should not have a significant impact on a variable. It is also assumed that any re-projection changes can only increase spatial error with respect to the original image projection and therefore it sufficient to consider only $\Delta_{P\downarrow}$. The g entries in Table 2 can be modeled analytically by assuming a known image content, for example, a checkerboard pattern image (deterministic image content) or constant intensity image with superimposed Gaussian noise (statistical image content). We present an illustration example including such derivation in Section 3.

While measuring memory size and computing image entropy are straightforward calculations, one has to be aware of multiple spatial error metrics. In our investigation, we defined a spatial error metric of an image i_j by computing the difference between any set of features F over defined closed area boundaries before and after re-projection according to Equation (2). For instance, we chose a mean elevation feature within county boundaries for spatial error evaluation in our experimental section that follows the definition below. We present an experimental evaluation of spatial error in Section 5.

$$Error_{\text{NewProjection}}(i_j) = \frac{\sum_{k=1}^{\#boundaries} | F_{\text{OldProjection}}(k, i_j) - F_{\text{NewProjection}}(k, i_j) |}{\#boundaries} \quad (2)$$

3. ANALYTICAL MODELING OF G FUNCTIONS

3.1. Illustration Example

Let us consider two images, i_1 and i_2, which are of the same geographic projection and have the same number of bands and the same size. Image i_1 has a resolution of 30 m/pixel, an initial entropy measurement of $h_1 = 76.00$, and a size of $m_1 = 152000$ bytes. Image i_2 has a geographic resolution of 1000 m/pixel, an initial entropy measurement of 85.00, and a size of $m_2 = 425{,}000$ bytes. Also, i_1 has a sample type of float (IEEE floating point numbers represented with 32 bits), while i_2 has a sample type of double (IEEE double precision numbers represented with 64 bits). Since the band number and projection of the two images are both equal, we will ignore their parameters in our set A and corresponding family of functions f_ω. Therefore, $A = \{A_\omega\}, \omega \in \{r_1, r_2\} \times \{s_1, s_2\}$, where $r_1 = 30$, $r_2 = 1000$, s_1= float (or '32'), s_2 = double (or '64'). Throughout, we denote $A_\omega = A_{r_1, s_1}$ as $A_{1,1}$ and similarly for the rest of the A_ω. The description of each set within A is in Table 3.

Table 3. Description of the input parameter set A.

Spatial Resolution\Sample Type	Float ('32')	Double ('64')
30m/pixel	$A_{1,1}$	$A_{1,2}$
1000m/pixel	$A_{2,1}$	$A_{2,2}$

We are searching for the optimum choice of f_ω as:

$$\omega_{optimal} = \max_\omega \sum_{j=1}^{2} \frac{H(i_{j,\omega})}{M(i_{j,\omega})}; \omega \in (30,1000)x('32','64') \quad (3)$$

Geometrically, we want to maximize the slope of the line determined by a particular choice of f_ω:

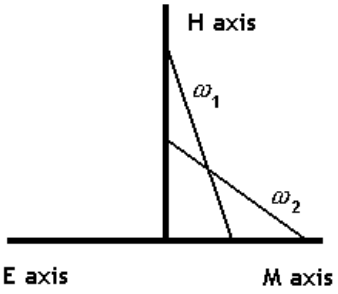

Figure 3. Optimization along H and M axes.

The calculations below evaluate changes of h and m due to all possible image operators to match spatial resolution and sample type variables.

For i_1, we have:

$$A_{11} : \frac{h_1}{m_1} \Rightarrow \frac{h_1}{m_1} \quad , A_{12} : \frac{h_1}{m_1} \Rightarrow \frac{h_1}{2 \cdot m_1} \quad (4)$$

$$A_{21}:\frac{h_1}{m_1}\Rightarrow\frac{\Delta_{R\downarrow}h_1}{\Delta_{R\downarrow}m_1}=\frac{g_1(h_1)}{0.03m_1},\ A_{22}:\frac{h_1}{m_1}\Rightarrow\frac{\Delta_{R\downarrow}h_1}{\Delta_{R\downarrow}(2m_1)}=\frac{g_1(h_1)}{0.06m_1} \tag{5}$$

For i_2, we have:

$$A_{11}:\frac{h_2}{m_2}\Rightarrow\frac{\Delta_{S\downarrow}h_2}{\Delta_{R\uparrow}(0.5m_2)}=\frac{g_2(s_2)}{16.66m_2},\ A_{12}:\frac{h_2}{m_2}\Rightarrow\frac{h_2}{\Delta_{R\uparrow}m_2}=\frac{h_2}{33.33m_2} \tag{6}$$

$$A_{21}:\frac{h_2}{m_2}\Rightarrow\frac{\Delta_{S\downarrow}h_2}{0.5m_2}=\frac{g_2(s_2)}{0.5m_2},\ A_{22}:\frac{h_2}{m_2}\Rightarrow\frac{h_2}{m_2} \tag{7}$$

For our choice of f_ω, we must determine the functions g_1 and g_2 in order to find the optimal ω:

$$\omega_{optimal}=\max_\omega\{(A_{1,1}^{i_1}+A_{1,1}^{i_2});(A_{1,2}^{i_1}+A_{1,2}^{i_2});(A_{2,1}^{i_1}+A_{2,1}^{i_2});(A_{2,2}^{i_1}+A_{2,2}^{i_2})\} \tag{8}$$

3.2. Analytical Modeling of $g_1(h)=\Delta_{R\downarrow}h$ Function

The effect of subsampling on the entropy of an image is difficult to model given its dependence on the image itself. We experimented with various real images and some simulated images to motivate our choice of model. Intuition would suggest that since an image contains less data after subsampling, its entropy should decrease. This is indeed the overall trend, and our results indicate that there is a mostly linear relationship between the subsampling ratio and the decrease in image entropy. So, we selected a simulated image to provide some model for this dependency. Our simulated image was a standard white and black checkerboard with Gaussian noise added to perturb the pixel values. We noted its entropy values for various subsampling rates, and calculated the linear function which most closely modeled this behavior, in a least squares sense. Table 4 summarizes the results:

Table 4. Entropy as a function of subsampling ratio. The difference entropy value is obtained by subtracting the entropy of subsampled data from the entropy of original data (subsampling=1).

Subsampling Ratio	Entropy Value	Difference Entropy Value
1	84.147194	0
2	84.17162	-0.024426
4	84.27841	-0.131216
6	83.55234	0.594854
8	83.813095	0.334099

Performing a least squares fit on the data in columns 2 and 3 yields the following approximation:

$$g_1(h_{orig})=(-0.2467\cdot\left(\frac{r_{orig}}{r_{new}}\right)+.481468)+h_{orig} \tag{9}$$

where r_{orig} is the original spatial resolution of the image and r_{new} is the desired spatial resolution.

3.3. Analytical Modeling of $g_2(h)=\Delta_{S\downarrow}h$ Function

The analytical modeling was performed with simulated images (as in Sec. 3.2). We chose to collect entropy data on one simulated image (a checkerboard image with Gaussian noise) and a real image depicting elevation data since

entropy calculations cannot distinguish between noise and true information. The entropy of these images as a function of sampling rate decreased drastically (approximately eight-fold) when we decreased the sample type by two orders of magnitude. The dramatic drop in a checkerboard image occurred when we went from type double (2^6 = 64 bits) to type short (2^4 = 16 bits). A similar change occurred in the elevation data when going from type float (2^5 = 32 bits) to type byte (2^3 = 8 bits). Based upon these experimental results, we choose to model image entropy as a function of bits per pixel in the following way. Our entropy predictor, $\tilde{g}_2$, will be dependent on the following quantities:

$H(i_{j,\omega}^{orig}) = h_{orig}$ = the original entropy of the image

$b_{orig} = \lg_2(s_{orig})$ = logarithm base two of the original number of bits per pixel for the image

$n = \lg_2(s_{new})$ = logarithm base two of the number of bits per pixel for the proposed sample type

We take $\tilde{g}_2$ as a function of *n* and for a given image, $\tilde{g}_2(n)$ should produce output as follows ($0 < \Delta_1 < \Delta_2 < \Delta_3$):

$$\tilde{g}_2(b_{orig}) = h_{orig} \tag{10}$$

$$\tilde{g}_2(b_{orig} - 1) = h_{orig} - \Delta_1 \tag{11}$$

$$\tilde{g}_2(b_{orig} - 2) = \frac{h_{orig}}{8} - \Delta_2 \tag{12}$$

$$\tilde{g}_2(b_{orig} - 3) = \frac{h_{orig}}{8} - \Delta_3 \tag{13}$$

For all other values of n, we are not concerned with the value of the entropy predictor $\tilde{g}_2$. Thus, the two essential quantities we wish to model are: 1) a reduction in entropy by a factor of 8 when sampling two orders of magnitude below original sample type and 2) the effect of a slowly decreasing linear function. The function, $g_2(n)$, satisfies these properties:

$$g_2(n = \lg_2(s_{new})) = h_{orig}\left(\frac{2^{n-1}}{2^{b_{orig}}} + T_1 + T_2 + T_3\right) + m \cdot (2^{b_{orig}} - 2^n) \tag{14}$$

where T_1, T_2, and T_3 are defined as:

$$T_1 = \frac{(n - (b_{orig} - 2)) \cdot (n - (b_{orig} - 3))}{12}; T_2 = \frac{(n - b_{orig}) \cdot (n - (b_{orig} - 2)) * (n - (b_{orig} - 3))}{\frac{-24}{7}} \tag{15)(16}$$

$$T_3 = \frac{(n - b_{orig}) \cdot (n - (b_{orig} - 1)) \cdot (n - (b_{orig} - 2))}{-96} \tag{17}$$

The *T* polynomials are Lagrange-type polynomials chosen to introduce cancellations when needed to model the large decrease in entropy we observed. The last term, $m \cdot (2^{b_{orig}} - 2^n)$, simulates a slowly increasing linear function (decreasing if we decrease bits per pixel as we go to the right in our charts). For now, we choose *m* = -0.05. One can verify that the above equations for $n = b_{orig}$ lead to h_{orig}.

Table 5: Measured and predicted values for g_2 (entropy dependency on sample type) for the checkerboard image with Gaussian noise.

s – bits per pixels	$n = \lg_2(s_{new})$	$h_{measured}$	$g_2(n) = h_{predicted}$
64	6	85.1208	85.12081
32	5	85.1208	83.5208
16	4	16.89748	8.24
8	3	16.89748	7.84

Table 6: Measured and predicted values for g_2 (entropy dependency on sample type) for the elevation image.

s – bits per pixels	$n = \lg_2(s_{new})$	$h_{measured}$	$g_2(n) = h_{predicted}$
32	5	75.161194	75.161194
16	4	69.760345	72.7612
8	3	2.9766	8.19515

The results of Sections 3.1 and 3.2 can be directly applied to determine $\omega_{optimal} = \max_{\omega}\{(A_{1,1}^{i_1} + A_{1,1}^{i_2}); (A_{1,2}^{i_1} + A_{1,2}^{i_2}); (A_{2,1}^{i_1} + A_{2,1}^{i_2}); (A_{2,2}^{i_1} + A_{2,2}^{i_2})\}$. Using the models of g_1 and g_2 given in this section, working out our numerical examples yields the results shown in Table 7

Table 7. Numerical values for each choice of map parameters.

	$A_{1,1}$	$A_{1,2}$	$A_{2,1}$	$A_{2,2}$
$\frac{H(i_1)}{M(i_1)} + \frac{H(i_2)}{M(i_2)}$	0.0005697	0.00028497	0.01729	0.008

Thus, we select $\omega = (r_2, s_1)$ for $\omega_{optimal}$ and consider it to be the 'best' set of parameters given our theoretical model.

4. MAP MOSAICKING IMPLEMENTATION

With the above theoretical framework in mind, our problem is now to mosaic a set of transformed images given by the selection of a particular f_ω. In this section, when appropriate, we highlight how and why we chose to eliminate possible mappings from ***I*** to ***A***. We arrive at a single set of transformed images from which we compose our map mosaic. From a software design view point, all decisions on how to resolve map dissimilarities are addressed automatically based on map descriptive (header) information before the final mosaicked map is formed (and any memory allocated).

4.1. Map Mosaicking Prerequisites

In order to perform map mosaicking, one needs software tools to load maps into data structures, extract georeferencing information, visualize maps and perform coordinate transformations that are prevalent in the GIS domain. In this work, we have used a library of software tools called Image To Knowledge (I2K) developed by the Automated Learning Group (ALG) at the National Center for Supercomputing Applications (NCSA).

I2K is a Java application for the analysis and visualization of many image formats, including those commonly used in remote sensing applications. A discussion of the various file formats and I/O issues surrounding the handling of georeferenced images in I2K is beyond the scope of this document, but a detailed description can be found in the I2K documentation (Bajcsy et al. 2001). The details about extracting georeferencing information and coordinate transformations are described in (Alumbaugh et al. 2002).

4.2. Resolving Dissimilar Spatial Resolution And Execution Of Adjustments

Our choice of spatial resolution for the final mosaicked image is the coarsest resolution from among all of the images (regarding horizontal and vertical spatial resolution as separate). That is, we now consider only those f_ω where $\omega \in \{r_c\} \times S \times B \times P$, $r_c = \max\{r : r \in R\}$. In practical terms, we subsample each of the images before going on, maintaining a separate ratio for the vertical and horizontal directions. After this subsample stage, we no longer need be concerned with spatial resolution issues, as every image is the same resolution in both the vertical and horizontal directions.

4.3. Resolving Dissimilar Data Types And Numbers Of Bands

Before we can actually allocate memory space for the final mosaicked image, we must decide what data type to use and the number of bands to have in the image. In our approach, we seek to preserve as much information as poss

ible when it comes to data types and number of bands. Therefore, if one map piece has three bands and the rest have one, we alter the other images so that they contain three bands, each with the same information. If one image has type double (IEEE 64-bit floating point number) and the rest have type float (IEEE 32-bit floating point number) then we convert each of the float images to double, using built-in conversion utilities in I2K. In the language of the previous section, we further restrict our possible sets in A to those where $b = b'$, $b' = \max\{b : b \in B\}$ and those sets in A where $s = s'$, $s' = \max\{s : s \in S\}$. The software can handle images with any number of bands in all standard primitive data types (byte, short, int, long, float, double). Other approaches (those that do not choose the largest sample type) would be concerned with loss of information due to a decrease in data type. That is, they would want to know the effect of $\Delta_{S\downarrow}$ on h (represented by g_2 in Table 2). In that case, a model for the entropy loss, such as that presented in the previous section, would be helpful in determining the best choice.

4.4. Resolving Dissimilar Geographic Projections And Execution By Image Splitting

A fundamental question in the mosaicking process is what map geographic projection should be preserved among all maps. In the proposed mosaicking method, a geographic projection of the most northerly map is chosen, although the choice could be left up to the user. Simply for consistency, we note that this represents an arbitrary selection of some single p_i from the set P. The remainder of this section describes how we go about executing this change of projection.

Before converting all other geographic projections into the chosen one, it is necessary to determine dimensions of the final image by examining the geographic placements of all input maps. To find the final image dimensions in a computationally efficient way, the mosaicking method makes a 'best guess' by iterating along the edges of each image (map) looking for a latitude and longitude value that is most distant from our chosen image.

Execution of image splitting: Given a geographic projection, a map defined in another geographic projection is converted first by image splitting, georeferencing it in a desired projection and then inserting it into the appropriate place of the final mosaicked image. The procedure can be described as follows. First, a block size for each image piece is chosen with the default block size of 10 by 10 pixels. Second, the image is then 'chopped' into pieces of this size starting from the upper left and working down to the lower right, with any left over bits handled with appropriately smaller blocks. Third, image blocks are inserted into the final mosaicked image. For each block, we can calculate the column and row in the new image array that most closely corresponds to the latitude and longitude of the upper left corner of that block. An I2K GeoConvert object makes this calculation for us. We then copy the contents of the image block into the appropriate entries of the image array in the mosaic image. This image splitting technique has the benefit of being instantly applicable on any projection supported in I2K and is, in fact, independent of the projection of any of the images we wish to mosaic.

5. COMPARATIVE EXPERIMENTAL EVALUATIONS OF SPATIAL ERROR

In order to evaluate the developed mosaicking method, we focused on the error introduced by converting dissimilar geographic projections. The adjustments due to dissimilar data types or numbers of bands did not have an impact on the error of the final image. The adjustment of spatial resolution could be user driven so that a user would choose the final spatial resolution based on the trade-offs between memory requirements of the final image and accuracy of the spatial information. Thus, the loss of information due to spatial resolution adjustments could be zero by selecting the finest resolution among all maps and we have not evaluated it.

We describe next (1) the experimental data sets, (2) the proposed error metric and (3) the conducted experiments with the data sets using the error metric for evaluating multiple mosaicking methods followed by (4) obtained results.

5.1. Description of Experimental Data Sets

The error introduced by converting dissimilar geographic projections is evaluated with three data sets defined in three different geographic projections. The three datasets were: a National Elevation Dataset image for the state of Illinois, LandSAT data in the UTM Zone 15 projection, and a categorical forest label dataset for the United States in the Lambert Azimuthal Equal Area projection (obtained from the USGS).

5.2. Error Metric

In general, an error metric definition is application dependent. In this work, we chose our end GIS application of map mosaicking to be feature extraction over a set of geographic boundaries. In particular, we chose mean elevation values within county boundaries. The error measurements were made as follows:

$$Error_{UTM} = \frac{\sum_{i=1}^{\#counties} | \mathrm{E}_{NED}(i) - \mathrm{E}_{UTM}(i) |}{\#counties} \tag{18}$$

$$Error_{Lambert} = \frac{\sum_{i=1}^{\#counties} | \mathrm{E}_{NED}(i) - \mathrm{E}_{Lambert}(i) |}{\#counties} \tag{19}$$

Here $E_{NED}(i)$ is the evaluation of mean elevation at the *ith* county for the original NED dataset. The other map projections have similar elevation functions. A small technical note: the NED dataset for the state of Illinois came on two separate CDs. For simplicity, the ArcMap processing was only done on one of them. Thus, the ArcMap error rates were computed over approximately half of all Illinois counties. One example of the end GIS application would be finding average elevation and spectral signature over all counties in Illinois from multiple maps and visualizing them together. Thus, we can assess the error (or accuracy) of mosaicking by comparing the extracted features of the same geographic boundary from a single map and from a mosaicked (and reprojected) map.

The NED dataset was mosaicked with a LandSAT image in a UTM Northern Hemisphere projection in one image and a larger scale Forest Label image in a Lambert Azimuthal Equal Area projection in another. An ESRI Shapefile of the county boundaries for the state of Illinois were overlaid on the mosaicked images to show proper georegistration and to extract map average elevation features. The differences between the features extracted from a mosaicked image and a single map are presented to demonstrate the accuracy of the developed mosaicking technique. Thus, we produce error measurements from the UTM mosaicking and the Lambert mosaicking for the proposed I2K method, the ArcMap resampling method, and the ArcMap on the fly reprojection.

5.3. Mosaicking Experiments

The I2K results were achieved through the use of the Load Many GeoTiles tool. For each evaluation, we loaded the NED dataset file and another file with dissimilar georeferencing and data representation properties. A shapefile for the counties of the state of Illinois was overlaid. Average elevation values were calculated for the counties and saved out to a dbf file with a GeoFeature tool (Bajcsy et al. 2001). These average elevation values were compared to the average elevation values obtained by loading the NED dataset by itself (thus without re-projection) and using the same computation procedure and the same GeoFeature tool as before.

The ArcMap results below were achieved by a slightly more complicated three-step procedure. First, the Illinois portion of the National Elevation Dataset (NED) was obtained from the USGS in GridFloat format. The dataset was imported to the native 'Grid' format for ArcMap using the ArcToolBox. The resulting grid was added to a blank map as a Layer. A standard Shapefile containing the boundary information for the counties in Illinois was added as another Layer. Viewing both at once in the map showed that ArcMap was able to correctly georeference the elevation data. Second, the Spatial Analyst tool was loaded into ArcMap. In the Zonal Statistics dialog box, there is an option to compute the mean value of each zone, where the user can define the zone. The default option for defining the zones is set for the COUNTY label defined in the dbf file associated with the Shapefile. Finally, ArcMap computes the average elevation value of each zone and puts the information in table that the user can save out. This gave the E_{NED} scores for all the counties.

The average elevation results for the resampled NED data for Illinois were obtained as follows. The reprojection was done by a Visual Basic Script found in the Knowledge Base section of the ESRI support site (ESRI 2003). The method supports single band mosaicking (which is all that was needed here), but there exist workarounds (through the MAKESTACK command) to mosaic a multiband image. First, the NED data from the USGS was imported to a Grid as described above. Then, the script below was entered in as the source code in the 'ThisProject' source window. Some values were changed to get the correct working directory and filenames. Also, the script was modified to reproject the raster to UTM Zone 15 (esriSRProjCD_NAD1983UTM_15N) and then to the Lambert Azimuthal Equal Area projection for North America (esriSRProjCS_NAD1983N_AmericaLambert). The rasters were saved out in the TIFF format. Each TIFF image was reloaded in a new, blank map. Zonal statistics were again calculated as described above.

The mean elevation results for the 'on the fly' reprojection were obtained with a combination of ArcMap and I2K functionality. First, either the LandSAT data or the forest label was loaded. Then, the NED dataset was loaded. ArcMap warns the user that the dataset is in a different projection, but will load the NED dataset anyway, performing its 'on the fly' projection to make a consistent image. This image was exported to TIFF, and then the elevation values were evaluated in I2K using the manner given above. ArcMap will not allow zonal statistics to be computed for the 'on the fly' projection, so the exporting of the image was necessary.

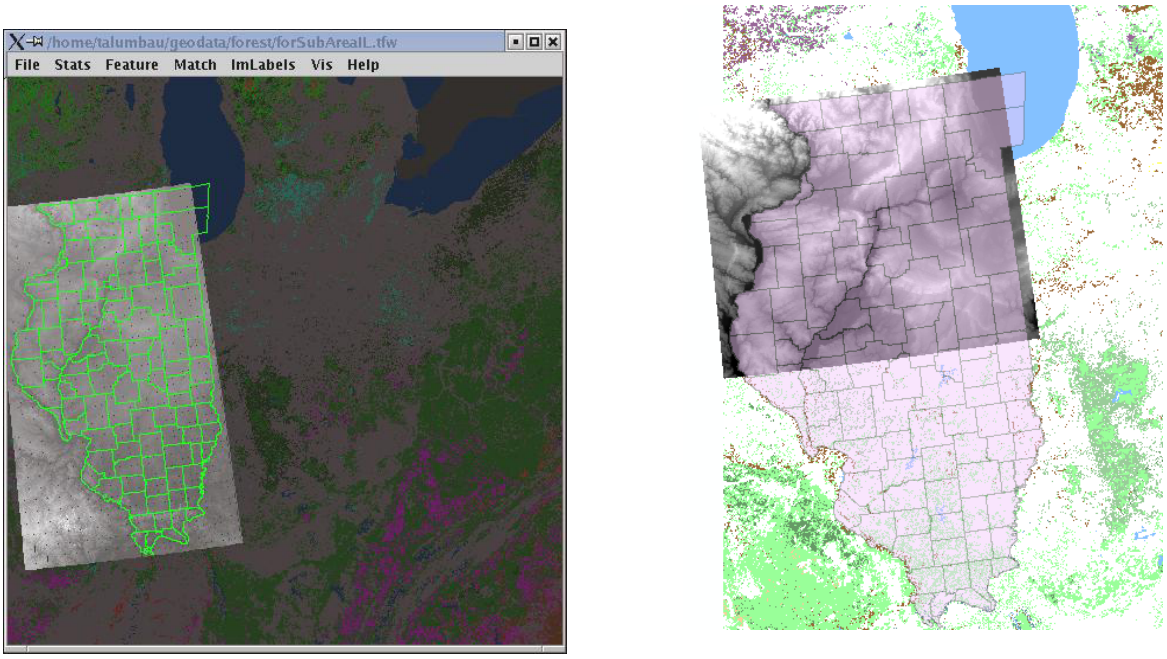

Figure 3. NED data mosaicked with forest label data in Lambert projection (I2K left, ArcMap right).

Results

In Figure 3, the NED image is combined with the forest label data. The three-band, RGB values of the categorical data have been converted to three float bands of the same values. In the forest label data, the elevation dataset has been extensively subsampled due to drastic differences in the resolutions. The one arc-second resolution

of the NED is roughly equivalent to 30 meter/pixel resolution. However, the forest label data has a resolution of 1000 UTM meters for each pixel. Thus, we can expect a certain loss of precision in the NED portion of the image, along with any inaccuracies resulting from the mosaicking process.

Table 8. Geographic projection errors computed using an error metric defined in and evaluated for the developed I2K mosaicking method and ArcMap resampling and 'on the fly' reprojection methods.

	$Error_{UTM}$	$Error_{Lambert}$
I2K Mosaicking method:	0.013743485	0.045777147
ArcMap Resampling	0.000074072	0.000054601
ArcMap 'on the fly' reprojection	0.068954284	0.054499059

6. SUMMARY

The I2K mosaicking method appears inferior to the more computationally intensive resampling method. However, the method does appear to introduce less error for our metric than the "on the fly" reprojection from ArcMap. Getting ArcMap to correctly perform the desired reprojections and statistical processing was a laborious process, showing ArcMap as highly capable but not especially inviting to the novice. It is likely that the I2K method could be implemented to run very fast in the ArcMap package. Direct time trials were not discussed here because the ArcMap tools run in native code, while I2K needs a Java Virtual machine. Further, the I2K method relies heavily on dozens (if not hundreds) of instantiations of objects, which is a notoriously slow process. It is essentially just memory accesses, though, so more efficient execution (even in the Java implementation) is clearly possible.

REFERENCES

Alumbaugh, T. J. and P. Bajcsy (2002). "Georeferencing Maps with Contours" http://alg.ncsa.uiuc.edu/documents/TR-20021011-1.doc

American Society of Civil Engineers (ASCE), American Congress on Surveying and Mapping (ACSM), and the American Society for Photogrammetry and Remote Sensing (ASPR) (1994). *Glossary of Mapping Sciences.*

Bajcsy, P. and T.J. Alumbaugh (2003). "Georeferencing Maps With Countours" Journal proceedings of 7th World Multiconference on Systemics, Cybernetics, and Informatics.

Bajcsy P. et al. (2001) "Image To Knowledge (I2K)," Software Documentation at http://alg.ncsa.uiuc.edu/do/index

Burrough, Peter A. McDonnell, Rachael A. (1998), *Principles of Geographical Information Systems.* Oxford University Press.

Curran, Paul J. (1985), *Principles of Remote Sensing.* Longman Group Limited.

ESRI Knowledge Base VBScript site (2003): http://support.esri.com/index.cfm?fa=knowledgebase.techArticles.articleShow&d=20552

Extensive collection of remote sensing imagery for a fee: http://www.spaceimaging.com

Federal Geographic Data Committee. *Federal Geographic Data Committee Metadata Standard*: http://www.fgdc.gov/standards/standards.html

Freely available LandSAT data: https://zulu.ssc.nasa.gov/mrsid/

Imagery for U.S. states and principalities available at the United States Geographical Survey site: http://www.usgs.gov

ESRI (1998). "ESRI Shapefile Technical Description." http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

Shao, Guofan. Zhu, Huazhong. Mills, Walter L. Jr. (2000), "An Algorithm for Automated Map Mosaicing Prior to Georegistration" *Geographic Information Sciences* Vol. 6 No.1 June 2000 p.97-101.