2019

Details of Deformable Part Models for Automatically Georeferencing Historical Map Images

Nicholas Howe
Smith College, nhowe@smith.edu

Jerod Weinman
Grinnell College

John Gouwar
Grinnell College

Aabid Shamji
Grinnell College

Details of Deformable Part Models for Automatically Georeferencing Historical Map Images

Nicholas R. Howe*
nhowe@smith.edu
Smith College
Northampton, MA

Jerod Weinman*
jerod@acm.org
Grinnell College
Grinnell, IA

John Gouwar
Aabid Shamji
Grinnell College
Grinnell, IA

ABSTRACT

Libraries are digitizing their collections of maps from all eras, generating increasingly large online collections of historical cartographic resources. Aligning such maps to a modern geographic coordinate system greatly increases their utility. This work presents a method for such automatic georeferencing, matching raster image content to GIS vector coordinate data. Given an approximate initial alignment that has already been projected from a spherical geographic coordinate system to a Cartesian map coordinate system, a probabilistic shape-matching scheme determines an optimized match between the GIS contours and ink in the binarized map image. Using an evaluation set of 20 historical maps from states and regions of the U.S., the method reduces average alignment RMSE by 12%.

CCS CONCEPTS

• **Information systems → Geographic information systems**; *Content analysis and feature selection*; *Image search*; • **Applied computing → Graphics recognition and interpretation**;

KEYWORDS

GIS, georeferencing, historical maps, vector-image alignment

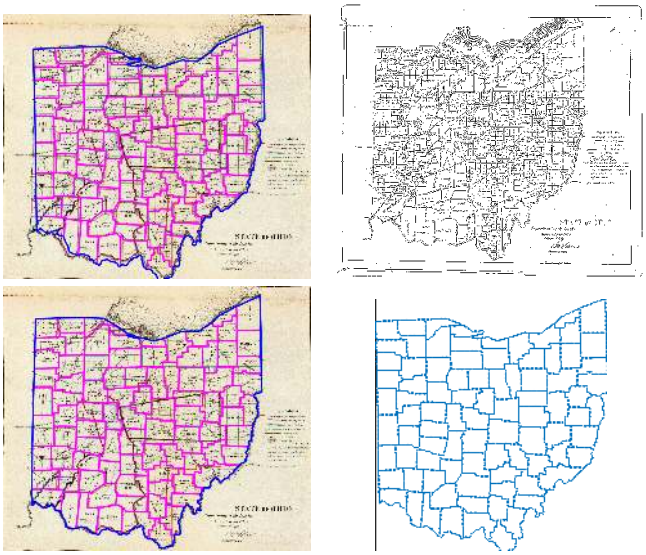

Figure 1: Clockwise from top left: Initial alignment superimposed on a map (D0042-1070001 [2]); map edge skeleton; geometric model from GIS data; final alignment.

1 INTRODUCTION

Maps provide a dense, information-rich history of geography, politics, and power [17]. Maps from the pre-digital era have become increasingly available through online archives. Though coarse geographic metadata (e.g., the map region) usually accompanies these artifacts, more precise geographic alignment is available far more rarely at present. Manually creating such alignments requires painstaking effort by human annotators [11]. However, blending views of maps across different eras or modalities can provide powerful new cues to historians and other scholars [20].

This work offers a method for improving automated alignment of historical map images to the Earth's geography, a process known as *georeferencing* [13]. Specifically, we build on prior work that provides an approximate alignment based purely on the arrangement, textual content, and typography of *toponyms* (place names) on the map [26, 27]. This work adapts a shape-matching algorithm (related to those previously used for word spotting [15, 16]) to the task of matching contours from GIS data—geographical and political boundaries, roadways, etc.—to historical map image contents.

Historical GIS data is not widely available. However, our model's flexibility allows it to align contemporary GIS data to historical maps reasonably well. Our primary contribution includes development of this flexible shape-matching strategy to improve an approximate initial alignment.

We have added extensive ground truth correspondences to an existing dataset of textually annotated historical maps to facilitate geographic empirical analyses. Whereas previous work demonstrated the utility of approximate georeferencing by measuring a reduction in OCR (text recognition) error, the augmented benchmark map data in this work allows us to precisely measure the improvement in georeferencing of the previous approach as well as the newly proposed techniques.

Section 2 describes the necessary background, including related work and basic geographical practices. Section 3 details the approach and Section 4 evaluates the model both qualitatively and quantitatively.

This technical report is an extended version of the short paper entitled "Deformable Part Models for Automatically Georeferencing Historical Map Images" which was published in Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'19), 2019. DOI:10.1145/3347146.3359367

*Both authors contributed equally to the paper

2 BACKGROUND

The problem of aligning data from multiple modalities arises in many fields but particularly in geospatial processing. Hackeloeer et al. [13] enumerate approaches and applications. Here we focus on the specialized task of aligning vector data to raster map images.

2.1 Related Work

To align vectors and geographical images, there is a long history of using road lines or intersection points [3, 4, 18], which tend to be very stable (some modern road paths date to ancient Roman times [5, p. 1:35]). Although fast, these methods often assume the matching involves only scale and translation [3]. Li's method [18] included rotation, but it may not be particularly robust to noise [5, p. 1:36]. Weinman's toponym-based methods [26, 27] require no such assumptions, robustly aligning with a full affine transform using only map region metadata and a gazetteer (cf. Section 3.2.1).

We can further distinguish between orthoimagery [3, 4, 18] and map images as the alignment target. In the latter category (to which this work also belongs), Duan et al. [6] recently proposed a method that adjusts an initial vector-image alignment by searching for consistent local control point deformations that improve the putative alignment. They include a raster foreground extraction method that identifies colors of interest based on a narrow band around the initial vector alignment, resulting in a binarized image for matching.

Although other recent methods have trained CNNs for alignment [21] and interpolation [28], our deformable part model requires no training data and is based solely on a geometric consistency score and a simple distance metric. Initially proposed by Fischler & Elschlager [10], deformable part models have more recently been applied to human pose recovery [8] and visual object recognition [7]. Work in document analysis has developed them into tools for matching 2D arrangements of curvilinear segments, with applications to word spotting [15] and signature verification [16].

2.2 Geographic Coordinates and Projections

Geodesists have updated our knowledge of the Earth's shape over many centuries, describing it approximately as an ellipsoid. Geographers create a *datum* by choosing a particular alignment of the ellipsoid to the surface of the earth. A *geographic coordinate system* (GCS) assigns an origin—equator and prime meridian—for all points on that surface. In this work, we employ the North American Datum of 1983 (NAD83), presently used for geospatial work throughout the United States federal government. Before this, the North American Datum of 1927 (NAD27) used the Clarke 1866 model as the reference ellipsoid. Although that shape would likely have been used in our test maps, the discrepancy should be negligible at their scales. Rather, a much greater degree of error will likely be due to our ignorance of the *projection* used by the cartographer.

Printed maps must represent the three-dimensional, ellipsoidal structure of the Earth on a two-dimensional plane. Cartographers have created a wide variety of projections for transforming spherical geographic coordinates to Cartesian map coordinates [22]. In this work we follow Weinman [27], focusing only on two projection families that appear manifest in the test maps: cylindrical and conic. Specifically, we consider the equal area cylindrical projection and Hassler polyconic projection. Both projections require a standard

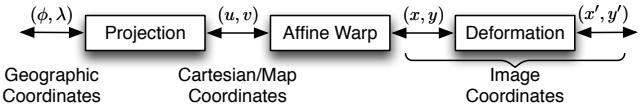

Figure 2: Coordinate transformation sequence used for automatic georeferencing. After an initial affine warp, the final deformation may be another rigid affine warp or a non-rigid thin plate spline interpolation.

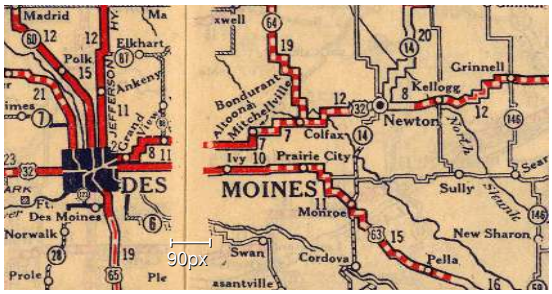

Figure 3: Extra margins leave a gap at the binding of two leaves constituting a single map (D0117-5755024 [2], original map copyright ©1927 Rand McNally).

parallel (often the Equator) while the polyconic projection also requires a central meridian where the cones are tangent to the globe. Specific settings for these vary by map and the precise method is described in further detail below. The technique (cf. Section 3.2.1) is not limited to these projections; indeed, any projection may be included as a candidate over which to optimize.

3 METHODS

We seek to produce a mapping $(x', y') \leftrightarrow (\phi, \lambda)$ that associates a position on the map image to its equivalent geographic coordinate (latitude ϕ and longitude λ) within the map boundaries. Figure 2 shows the overall geometric processing framework of our approach. Estimation of transformation parameters based on toponyms can provide an effective starting point and baseline [27], but often exhibits visible systematic errors (see Figure 1). The methods described below give a more precise alignment by matching GIS vector data to the visual structure in the map image.

Because historical maps vary in the precision and accuracy of their details, it may be impossible to find a perfect affine or other rigid match to modern geospatial data. Printing errors may also introduce systematic discrepancies (as shown in Figure 3).

We therefore adapt the inkball model [15], a flexible matching technique previously used for word spotting. These deformable part models represent curvilinear segments as linked chains of control points that can move independently while seeking to retain their original shape as much as possible. Each control point tries to find a location in proximity to document ink that also maintains its position relative to linked neighbors. Deviations from either goal are penalized via a quadratic spring-like potential. Jointly optimizing all point locations is difficult, but efficient algorithms exist when the link structure forms a tree [8]. While many deformable part models

use tree structures for more efficient inference, this work employs general graph models instead. In applying these techniques to map georeferencing, models will represent features such as roads and the boundaries of states and counties; accounting for loops is therefore essential. Loops complicate the model-fitting process, but with the right algorithms useful results may still be achieved in practice.

In the remainder of this section, we first elaborate on the deformable structure of the part models used in this work. We then describe two methods for configuring the models before finally detailing the update process that yields the resulting model fit.

3.1 Model Structure

The inkball models employed in this paper represent curvilinear features (boundaries, roads, rivers, etc.) in a manner that provides spatial flexibility when matching to a map image. The model can be conceptualized as semi-rigid chains of control points that trace a target shape, linked by connections that can stretch or otherwise deform with an energy cost equivalent to a spring-like potential function. When disks of ink are placed at densely sampled control points ("inkballs"), such models can generate plausibly deformed versions of the original shape. The sparse point correspondences found by optimizing the match between such a model and a map raster image can then guide the creation of a dense coordinate transformation that takes an initial rigid coordinate frame (x, y) to the adjusted coordinate frame (x', y'), as shown in Figure 2.

Model building begins by transforming the selected GIS shape features to be matched from geographic to image coordinates (cf. Figure 2) using the best available parameter estimates. Such parameters might be provided manually, estimated from toponym locations, or produced via the novel method proposed below in Section 3.2. We rasterize these shape vectors into a skeleton binary image at the map raster image's native resolution. From this binary image, we choose a set of control points C with locations $\mathbf{v}_i$ for $i \in C$ at junctions plus additional points set apart by a fixed distance (4 pixels). Points that neighbor each other in the rasterized shape image become neighbors in the model, and we record the observed spatial offsets between neighbors $\mathbf{t}_{ij} \triangleq \mathbf{v}_i - \mathbf{v}_j$. Neighbors may be represented via an adjacency matrix N, with $N_{ij} = 1$ when nodes i and j are neighbors (and zero otherwise). In general, this results in a loopy model (a graph with cycles).

3.2 Model Configuration

Here we describe two methods for constructing the model configuration. The first method is based on an approximate georeferencing technique from prior work, while the second introduces a novel multisample consensus-based approach.

3.2.1 Toponym-based Georeferencing. Previous work by Weinman [26] approximately georeferences map images solely for the purpose of improving toponym recognition, assuming that knowledge of a pixel's location on the globe tightly constrains the space of possible toponyms. To that end, each word image goes through a text recognition system, which produces a list of candidate words and their corresponding probabilities. These candidate words, probabilities, and their image locations are used as part of an MLESAC robust estimation process [25] to produce an affine alignment for georeferencing. Because the method is based on a sample consensus focused on rejecting outliers, the resulting maximum likelihood parameter estimates are necessarily quite coarse, even when re-estimated from the final inliers (rather than just the minimum sample set). Moreover, the method uses a fixed error tolerance and inlier probability, both of which can deviate substantially from map to map. To improve this initial toponyn-based alignment, Weinman [27] optimizes the overall probability model via an Expectation-Maximization (EM) algorithm that adjusts all these parameters.

This initial estimation process—whether based on MLESAC or the EM adjustment—finds an affine transform between the projected Cartesian map coordinates (u, v) and the final rendered image coordinates (x, y), as shown in Figure 2. Because the projection used for a given map is unknown, the initial georeferencing process may be run using each candidate projection (just two in the current framework) before choosing the one with the best score.

Not only is the projection family itself typically unknown, the projection parameters are also unknown. We use the map's region (part of the metadata associated with the map) to determine a central meridian and standard parallel required for the unknown map projection parameters. As a proxy for bounding box center, we take these projection parameters to be the graticular center of all the *Populated Place* and *Civil* features from the given region in the Geographic Names Information System (GNIS)[1] provided by the U.S. Geological Survey (USGS).

The projection selected by the algorithm and the affine transformation takes the GIS shape data from geographical (ϕ, λ) space to the image (x, y) space (Figure 2), where the GIS rasterization is performed and model configurations $\{\mathbf{v}_i\}$ are found.

3.2.2 Tree-based Configurations. As an alternative to relying on prior work for the model configuration, we propose a novel approach that employs the same inkball models used for the main model-fitting method (cf. Section 3.3). Because these model graphs have cycles, efficient techniques for finding global optima are not available. However, it is possible to perform efficient optimization on closely related models that are tree structured [8]. The configurations obtained by such tree-structured models tend to be correct in some areas and incorrect in others. Collectively, the configurations obtained from multiple tree-structured models may contain enough information to correct the errors present in individual results.

Model configuration proceeds as follows. First, we take a very crude projection—simply treating the geographic coordinates (ϕ, λ) as Cartesian coordinates (u, v)—and then we independently apply linear scaling so that the target structures occupy 80% of the vertical and horizontal extent of the map. This scaling takes the arbitrary (u, v) Cartesian coordinates of the "projection" to image coordinates (x, y). Next, we create a loopy model as described in Section 3.1 above. From this model, we derive m tree-structured models by selecting a seed node and adding nodes one at a time to generate a random spanning tree. Step by step, a single link is added between the previously incorporated nodes and any unincorporated neighbor chosen at random, until every control point has been included in the tree.

[1] https://geonames.usgs.gov; each GNIS feature with a unique identifier is assigned to a class (e.g., *Arroyo*, *Basin*, *Cliff*, *Dam*, etc.), and is given an official location and name (along with historical or alternate names and spellings).

We then efficiently fit each of the m models to the map as a standard, tree-structured inkball model [15] to give m individual configurations. Finally, we combine these configurations into one by taking the median in each coordinate for every control point i:

$$\mathbf{v}_i \triangleq \text{median}_{k\in[1,m]} \left\{\hat{\mathbf{x}}_i^k\right\} \tag{1}$$

where $\hat{\mathbf{x}}_i^k$ is the estimated coordinate for the ith control point in the kth tree model.

Because the resulting model is flexible and estimated from median tree-fit locations, the combined configuration may not be entirely self-consistent. To remedy this, we fit an affine transformation that brings the raw projected shape points in (u, v) space as close as possible (in a least-squares sense) to the tree-based models' median control point locations in (x, y) space. As in the toponym-based approach, this identity projection and affine transform are used to take the GIS data to image space, where it is rasterized to configure the model structure described in Section 3.1.

3.3 Model Fitting

Given the model structure described above, we search for matching control point locations located on or near the map image ink but otherwise preserving the general model shape (relative positions).

In preparation for this process, the raster (RGB) map image is converted to the desired form via several preprocessing steps. Because the initial image resolutions are very high, they are first downsampled by a factor of 8. RGB is converted to grayscale and then to a binary image using Howe's method [14]. Finally the lines are thinned to one pixel wide [12].

Two potential functions model the joint preferences for each control point to be located on the map image ink and for neighboring control points to have their preferred relative positions. Letting $\mathbf{x}_i$ be control point i's putative location on the image grid, the ink-preferring local potential is

$$\psi_i(\mathbf{x}_i) \triangleq \exp\left\{-D(\mathbf{x}_i)^2\right\}, \tag{2}$$

where D represents the minimum distance from the given point $\mathbf{x}_i$ to the ink (as represented by the binary skeleton), which can be computed efficiently by the distance transform [19]. The interaction potential between two control points for which $N_{ij} = 1$ is

$$\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \exp\left\{-\left\|(\mathbf{x}_i - \mathbf{t}_{ij}) - \mathbf{x}_j\right\|^2\right\}, \tag{3}$$

which decreases with the squared distance of $\mathbf{x}_j$ from its expected location $\mathbf{x}_i - \mathbf{t}_{ij}$.

Model fitting then proceeds via a form of message passing. Each control point maintains a record of what has been determined about its 2D location, represented as a grid of beliefs $b_i(\mathbf{x}_i) \in (0, 1)$ at the same resolution as the map image. Following an initialization step described below, the estimated locations are updated iteratively in turn to take into account information passed to it by its neighbors:

$$b_i^{(t+1)}(\mathbf{x}_i) \propto \psi_i(\mathbf{x}_i) \prod_{j:N_{ij}=1} \max_{\mathbf{x}_j} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)\, b_j^{(t)}(\mathbf{x}_j) \tag{4}$$

These values are normalized so they sum to one over the entire grid. Intuitively, the belief update incorporates the local potential ψ_i, preferring locations near the map ink, but also accounts for the neighboring control point locations so as to optimize the interaction potential weighted by the current belief for those locations. This optimization (inside the product) can be computed efficiently via a linear-time generalized distance transform [9] (cf. Section 3.3.4).

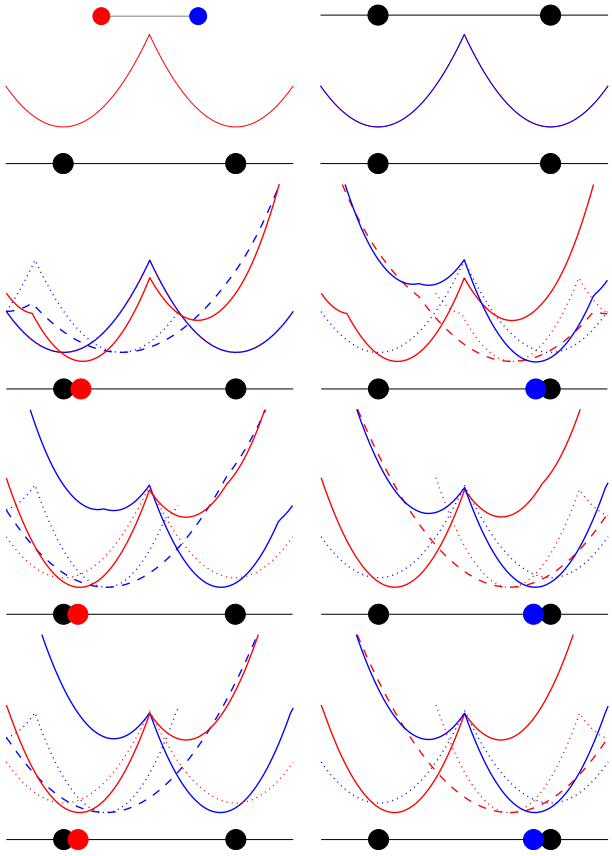

Figure 4: Inference process for a simple 1D model. First row: model with small point separation t_{ij} (left) and target ink with larger separation (right). Second row: initial negative log belief functions $\rho_i^{(0)}$ based on squared distance from ink. Following rows: iterative update steps. To update red distribution, offset blue according to model separation t_{ij} (yielding dotted blue curve), apply the generalized distance transform Γ_{ij} (yielding dashed blue curve), add it to the naive belief $\rho_i^{(0)}$ (yielding dotted red curve) and normalize (yielding new solid red curve $\rho_i^{(t+1)}$). Blue update follows the same steps with reversed roles. Results stabilize after just a few iterations.

After a chosen number of iteration rounds r, the final node positions are set to the most-preferred location:

$$\hat{\mathbf{x}}_i = \arg\max_{\mathbf{x}_i} b_i^{(r)}(\mathbf{x}_i). \tag{5}$$

Figure 4 illustrates the inference process for an extremely simplified 1D model. Figure 5 shows a match in 2D. Figure 6 provides a visualization of the alignment process in action for map D5005-5028102. Many control points find their optimal position after just one iteration round, but some continue to move around and the

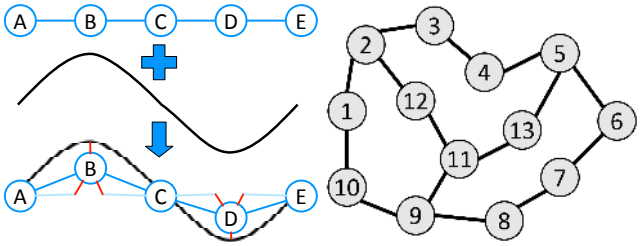

Figure 5: 2D fitting. Left: A straight model fitting to a curvy line will adopt a configuration that compromises between model deformation and proximity to ink. Red lines show tension. Right: Visitation order for a simple graph under the clockwise rule (cf. Section 3.3.2).

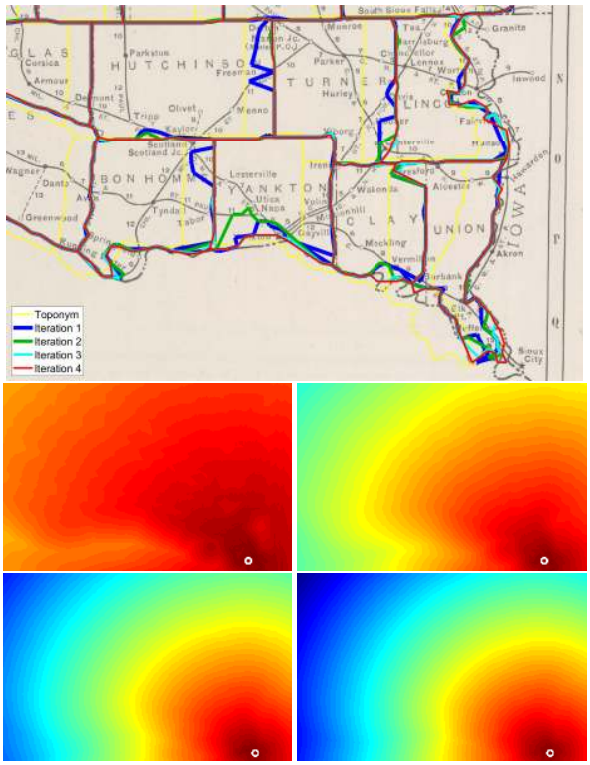

Figure 6: Evolution of alignment. Top: movement of boundaries with iterations. Bottom: Evolving location belief map (square root of log scale, $\sqrt{\log b_i}$) for one control point at the state's southeast corner. (D5005-028102 [2], original map copyright ©1924 Rand McNally)

initially diffuse estimates of the possible location grow more focused over time.

3.3.1 Initialization. The point configuration used to build the model provides a rough, if imperfect, initial position for control points. We use this to construct an initial belief $b_i^{(0)}$ based on the original local potential ψ_i of Equation (2) and a secondary potential ψ_i' preferring locations $\mathbf{x}_i$ near the estimated initial position $\mathbf{v}_i$,

$$\psi_i'(\mathbf{x}_i) \triangleq \exp\left\{-\kappa \left\|\mathbf{x}_i - \mathbf{v}_i\right\|^2\right\}, \tag{6}$$

where $\kappa = 0.01$ is an adjustable scale parameter. We combine these potentials to initialize the belief function,

$$b_i^{(0)}(\mathbf{x}_i) \propto \psi_i(\mathbf{x}_i)\,\psi_i'(\mathbf{x}_i). \tag{7}$$

After initialization, the model positions evolve toward ink, preserving shape, but retaining no further bias for the initial configuration.

3.3.2 Visitation Order. Inference on graphical models with loops is known to be NP-hard [8] and is typically attempted via approximation algorithms. Because convergence to a global optimum is not guaranteed, details of the update process can affect the quality of the final result. We update beliefs incrementally using Equation (4) according to a *clockwise rule* which proceeds as follows: On odd rounds, begin at the $\mathbf{v}_i$ point with the leftmost x-coordinate and move upwards to start. Traverse the graph, choosing the next unvisited neighbor in a clockwise direction from the previous node. Maintain a stack of passed-over branches, so that if the current node has no unvisited neighbors the traversal can be restarted from the next unvisited node in the stack. If the graph is disconnected then each component can be treated in the same manner. On even rounds, use the same order in reverse. See Figure 5 for an example.

3.3.3 Termination. Ideally the message passing would proceed until it converges on the globally optimal solution. In practice the configuration may become stuck in a local minimum or even fail to achieve a consistent state. Nevertheless the b_i estimates tend to become mostly localized and stable after just three to four rounds, and further computation generates little change. We therefore terminate the iterations after four rounds, $r = 4\,|C|$, and take each control point position as described in Equation (5).

3.3.4 Implementation. Because the optimization in Equation (4) appears to have a runtime complexity quadratic in the number of image pixels, we clarify here how it can be computed in linear time using the generalized distance transform on a log-space representation. Let $\rho_i(\mathbf{x}_i) \triangleq -\log b_i(\mathbf{x})$ with Z_i the proportionality constant ensuring $\sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) = 1$. First, we create α_{ij}, a translated version of the log-belief that aligns the expected location of the neighboring control point j with the current control point's putative location:

$$\tilde{\mathbf{x}}_{ij} \triangleq \mathbf{x}_j + \mathbf{t}_{ij} \tag{8}$$

$$\alpha_{ij}(\tilde{\mathbf{x}}_{ij}) \triangleq \rho_j(\mathbf{x}_j + \mathbf{t}_{ij}). \tag{9}$$

We then observe that the generalized distance transform [9] on α_{ij} efficiently calculates Γ_{ij}, the minimum combination of squared deviation and belief support for that deviation:

$$\Gamma_{ij}(\mathbf{x}_i) \triangleq \min_{\tilde{\mathbf{x}}_{ij}} \left\|\mathbf{x}_i - \tilde{\mathbf{x}}_{ij}\right\|^2 + \alpha_{ij}(\tilde{\mathbf{x}}_{ij}). \tag{10}$$

In this log-space representation, we then have that

$$\rho_i^{(t)}(\mathbf{x}_i) \leftarrow D(\mathbf{x}_i)^2 + \sum_{j:N_{ij}=1} \Gamma_{ij}(\mathbf{x}_i) + \log Z_i, \tag{11}$$

where D is the standard Euclidean distance transform [19] and each term may be calculated in time linear in the number of pixels.

Table 1: Information about map benchmark data [1].

Map Id.	Year	Atlas	Map Number	Map Region	Map Scale
a	1896	D0017	1592006	Central Calif.	443,529
b	1875	D0041	5370006	Florida	1,330,560
c			5370026	New Mexico	1,330,560
d	1866	D0042	1070001	Ohio	1,140,480
e			1070005	Wisconsin	1,140,480
f			1070006	Minnesota	1,140,480
g			1070007	Iowa	1,140,480
h			1070010	Arkansas	1,140,480
i			1070012	Mississippi	1,140,480
j	1879	D0089	5235001	Northern U.S.	7,500,000
k	1879	D0090	5242001	Missouri	1,966,700
l	1927	D0117	5755018	Indiana	1,100,000
m			5755024	Iowa	1,250,000
n			5755025	Nebraska	2,000,000
o			5755035	Wyoming	2,200,000
p			5755036	Montana	2,500,000
q	1924	D5005	5028052	N. Carolina	N/A
r			5028054	S. Carolina	N/A
s			5028100	N. Dakota	N/A
t			5028102	S. Dakota	N/A

3.3.5 Densification. Model alignment only fixes the 2D transformation $(x, y) \leftrightarrow (x', y')$ at the control points (where $\mathbf{v}_i \leftrightarrow \hat{\mathbf{x}}_i$). In order to fix the map coordinates of any locations that are not in the model, we must derive a dense 2D correspondence from the sparse control point data. We explore two methods:

Affine: Fit an affine transform to the control point matches using robust linear least squares.

Robust TPS: Robustly fit a thin plate spline to the control point matches, excluding outliers [24]. This allows local non-linear warping of the Cartesian grid.

The affine transform must optimize across all control points, whereas the thin plate spline only enforces local consistency and allows more variation between separated regions. While they differ in flexibility, both of these transforms imply some form of regularization and may not be able to fully express the actual relationship between GIS coordinates and the map's content.

4 EXPERIMENTS

This section presents the results of testing the methods just described on a collection of 20 historical maps. For comparability, we use the same set as in prior work [27]. The main results use toponym-based configurations on county or state boundaries. Additional experiments test variant models using roadway data and tree-based configurations.

4.1 Data

The U.S. state and county boundaries used for shape models (cf. Section 3.1) are taken from the U.S. Census Bureau Cartographic Boundary Files.[2] Highway paths are taken from the U.S. Geological Survey (USGS) National Transportation Dataset (NTD).[3]

As our benchmark, we take map images previously used for testing toponym text recognition [27], all drawn from the David Rumsey collection [2]. The 20 map images used in this benchmark data set range from years 1866–1927 and are drawn from seven atlases [1]; see Table 1. While the original ground truth annotation includes the bounding polygons and text transcription of toponyms, it does not include georeferencing information. For this work, we have added sets of ground truth correspondences between geographical coordinates (latitude and longitude) and pixel coordinates (row and column) in the map images.[4]

These correspondences arise from two different manual matching processes, one graphical and one textual. In the first case, human annotators matched the geographical and image coordinates of well-localized map locations—e.g., junctions, corners, or other topologically distinct points. We take the ground truth geographical coordinates from GIS data provided by the USGS; specifically state and county boundaries from the NBD, and road junctions from the NTD. In the second case, human annotators matched the labeled toponyms on the maps with entries from the USGS Geographic Names Information System (GNIS), subsequently allowing the system to put the image coordinates of the toponym's place marker (usually a city) in correspondence with the geographical coordinate reported by the GNIS. By combining correspondences from both methods, annotators produce a sufficient number of well-distributed points so that a parametric alignment could fit reasonably well over the entire map image, given a reasonable projection. The number of ground truth points for each map is given in Table 2.

4.2 Results

Qualitatively, alignments found by the proposed technique look more accurate than those of prior work. Figure 7 shows several typical examples for comparison. In each case the initial alignment is in the right neighborhood but includes a systematic error. The proposed method brings the model structures into closer juxtaposition with the corresponding map features.

The degree of improvement depends upon the densification method chosen. Although both the thin plate spline and affine densification methods work from the same control point fits, the former sometimes performs better than the latter. This is particularly true in local areas where a global, rigid affine fit cannot express the true local alignment, whether due to printing defects (cf. Figure 3) or a poor projection estimation. On the other hand, when the control point fit includes spurious matches, a global affine transform is more likely to successfully ignore the outliers than a local method like the thin plate spline, even with robust methods. (See discussion of Figure 8 below.)

Table 2 shows detailed results for each test map. The overall outcomes shown for the affine and thin plate spline densifications are significantly better than Weinman's toponym-based fit [27] for RMSE in both kilometers and pixels ($p < 0.011$).[5] The method included in this table was chosen for strong overall performance,

[2] https://www.census.gov
[3] https://www.sciencebase.gov/catalog/item/4f70b1f4e4b058caae3f8e16
[4] Data available at http://doi.org/11084/23330
[5] All significance tests use a paired, one-sided Wilcoxon signed-rank test.

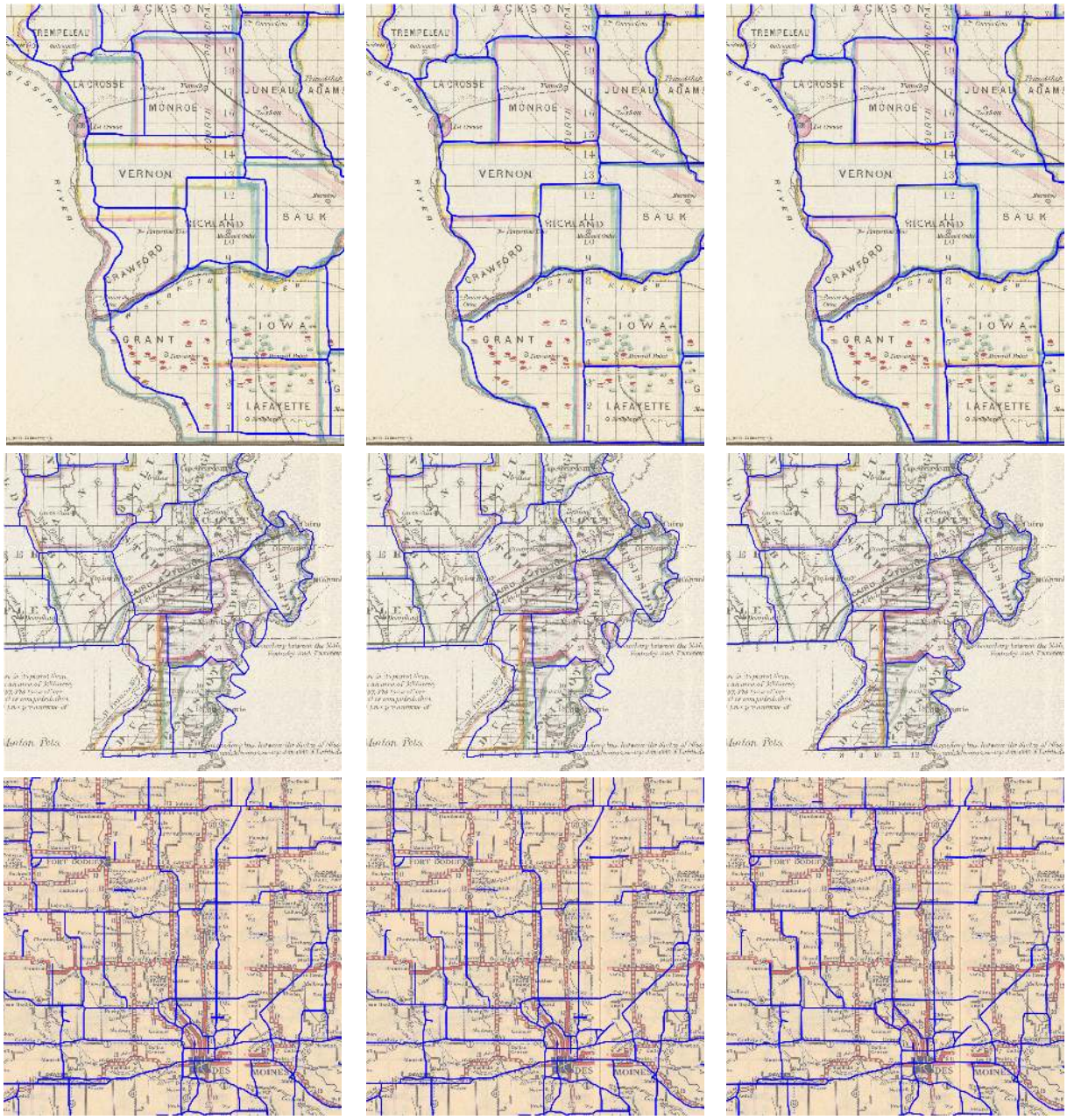

Figure 7: Sample results. Left: starting toponym alignment. Center: affine alignment. Right: thin plate spline alignment. (Maps top–bottom: D0042-1070005 and D0042-1070009 using state and county boundaries, D0117-5755024 using state boundary and major roadways.)

but is not necessarily best on each individual map. Overall results show Weinman's toponym-based adjustments [27] reduce RMSE (in pixels) by 41% over the prior technique [26], and the approach in this work nets a further 12% reduction.

4.2.1 Model Sensitivity. Where errors occur with both densification techniques, they can usually be traced to mismatch between model and image. Political boundaries, transportation networks, even rivers and shorelines can change over time. Modern data can easily include features not present in historical maps. Attempts to find a match that does not exist can lead to incorrect alignments as discussed below.

Figure 8 shows such a failure case for the thin plate spline. Maps from atlas D0117 do not include county boundaries, so the control points on a county boundary match a road to the south instead. The

Table 2: Alignment results of individual maps, grouped by atlas. GT is best (least-squares) affine fit to the ground truth correspondences after projection. SAC is the initial toponym-based MLESAC affine fit [26]. EM is the adjusted toponym-based affine fit [27]. Others are models described in this work with two densification (cf. Section 3.3.5) strategies and GIS layers indicated.

Map Id.	Num. Pts	RMSE (km) GT	SAC	EM	State,County Aff	State,County TPS	State Aff	State TPS	RMSE (px) GT	SAC	EM	State,County Aff	State,County TPS	State Aff	State TPS
a	43	1.61	3.78	2.69	*2.72	***2.28**	-	-	40.7	95.5	68.4	*66.5	***61.3**	-	-
b	58	10.40	27.41	12.51	11.61	11.27	11.22	**10.92**	84.2	206.1	100.3	94.6	93.3	**90.8**	91.3
c	46	5.78	41.85	12.56	11.74	11.29	7.79	**7.49**	47.7	297.0	99.1	89.8	81.8	**63.8**	64.8
d	222	4.02	6.96	4.72	4.42	**2.99**	6.83	3.82	45.0	79.4	52.6	49.0	**33.8**	83.9	48.7
e	134	2.50	10.59	5.18	4.47	**2.82**	5.18	3.49	25.4	106.6	50.4	38.7	**32.5**	56.5	36.2
f	141	8.55	20.56	9.57	7.55	7.76	**7.01**	7.09	85.0	238.1	97.1	74.8	74.5	**69.8**	94.2
g	176	3.71	10.31	5.13	3.81	**3.61**	4.07	3.69	36.5	101.8	49.7	37.2	**34.0**	39.7	36.8
h	122	3.41	6.66	4.11	3.44	**3.03**	3.72	4.08	33.4	66.9	40.1	33.5	**29.4**	36.3	38.3
i	127	4.05	6.91	4.50	**4.29**	4.30	4.70	4.44	41.1	69.3	45.5	43.5	**42.8**	47.6	45.5
j	71	10.39	39.83	35.78	**40.91	****35.57**	40.91	**35.57**	16.9	62.6	55.5	**63.6	****56.7**	63.6	**56.7**
k	260	6.70	9.31	7.48	6.71	6.95	**6.60**	6.82	39.0	54.5	44.6	40.0	41.9	**39.3**	40.3
l	620	3.04	3.61	3.14	3.13	3.25	**3.02**	3.52	45.3	54.5	46.8	46.1	50.5	**45.2**	48.8
m	782	3.18	3.34	**3.20**	3.55	4.66	3.31	3.66	37.8	39.7	**38.1**	41.7	53.1	39.0	41.8
n	597	5.83	7.11	6.26	7.16	6.63	6.14	**6.11**	48.1	59.7	52.0	64.1	51.4	51.1	**50.5**
o	306	3.12	8.30	4.41	6.98	5.27	**4.00**	4.27	27.0	71.2	38.1	60.7	43.1	**34.7**	37.6
p	483	6.88	7.18	7.31	7.99	8.37	**7.12**	7.34	48.0	50.5	51.5	52.8	59.8	**50.1**	51.0
q	384	2.27	2.97	2.51	2.40	2.41	2.41	**2.38**	26.1	33.6	28.8	**27.5**	27.7	27.7	27.7
r	251	2.08	2.77	2.30	2.48	**2.11**	4.24	2.20	26.0	34.2	28.7	35.5	**26.5**	68.2	27.4
s	388	2.15	3.74	2.98	2.37	**2.28**	2.69	2.57	25.1	43.4	34.9	27.7	**26.2**	31.5	31.7
t	194	2.51	5.49	5.45	2.68	**2.64**	3.03	2.97	27.4	61.1	58.0	29.0	**27.8**	32.7	32.1
Average		4.61	11.43	7.09	7.02	**6.47**	-	-	40.28	91.29	54.01	50.82	**47.4**	-	-

*Metadata indicates sub-state regional map; model uses only GIS county boundaries
**Metadata indicates multi-state map; model uses only GIS state boundaries

thin plate spline adjusts for this displacement, warping the straight county lines into a southward curve. In this case, the global nature of the affine fit helps it to avoid such local errors. However, it would be better to avoid the spurious matches entirely.

Even if the fit is not misled by spurious control point matches, they are certainly not likely to improve the result. Wherever possible, human supervision or map metadata should be used to ensure that models match the image contents. To demonstrate this concretely, we built a set of alternate models using state boundaries plus major roads (state highways and interstates from the NTD) instead of county boundaries and fit them to the D0117 road atlas maps. Although the map's road network exhibits discrepancies compared to the modern data, the resulting thin plate spline fits shown in Table 3 nevertheless improve significantly ($p < 0.035$) on the match for the county-boundary models used in Table 2, reducing RMSE (in kilometers) by 27% to a rate even lower than the best rigid affine fit to the ground truth points.

Figure 9 highlights another instance where attempting to match features that are not present in the image contributes to a poor alignment result. In this case the control points along the state boundary largely match to the correct location. Unfortunately, the modern county boundaries just below are not present in the map, and instead generate spurious correspondences along a river. These misidentifications compound problems with the choice of projection, causing the algorithm to place the boundary in the wrong location. Using a model with only state boundaries can reduce such errors, as shown in Table 2(f).

Table 3: Alignment results using state boundary and roads. GT is best (least-squares) affine fit to the ground truth correspondences after projection. EM is the adjusted toponym-based affine fit [27]. Others are the model described in this work with two densification methods (cf. Section 3.3.5) and GIS layers indicated.

Map Id.	RMSE (km) GT	EM	State,County Aff	State,County TPS	State Aff	State TPS	State,Roads Aff	State,Roads TPS
l	3.04	3.14	3.13	3.25	3.02	3.52	3.00	**2.25**
m	3.18	3.20	3.55	4.66	3.31	3.66	3.51	**2.57**
n	5.83	6.26	7.16	6.63	6.14	6.11	8.10	**5.02**
o	3.12	4.41	6.98	5.27	4.00	4.27	4.09	**3.99**
p	6.88	7.31	7.99	8.37	7.12	7.34	7.34	**7.00**
Avg.	4.41	4.86	5.76	5.64	4.72	4.98	5.21	**4.17**

4.2.2 Tree-based Configurations. While the previously cited results all use the toponym-based configurations as their starting point, Table 4 shows the performance of new models built from raw latitude/longitude data using $m = 5$ tree fits, as described in Section 3.2.2. Because the technique relies on a crude heuristic to

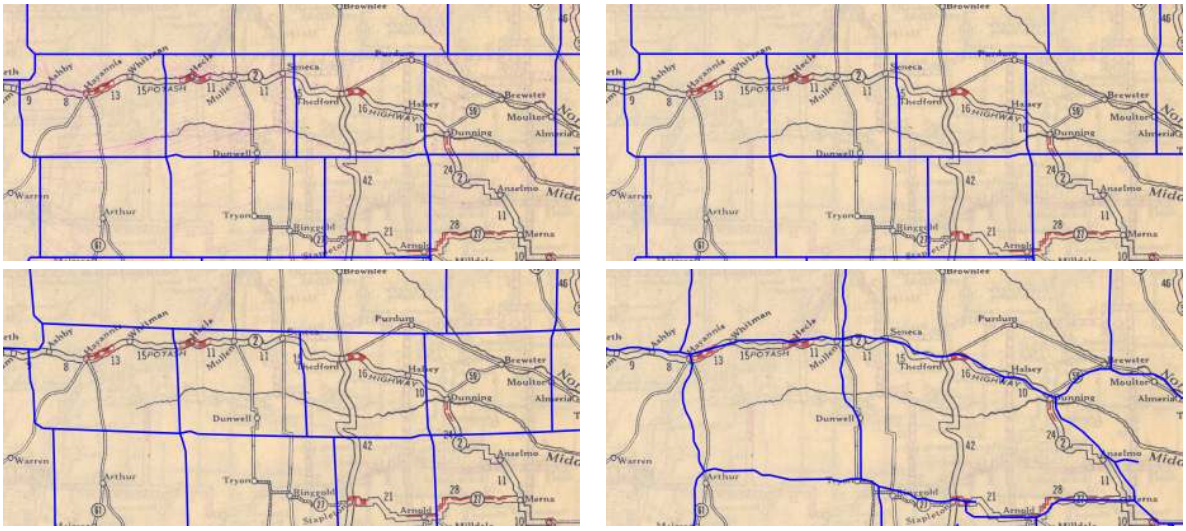

Figure 8: Importance of the right model. Top left: starting toponym alignment. Top right: affine result. Bottom left: thin plate spline result. Bottom right: thin plate spline result using a road model. (D0117-5755025 [2], original map copyright ©1927 Rand McNally)

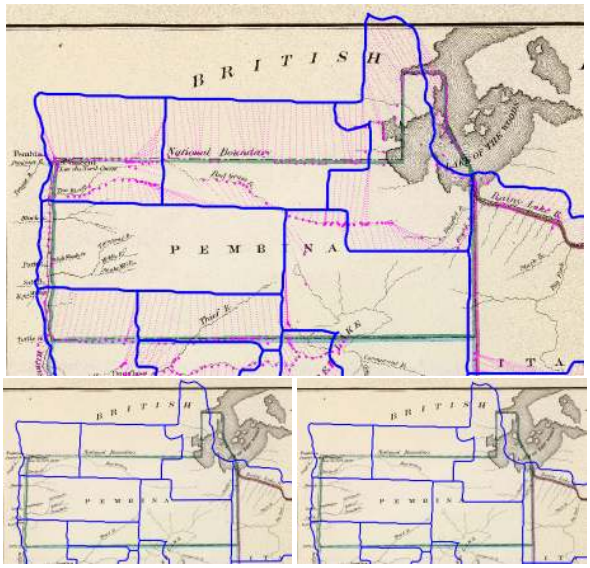

Figure 9: A failure case. Top: starting toponym alignment with control point fits. Bottom left: affine result. Bottom right: thin plate spline result. (D0042-1070006 [2])

scale the model, it fails on about half the attempts. Nevertheless on the remainder it produces reasonable results, and for certain maps (e.g., D0090-5242001) it achieves the best results of any method, as illustrated in Figure 10. In this map, the toponym configuration contains a significant alignment error due to a dearth of toponyms in Michigan's upper peninsula, leading the resulting models astray. The tree-based configuration is unaffected and matches well using the shape alone.

Table 4: Alignment results comparing model configurations: Toponym-based (cf. Section 3.2.1) versus Tree-based (cf. Section 3.2.2) with state and county boundaries as GIS layers and two densification methods (cf. Section 3.3.5). GT is best (least-squares) affine fit to the ground truth correspondences after projection. SAC is the initial toponym-based MLESAC affine fit [26]. EM is the adjusted toponym-based affine fit [27]. Results shown only for tree fits that perform no worse than SAC.

	RMSE (km)						
Map Id.	GT	SAC	EM	Toponym		Tree	
				Aff	TPS	Aff	TPS
b	10.40	27.41	12.51	11.61	**11.27**	15.41	12.95
c	5.78	41.85	12.56	11.74	**11.29**	24.00	20.87
d	4.02	6.96	4.72	4.42	**2.99**	3.76	3.07
e	2.50	10.59	5.18	4.47	**2.82**	3.29	2.87
g	3.71	10.31	5.13	3.81	**3.61**	4.79	3.96
h	3.41	6.66	4.11	3.44	**3.03**	5.66	5.70
i	4.05	6.91	4.50	**4.29**	4.30	6.66	6.61
k	6.70	9.31	7.48	6.71	6.95	**6.70**	6.83
q	2.27	2.97	2.51	**2.40**	2.41	2.72	2.44
r	2.08	2.77	2.30	2.48	**2.11**	2.37	2.29
s	2.15	3.74	2.98	2.37	**2.28**	3.34	2.51
t	2.51	5.49	5.45	2.68	**2.64**	4.48	3.28

5 CONCLUSION

The methods proposed herein significantly improve the accuracy of map georeferencing as compared to prior work. They are most effective when the models used correspond to the actual map contents. Future work should look at techniques for post-alignment verification so that bad matches can be identified and improved.

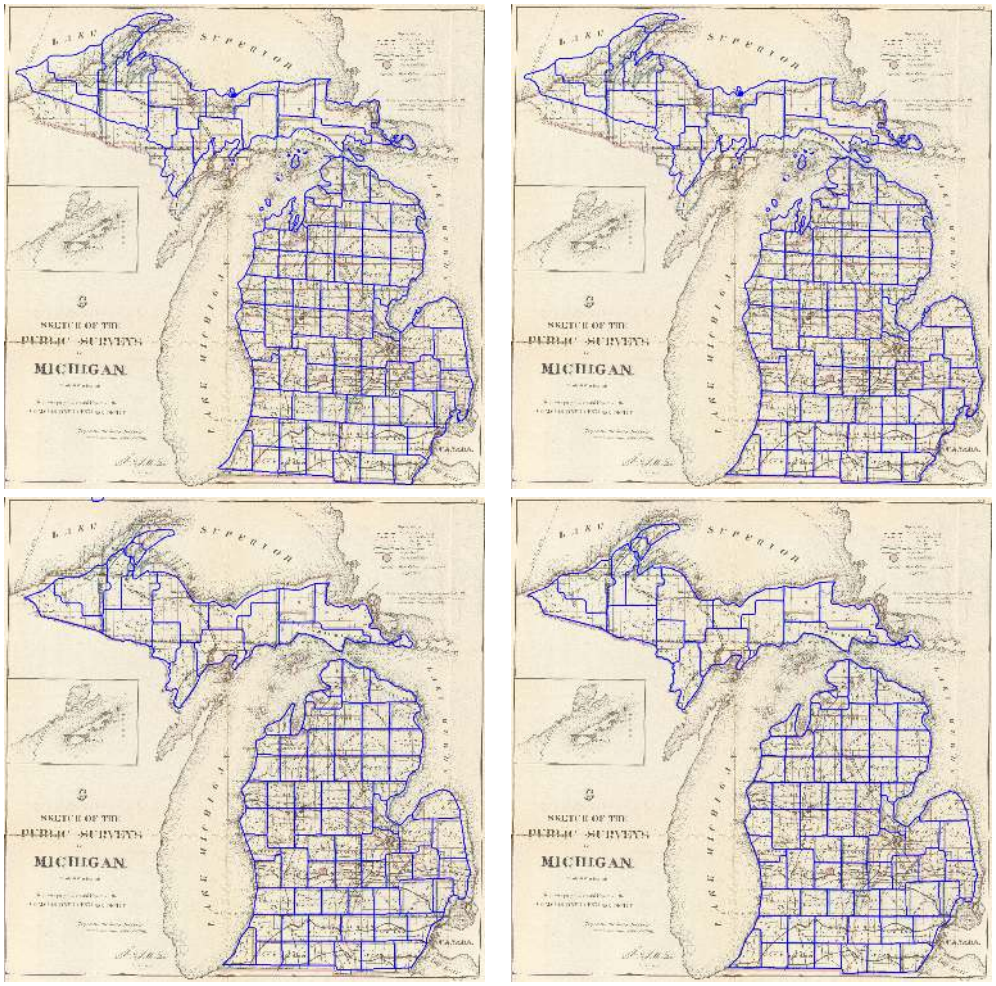

Figure 10: Effects of model configuration. Top row, l-r: toponym-based configuration, thin plate spline result. Bottom row, l-r: tree-based configuration, thin plate spline result. (D0042-1070004 [2])

Depending on the resources available, the core method proposed here could be applied in a number of different scenarios. When initialized via toponym matching or the automatic tree-based approach described above, it could form part of an automated pipeline for processing a large map collection [23]. Alternately, for human-in-the-loop configurations, it could provide a "snap to borders" functionality once the user provides a crude initialization.

The proposed method can be further refined in a number of ways. Binarization should not be the only approach to feature detection, since some maps use color changes instead of lines to represent borders. Furthermore, borders and roads are not the only geographic feature that can be easily matched. Railroads, rivers, shorelines, and other curvilinear structures could be added to the model. Even toponym features can be incorporated. On the other hand, not all features are present in or appropriate for all maps, and automatically determining which should be used in any individual case remains an open problem. In summation, this paper provides a useful tool for aligning historic maps to modern geographic resources, but more work remains to be done.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1526350.

REFERENCES

[1] Larry Boateng Asante, David Cambronero Sanchez, Ravi Chande, Dylan Gumm, and Jerod Weinman. 2017. *A Data Set of Annotated Historical Maps.* Technical Report. Grinnell College. https://doi.org/11084/19349

[2] Cartography Associates. [n. d.]. David Rumsey Map Collection. ([n. d.]). http://www.davidrumsey.com

[3] Ching-Chien Chen, Craig A. Knoblock, and Cyrus Shahabi. 2008. Automatically and accurately conflating raster maps with orthoimagery. *GeoInformatica* 12, 3 (2008), 377–410.

[4] Yao-Yi Chiang, Craig A. Knoblock, Cyrus Shahabi, and Ching-Chien Chen. 2009. Automatic and accurate extraction of road intersections from raster maps. *GeoInformatica* 13, 2 (2009), 121–157.

[5] Yao-Yi Chiang, Stefan Leyk, and Craig A. Knoblock. 2014. A Survey of Digital Map Processing Techniques. *ACM Comput. Surv.* 47, 1 (May 2014), 1:1–1:44.

[6] Weiwei Duan, Yao-Yi Chiang, Craig A. Knoblock, Vinil Jain, Dan Feldman, Johannes H. Uhl, and Stefan Leyk. 2017. Automatic Alignment of Geographic Features in Contemporary Vector Data and Historical Maps. In *Proc. Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery (GeoAI '17)*. 45–54.

[7] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. 2010. Object Detection with Discriminatively Trained Part Based Models. *IEEE Trans. PAMI* 32, 9 (2010), 1627–1645.

[8] Pedro Felzenszwalb and Daniel Huttenlocher. 2005. Pictorial Structures for Object Recognition. *Intl. J. Comp. Vis.* 61, 1 (2005), 55–79.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2012. Distance Transforms of Sampled Functions. *Theory of Computing* 8, 1 (2012), 415–428.

[10] Martin Fischler and Robert A. Elschlager. 1973. The Representation and Matching of Pictorial Structures. *IEEE Trans. on Computers* 22, 1 (1973), 67–92.

[11] Christopher Fleet, Kimberly Kowal, and Petr Pridal. 2012. Georeferencer: Crowdsourced georeferencing for map library collections. *D-Lib Mag.* 18, 11 (2012).

[12] Zicheng Guo and Richard Hall. 1989. Parallel thinning with two-subiteration algorithms. *Comm. of the ACM* 32, 3 (1989), 359–373.

[13] Andreas Hackeloeer, Klaas Klasing, Jukka M. Krisp, and Liqiu Meng. 2014. Georeferencing: a review of methods and applications. *Annals of GIS* 20, 1 (2014), 61–69.

[14] Nicholas Howe. 2013. Document Binarization with Automatic Parameter Tuning. *Intl. J. Doc. Analysis & Recog.* 13, 3 (September 2013), 247–258. DOI: 10.1007/s10032-012-0192-x.

[15] Nicholas Howe. 2013. Part-Structured Inkball Models for One-Shot Handwritten Word Spotting. In *Proc. ICDAR.*

[16] Nicholas Howe and Ji Won Chung. 2019. Symmetric Inkball Alignment with Loopy Belief Propagation. In *Proc. ICDAR.* 349–354. https://doi.org/10.1109/ICDAR.2019.00063

[17] Arthur Jay Klinghoffer. 2006. *The power of projections: how maps reflect global politics and history.* Praeger, Westport, Conn.

[18] Yan Li and Ronald Briggs. 2009. *An automated system for image-to-vector georeferencing.* Ph.D. Dissertation. University of Texas, Dallas, TX.

[19] Calvin R. Maurer, Rensheng Qi, and Vijay Raghavan. 2003. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. PAMI* 25, 2 (2003), 265–270.

[20] David Rumsey and Meredith Williams. 2002. Historical Maps in GIS. In *Past Time, Past Place: GIS for History,* Anne Kelley Knowles (Ed.). ESRI Press, Redlands, California, Chapter 1, 1–18.

[21] Martin Simonovsky, Benjamín Gutiérrez-Becker, Diana Mateus, Nassir Navab, and Nikos Komodakis. 2016. A deep metric for multimodal registration. In *Proc. MICCAI.* 10–18.

[22] John P. Snyder. 1987. *Map Projections: A Working Manual.* Technical Report Professional Paper 1395. U.S. Geological Survey, Washington, D.C.

[23] Sasan Tavakkol, Yao-Yi Chiang, Tim Waters, Feng Han, Kisalaya Prasad, and Raimondas Kiveris. 2019. Kartta Labs: Unrendering Historical Maps. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI 2019).* 48–51.

[24] Ruwan B. Tennakoon, Alireza Bab-Hadiashar, David Suter, and Zhenwei Cao. 2013. Robust data modelling using thin plate splines. In *Proc. Intl. Conf. DICTA.*

[25] P. H. S. Torr and A. Zisserman. 2000. MLESAC: a new robust estimator with application to estimating image geometry. *CVIU* 78, 1 (April 2000), 138–156.

[26] Jerod Weinman. 2013. Toponym Recognition in Historical Maps by Gazetteer Alignment. In *Proc. ICDAR.* 1044–1048.

[27] Jerod Weinman. 2017. Geographic and Style Models for Historical Map Alignment and Toponym Recognition. In *Proc. ICDAR.* 957–964.

[28] I. Yilmaz and M. Gullu. 2012. Georeferencing of historical maps using back propagation artificial neural network. *Experimental Techniques* 36, 5 (2012), 15–19.