Geographic and Style Models for Historical Map Alignment and Toponym Recognition

Jerod Weinman
Grinnell College, Grinnell, Iowa, USA
jerod@acm.org

***Abstract*—Recognizing the place names within textual labels on historical maps is complicated by many factors, such as curvilinear baselines and dense overlap with other textual or graphical elements. However, maps' alignment with known geography and inter-label typographic style consistencies provide strong cues for resolving uncertainty and reducing text recognition errors. We present a unified probabilistic model to leverage the mutual information between text labels and styles and their geographical locations and categories. This work also introduces likelihood functions to model label placement for polyline and polygon geographical features, such as rivers or provinces. We evaluate the methods on 30 maps from 1866–1927. By interleaving automated map georeferencing with text recognition, we reduce word recognition error by 36% over OCR alone. Incorporating category-style links reduces toponym matching error by 32%.**

***Index Terms*—Geographic information systems (GIS), georeferencing, map processing, toponym recognition**

I. Introduction

Densely layered with information, maps convey geography, history, and power. The dually textual and graphical nature of maps continues to challenge automated processes. Yet as libraries increasingly offer more digitized historical artifacts online, the need and opportunities for geospatial and textual search in pre-digital era maps continues to grow. While automated map processing has a long history in the document recognition community, scholars from many backgrounds increasingly appreciate opportunities afforded by such resources.

This paper introduces several new models to invert cartography by extracting information from scanned historical maps. In particular, we focus on processing cropped words, leaving text/graphics separation as preprocessing. More than a classification problem, we characterize it as a data linking task between a geographic information system (GIS) and the image. In an iterative process, text in labels is recognized by matching to features known to be at the geographical coordinates depicted on the image, an alignment itself provided by the hypothesized label text. To support our highly configurable model, the inference process incrementally adds GIS data and relaxes some constraints while adding others.

Our contributions are fourfold. First, we introduce a fine-grained model incorporating many possible categories of geographical features with learned map-specific biases. To support these, we derive efficient models for inferring the label placement of non-point features, both polyline and polygon (e.g., rivers and provinces). Finally, we learn map-specific correlations between each category and a latent representation of the font style in the corresponding label (e.g., river labels are italic). These are unified in a probabilistic model using unsupervised parameter adaptation.

The next section reviews geospatial concepts and related work. We formalize the model in Section III and describe parameter learning in Section IV. Section V provides an experimental evaluation using 30 historical North American maps from 1866–1927. We demonstrate a 20% reduction in word recognition error over a prior geography-based approach [1]—a 36% improvement over a robust OCR alone.

II. Background

Automated digital map processing has a long history; see Chiang *et al.* [2] for a complete survey. Here we review geospatial concepts and then assess closely related work before summarizing our GIS and historical map data.

A. Geographic Coordinate Systems and Projections

Modern geographic information systems rely on several model layers to construct a *geographic coordinate system* (GCS). First, geographers model the globe as an *ellipsoid* to capture the general shape of the earth. Then, a *datum* positions the ellipsoid relative to Earth's surface. Finally, the *GCS* adds a prime meridian, or origin. Cartographers then use some function to project GCS points on the ellipsoid to a map plane. *Georeferencing* inverts the process by mapping image pixels to geographical coordinates.

Because our text-based georeferencing does not require great precision and historical maps predate modern standards, we use a simple spherical model of the earth with GIS data based on the GCS North American 1983 standard.

Of the several cartographic projection families, we consider only two: cylindrical and conic [3]. Others (i.e., azimuthal and pseudocylindrical), are primarily designed for world-scale maps or used more often to cover southern-hemisphere continents. From these two families, we specifically allow the projection to be either an equal area cylindrical or the Hassler/American polyconic projection. The cylindrical projection requires only a central meridian, while the polyconic requires a central meridian and a parallel along which the cone is tangent. We explain how to estimate these parameters for each map in Section III-C1. While some of the historical maps in our evaluation do not strongly exhibit properties of either projection family, other maps clearly require the use of a particular family for reasonable georeferencing.

B. Related Work

Many works separate map text from graphics to enhance recognition, but fewer integrate top-down information available for map text recognition, as in this work. Yu *et al.* [4] correct recognition errors with a geographic dictionary by integrating OCR outputs from overlapping, georeferenced maps. Tarafdar *et al.* [5] recover from character detection failures by using partial dictionary matches to guide wordspotting search.

Rusiñol *et al.* [6] automatically georeference map images by recognizing coordinates printed along the borders, whereas Weinman [1] and Pawlikowski and Ociepa [7] use the placement of recognized toponyms within the map image, an approach we improve and integrate through iteration.

We present label placement models for polyline and polygon features similar to those described in general terms by Gelbukh *et al.* [8]. Their method, presented without empirical validation, requires a previously georeferenced map image and offers GIS-based post-processing for OCR error correction. By contrast, we address computational feasibility and our automated method fully integrates these processes.

C. Evaluation and GIS Data

1) Maps: To evaluate this work, we extend Weinman's annotations [1] to 30 historical maps spanning years 1866–1927 from the David Rumsey collection of North America,[1] identifying the map region (a list of states or counties), base polyline and height of individual words along with text transcriptions, grouping words into a label for a geographical feature, and assigning labels to a window when the image has more than one coordinate system.[2] The 12,578 words in 9,555 labels range from 8–223 pixels high (median 22px). A label also has image coordinates when associated with a point on the image (e.g., a city marker).

2) Gazetteer and GIS: We link the words in the map images to place names (official or historical) and coordinates for geographical features as given by the US Geographic Names Information System (GNIS).[3] Each GNIS entity has a unique feature identifier and is assigned to one of 65 classes (e.g., *Arch*, *Bay*, *Cape*, *Dam*, etc.); Section III-B1 describes the GNIS classes we use. GIS data linked to the GNIS provides polylines and polygons used by our model (Sections III-C3 and III-C4): the US Geological Survey's National Hydrography Dataset for *Bay, Canal*, *Lake*, *Reservoir, Stream*, and *Swamp* classes and the US Census Bureau's cartographic boundary files for *Island*s and counties.[4]

III. Probabilistic Model

Fig. 1 shows the unifying probabilistic model for recognizing the word text string $\mathbf{y}$ and corresponding geographical feature f from each cropped word image $\mathbf{W}$ in a map. Let $\mathbf{\Theta} = \langle \mathbf{\Omega}, k, \phi, \mathbf{\Gamma} \rangle$ represent all the (observed) hyperparameters and $\mathbf{\Upsilon} = \langle T, \sigma, \mathbf{A}, p, \gamma, \mathbf{\Lambda} \rangle$ represent all the other parameters that must be learned for each map. The remainder of this section details each conditional probability necessary to represent the joint probability of all values in the Bayesian network.

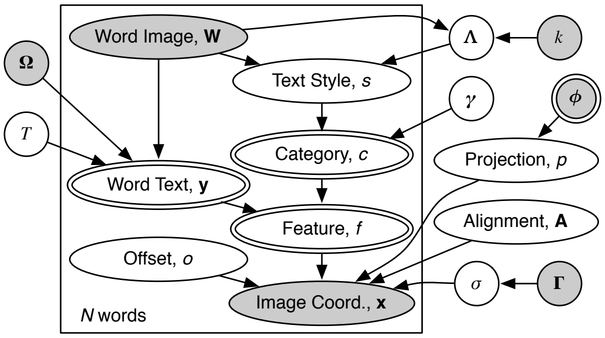

Figure 1. Bayesian graphical model for map alignment and text recognition. Observed values are shaded while those in the plate are replicated for each word. Double-lined values also depend on a gazetteer, omitted for clarity.

A. Word Recognition

To calculate the conditional probability of candidate character strings given the word image, $P(\mathbf{y} \mid \mathbf{W})$, we may use any system providing confidence scores for several alternative string hypotheses (in or out of lexicon) and robustness to the noisy, degraded backgrounds prominent in maps.

We adapt a semi-Markov CRF text recognizer [9] by replacing the character recognition module with a fully convolutional neural network (CNN)—inspired by Jaderberg *et al.* [10]—that takes a 32×32 grayscale image processed by five convolutional layers and two fully-connected layers. The Caffe-based CNN is trained on distorted, synthetically generated text using stochastic gradient descent with AdaGrad. We then tune the relative weights of the CRF's modules on held out map data.

To promote sub-optimal strings using geographical information *a posteriori*, a Viterbi-like parser with beam search [9] caches the CRF's top-scoring parse alternatives in a bounded priority queue. Given word image $\mathbf{W}$ and learned parser/CNN parameters $\mathbf{\Omega}$, we calculate the discrete, word-based probability with a temperature-adjusted softmax over the cached strings

$$P(\mathbf{y} \mid T, \mathbf{W}, \mathbf{\Omega}) \propto \exp\left(-\tfrac{1}{T} E(\mathbf{y}, \mathbf{W}, \mathbf{\Omega})\right), \qquad (1)$$

where $E(\mathbf{y}, \mathbf{W}, \mathbf{\Omega})$ is the cached score for string $\mathbf{y}$ and $T > 0$ is a free parameter, which adjusts the prior's confidence by moderating the energy before exponentiation.

We convert each normalized, cropped word image to grayscale by extracting its primary PCA component from the RGB channels with no text separation or other preprocessing. Initial word scores are calculated with a large general English dictionary and an additional region-specific list of toponyms.

B. Feature Category and Style Models

Maps serve different purposes by limiting the geographical entities represented. Morever, the distribution of geographical features varies by region. Such map-specific distributions of feature categories can help resolve ambiguities or correct errors. For example, when confidently recognized map elements

[1] http://davidrumsey.com

[2] Data available at http://hdl.handle.net/11084/19349

[3] https://geonames.usgs.gov; worldwide data at http://www.geonames.org

[4] https://nhd.usgs.gov and https://www.census.gov/geo

are mostly either cities or lakes, other elements are likely to belong to these same categories. We therefore learn map-specific prior probabilities over these categories.

While cartographers use few established standards to render geographic categories in explicit font styles, different categories often have different styles. For example, river and lake labels might be rendered in *Italics*, while county names may appear in ALL CAPS; state or region labels may be larger than those for cities and towns when font size is proportional to an item's level in a geographic hierarchy. We leverage such regularities to refine our model and enhance recognition.

1) Feature Categories: Weinman [1] considers only features from the states, counties, and cities that belong to the GNIS classes *Civil* and *Populated Place*, and subsequently models each map word as either an outlier or inlier corresponding to a feature in agreement with the alignment. These restrictions provide an important regularization for the early stages of model fitting but must eventually be relaxed to encompass a greater range of map features. Thus, rather than dichotomize, our model learns biases among a broader set of geographical classes by conditioning the feature prediction on a specific category.

Our model's final set of categories $\mathcal{C}$, and the GNIS feature classes comprising them, are as follows.

Waterbody *Lake*, *Reservoir,* and *Swamp* entities

Flowline *Stream* and *Canal* entities

Post Office *Post Office* entities marked "historical"

Park *Park* entities with names containing "State Park" or "National Park"

County *Civil* entities with names ending in "County", "Parish", or "Borough"

State *Civil* entities with names beginning "State" or "Commonwealth"

Place All remaining *Civil* and *Populated Place* entities

Finally, *Bay*, *Cape*, *Island*, *Military*, *Range*, and *Summit* entities are each given their own category of the same name.

In addition, to reduce diffusion of credit assignment in the probability model, we consolidate several closely related GNIS features by merging historical *Post Office* entities, townships (which are *Civil* entities), and *Populated Place* entities with the same names as cities, towns, or villages (also *Civil* entities) whose primary locations are within one kilometer of each other (about 10 pixels in our annotated map images).

The feature probability is a category-restricted count ratio,

$$P\left(f \mid \mathbf{y}, c\right) \triangleq \frac{M_c\left(f, \mathbf{y}\right)}{\sum_{f' \in F(c)} M_c\left(f', \mathbf{y}\right)}, \tag{2}$$

where $F\left(c\right)$ is the (mutually disjoint) set of features from category $c \in \mathcal{C}$ and $M_c\left(f, \mathbf{y}\right) \in \{0, 1\}$ indicates a case-insensitive and abbreviation-expanded match for word candidate $\mathbf{y}$ in the name of some feature $f \in F\left(c\right)$. We introduce the special feature $f_\emptyset$ to represent an unknown feature or a label for a non-geographic word on the map, which belongs to category $c_\emptyset$, the non-geographic outliers, forcing matches $M_{c_\emptyset}\left(f_\emptyset, \mathbf{y}\right) = 1$ for all strings $\mathbf{y}$ with $F\left(c_\emptyset\right) = \{f_\emptyset\}$ so that $P\left(f_\emptyset \mid \mathbf{y}, c_\emptyset\right) = 1$.

2) Category Styles: We learn a multinomial model over categories, conditioned on a discrete latent style $s \in [1, \ldots, k]$ for each cropped word

$$P\left(c \mid s, \boldsymbol{\gamma}\right) \triangleq \gamma_{cs}. \tag{3}$$

To model the text style of words, we adapt a local binary pattern (LBP) variant developed for writer identification by Nicolaou *et al.* [11]. Their descriptor eliminates rotational invariance, compactly capturing the frequency of stroke width and orientations by treating each radius independently.

LBP variants have also been used for script identification (e.g., Arabic versus Cyrillic) with supervised learning [12]. While both the writer and script identification tasks can leverage significantly more data (multiple text lines), toponym labels consist of one or just a few words. Moreover, text styles and their associations with particular categories vary among maps. Therefore we need unsupervised, map-specific learning of category-style associations. Toward this end, we utilize latent dirichlet allocation (LDA) [13] to learn mixtures of distributions of LBP "words" belonging to different "topics" (text styles). Each map serves as its own "corpus" of documents, which are the individual cropped (word) images to be recognized. To predict the style s of each cropped word image $\mathbf{W}$, we calculate its posterior distribution over k topics, denoted $P\left(s \mid \mathbf{W}, \boldsymbol{\Lambda}, k\right)$, where $\boldsymbol{\Lambda}$ are the parameters from LDA inference on the map (see Section IV-B7). Section IV-B gives expectation maximization updates for learning $\boldsymbol{\gamma}$.

We use $k = 16$ styles, validated on the held-out map training data, which allows for many combinations of typefaces, weights, and font sizes in a map. Prior to calculating LBP features, we geometrically normalize (i.e., horizontally straighten) the word images so they have consistent feature orientations but otherwise preserve scale, a meaningful text property. Following Nicolaou *et al.* [11], we drop the "all zero" LBP feature—akin to a "stop word" of text modeling—capturing only all-background patterns.

C. Geographical Location Models

A text label is likely to be offset from the corresponding geographical feature's image location, as predicted by the gazetteer's coordinates, map projection, and image alignment. While small-scale features are reasonably modeled as *points*, *polygon* areas (e.g., provinces) demand a different treatment. Moreover, *polyline* features such as rivers could have labels placed anywhere along them. Here we detail models that capture the text placement for these three feature types.

To support these models, we hypothesize a sparse grid of many potential feature "offsets" o around and along the localized word image (as shown in Fig. 2), with a uniform prior $P\left(o\right) \propto 1$. We denote the predicted point coordinate $\mathbf{x}_o$, where the coordinates are observed but the offset is latent.

1) Projection and Alignment: The GIS geographic coordinates must be projected to Cartesian map space before being transformed to image coordinates. Because the cartographer's projection is unknown, we use a 2D affine transform $\mathbf{A} \in \mathbb{R}^{2\times3}$, providing smaller residuals than a similarity transform.

The projection family prior $P(p)$ is uniform. The map region—given as meta-data (e.g., "US South", "Ohio", or "Denver")—controls projection parameters: the central meridian and parallel ϕ are the center of the graticular area covered by the region's GNIS *Civil* and *Populated Place* features.

Let $\overline{\mathbf{C}}_p(f) \in \mathbb{R}^{3\times S}$ represent the $S \geq 1$ projected Cartesian map coordinate(s) for feature f under projection p, with each coordinate augmented by a 1; the predicted image coordinate(s) are then given by $\mathbf{A}\overline{\mathbf{C}}_p(f)$.

All three label placement models—point, polyline, and polygon—share a scale parameter σ, which is estimated for each map separately. We give σ^2 an inverse gamma distribution, conjugate with the Gaussian used for point features. The hyperparameter $\boldsymbol{\Gamma}$ is a vector of residual errors observed on manually georeferenced training maps.

2) Point Feature Model: Most categories' features appear as points on small-scale maps; therefore an isotropic Gaussian models their map placements, offset from the text label,

$$P(\mathbf{x} \mid o, f, \mathbf{A}, p, \sigma) \triangleq \mathcal{N}\left(\mathbf{x}_o \mid \mathbf{A}\overline{\mathbf{C}}_p(f), \sigma^2\right). \tag{4}$$

Outliers could be placed anywhere, so their likelihood is uniform over the image domain,

$$P(\mathbf{x} \mid o, f_\emptyset, \mathbf{A}, p, \sigma) \triangleq (\text{Image Area})^{-1}. \tag{5}$$

Both (4) and (5) are used by Weinman [1].

3) Polyline Feature Model: Labels could be placed anywhere along linear features, such as rivers, so their likelihood functions must capture greater spatial variation than the Gaussian point model (4) by representing an observation's deviation from a central curve rather than a single predicted point. In the following, we review a Gaussian analog for polyline features, demonstrating a computational burden too great for this application, even when discretely approximated by convolution. We then provide a computationally simpler alternative using the distance transform.

We wish to calculate the likelihood of an observed point $\mathbf{x}$ deviating from a uniform prior along a line segment L:

$$P(\mathbf{x} \mid L) = \begin{cases} \frac{1}{|L|} & \mathbf{x} \in L \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $|L|$ is the segment length. We generalize from a single point by using an isotropic Gaussian to model deviation from the uniformly likely points of L. Like a Parzen window density, which places a Gaussian kernel at each of a set of observed samples and averages the likelihoods, we center a Gaussian along each point of the line and average by integrating over the line,

$$P(\mathbf{x} \mid L, \sigma) = \frac{1}{|L|}\int_L \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma)\, \mathrm{d}s = F_\sigma(\mathbf{x}, L), \tag{7}$$

which is the convolution of the Gaussian and uniform distributions. Jin and Tai [14] show how to calculate F_σ "analytically" (using the Gaussian error function, erf). We generalize to the polyline case as

$$P(\mathbf{x} \mid \mathcal{L}, \sigma) = \frac{1}{|\mathcal{L}|}\sum_{i=1}^{S} |L_i|\, F_\sigma(\mathbf{x}, L_i), \tag{8}$$

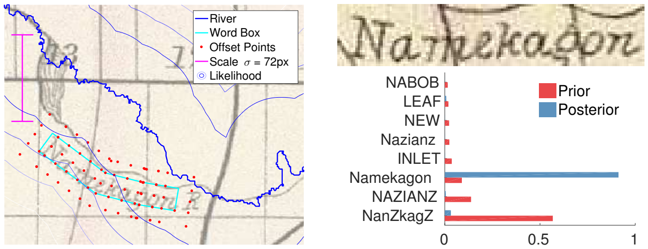

Figure 2. River name recognition: contours of polyline likelihood $P\left(\mathbf{x} \mid f, \hat{p}, \hat{\mathbf{A}}, \hat{\sigma}\right)$ for f=1579990 (GNIS ID for "Namekagon River") and comparison of prior $P(\mathbf{y} \mid \mathbf{W}, \boldsymbol{\Theta})$ and posterior $P\left(\mathbf{y} \mid \mathbf{x}, \mathbf{W}, \hat{\boldsymbol{\Upsilon}}, \boldsymbol{\Theta}\right)$.

where $\mathcal{L}$ has line segments L_i $1 \leq i < S$, and $|\mathcal{L}|$ is the polyline's total length. While this approach is exact, its computational cost is significant: each query point $\mathbf{x}$ (and there are several) from each of N cropped words must be evaluated for each of S segments along polylines for all F linear features in the GIS—$O(NFS)$ with high constant factors.

Using the convolutional integral (7) for generating 3D surfaces, Bloomenthal and Shoemake [15] advocated pixelizing the polyline to perform image convolution. Convolving once for each polyline and looking up probabilities of points from each of the N cropped words has complexity $O(N + FDK)$ where D is the number of pixels over which the convolution must be computed and K is the (separable) kernel size. Unfortunately, the scales ($\hat{\sigma} \approx 70$ px) learned during alignment (Section IV-B3) require impractically large kernels. We therefore propose a more efficient alternative that also has preferable properties as a density function for this application.

For a likelihood uniform along a polyline but otherwise decreasing with the distance from the polyline, we begin with an (un-normalized) Gaussian distance kernel

$$K_\sigma(r) \triangleq \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{9}$$

and define the placement likelihood for polyline features as

$$P(\mathbf{x} \mid o, f, \mathbf{A}, p, \sigma) \propto K_\sigma(D_{\mathcal{L}}(\mathbf{x}_o)), \tag{10}$$

where $D_{\mathcal{L}}(\mathbf{x})$ gives the minimal distance from $\mathbf{x}$ to the feature's projected, aligned polyline $\mathcal{L} = \mathbf{A}\overline{\mathbf{C}}_p(f)$; for $\mathbf{x} \in \mathcal{L}$, the distance $D_{\mathcal{L}}(\mathbf{x}) = 0$ so that the likelihood is maximal and uniform along the polyline, as shown in Fig. 2. The example demonstrates robustness to significant change in hydrography; the flowline depicted in the 1866 map deviates from the present path, yet the likelihood for the true feature is much larger than an outlier, correcting the word posterior.

Analytically calculating the point-to-polyline distance $D_{\mathcal{L}}(\mathbf{x})$ has the same computational burden as the Gaussian model (8). Fortunately, an $O(D)$ distance transform algorithm [16] calculates the unnormalized likelihood over a discrete image domain of size D, which we then normalize by summing the discrete values. Bypassing the inefficiency of large kernels, our $O(N + FD)$ distance-based likelihood (10) requires about 200× fewer operations than the analytical Gaussian (8) and 600× times fewer than convolution.

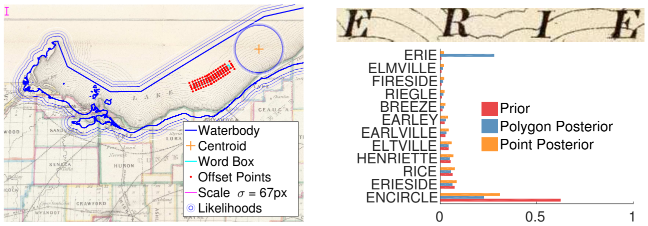

Figure 3. Lake name recognition: contours of polygon and point likelihoods $P\left(\mathbf{x} \mid f, \hat{p}, \hat{\mathbf{A}}, \hat{\sigma}\right)$ for f=1075813 (GNIS ID for “Lake Erie”) and comparison of prior $P\left(\mathbf{y} \mid \mathbf{W}, \boldsymbol{\Theta}\right)$ and posteriors $P\left(\mathbf{y} \mid \mathbf{x}, \mathbf{W}, \hat{\boldsymbol{\Upsilon}}, \boldsymbol{\Theta}\right)$.

4) Polygon Feature Model: We also require a likelihood for geographic features covering closed regions, rather than points or polylines. We construct the probability of a point $\mathbf{x}$ deviating from a prior uniform within the feature f's projected and aligned polygon $\mathcal{A} = \mathbf{A}\overline{\mathbf{C}}_p\left(f\right) \subset \mathbb{R}^2$ with area $|\mathcal{A}|$. Ultimately, we also adopt the distance model for areas due to complications with the convolution integral for polygons.

A Gaussian model of deviation from any of the uniformly likely points in region $\mathcal{A}$ yields the convolution integral

$$P\left(\mathbf{x} \mid \mathcal{A}, \sigma\right) = \frac{1}{|\mathcal{A}|}\int_{\mathcal{A}} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma\right) \mathrm{d}\boldsymbol{\mu}, \tag{11}$$

where the center of the Gaussian is a point in the polygon. While the convolution of a Gaussian with a polyline has an analytical solution (7) [14], we find no analytical equivalent for a convolution over a polygon area. However, Hubert [17] recently noted that an area integral like (11) can be transformed to a contour integral over the boundary via Green's theorem, which gives rise to analytical solutions for other, non-Gaussian kernels (e.g., squared Cauchy). Unfortunately, the computational burden remains $O\left(NFS\right)$, where S is the number of polygon boundary segments; discretizing polygons to convolve with a Gaussian also remains $O\left(N + FDK\right)$.

To address the inefficiency in the case of very large K, we once again use the kernel (9) to give the distance a Gaussian distribution, yielding the label placement likelihood

$$P\left(\mathbf{x} \mid o, f, \mathbf{A}, p, \sigma\right) \propto K_\sigma\left(D_{\mathcal{A}}\left(\mathbf{x}_o\right)\right), \tag{12}$$

where $D_{\mathcal{A}}\left(\mathbf{x}\right)$ is the distance transform for the function

$$M_{\mathcal{A}}\left(\mathbf{x}\right) = \begin{cases} 0 & \mathbf{x} \in \mathcal{A} \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

Thus $D_{\mathcal{A}}\left(\mathbf{x}\right)$ gives the minimal distance from $\mathbf{x}$ to the area $\mathcal{A}$, with $D_{\mathcal{A}}\left(\mathbf{x}\right) = 0$ for $\mathbf{x} \in \mathcal{A}$ so that the kernel value is maximal and uniform within the polygon, as shown in Fig. 3. The example shows a polygon likelihood and the same likelihood level contours if the feature were instead modeled as a point (i.e., the centroid). The word is too distant to be corrected by a point model (4), but the polygon likelihood (12) exceeds the outlier likelihood (5) enough to correct the word posterior.

With the distance transform [16], we calculate the unnormalized likelihood over a discrete image domain and sum the discrete values to normalize, a relatively efficient $O\left(N + FD\right)$ process.

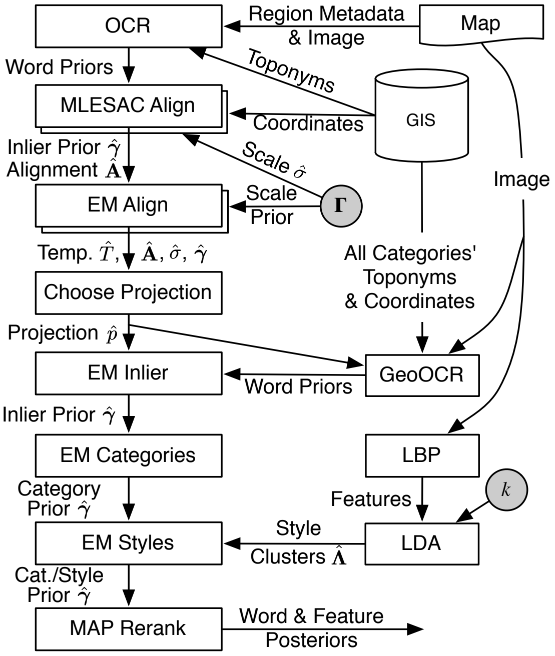

Figure 4. Learning process for the Bayesian model of Fig. 1. Outputs are cumulative and displayed only where updated.

IV. Coarse-to-Fine Inference

Using the Bayesian network of Fig. 1, we georeference each map by finding the most probable projection $\hat{p}$ and alignment $\hat{\mathbf{A}}$. Given these (and other) estimated values, we then calculate MAP estimates to recognize the strings $\mathbf{y}$ and underlying geographic features f for each cropped word.

Formally, we seek parameters $\boldsymbol{\Upsilon} = \langle T, \sigma, \mathbf{A}, p, \gamma, \boldsymbol{\Lambda}\rangle$ that maximize the log posterior probability of the data,

$$\begin{aligned} \mathcal{O}\left(\boldsymbol{\Upsilon}\right) \quad \triangleq \quad & \sum_i \log P\left(\mathbf{x}^i \mid \mathbf{W}^i, \boldsymbol{\Upsilon}, \boldsymbol{\Theta}\right) \\ & \log P\left(\sigma \mid \boldsymbol{\Gamma}\right) + \log P\left(\boldsymbol{\Lambda} \mid \left\{\mathbf{W}\right\}, k\right), \end{aligned} \tag{14}$$

where the remaining values ($\mathbf{y}$, o, s, c, and f for all N words) are efficiently marginalized. Limiting model flexibility early in the process is important to prevent poor fits and degenerate models in an objective fraught with local maxima. For each map, we therefore begin by using MLESAC [18] to produce a coarse alignment on a small subset of geographical categories, then refine the parameters with expectation maximization (EM), before finally adding more categories and expanding the representational power of the model with text styles. Fig. 4 illustrates the process detailed in the remainder of this section.

A. MLESAC: Initial Alignment

We use an MLESAC [18] variant to initialize the process, optimizing Eq. (14) for $\mathbf{A}$ and γ, with other values fixed [1]. To further limit model flexibility early, we set $k = 1$ to ignore styles, collapse all (inlier) categories together, and use only *Civil* and *Populated Place* entities modeled as points (even states and counties). To reduce the set of feature candidates and speed the search, we consider only the top two strings for each word and fix $\hat{T} \leftarrow 1$, retaining overly-confident OCR scores—correct words will agree with $\mathbf{A}$ and the rest will be rejected as outliers. The inverse gamma prior fixes the scale

$$\hat{\sigma}^2 \leftarrow \|\mathbf{\Gamma}\|^2 \Big/ |\mathbf{\Gamma}|, \tag{15}$$

with $|\mathbf{\Gamma}|$ the vector size (number of prior observations).

MLESAC iteratively samples three word images, then one feature and offset from each to estimate $\mathbf{A}$ from the resulting point correspondences; the outlier prior $\gamma_\emptyset$ is discretely sampled in $[0, 1]$ to maximize (14), resulting in the intial estimates $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\gamma}}$. Whereas Weinman [1] restricts sample models to be nearly vertical (i.e., Northerly) and match the region's scale within a factor of ten, we only limit distortions by limiting the shear magnitude to 0.05 and the axis scale ratio to 1.2. We run MLESAC for each projection p, preserving the projection-dependent results until after the next stage (EM), when we keep only the projection maximizing the objective $\mathcal{O}(\boldsymbol{\Upsilon})$.

B. EM Alignment

The parameter estimates from MLESAC initialize an iterative expectation maximization (EM) algorithm to further optimize the model for σ while updating $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\gamma}}$. In the first EM stage, we also interleave an optimization for the OCR prior's temperature T, a "T Step" to adapt the tradeoff between GIS and OCR; the initial value $\hat{T}$ is a maximum likelihood estimate from the training maps' OCR scores (1) alone. The updates to $\hat{T}$, $\hat{\mathbf{A}}$, $\hat{\sigma}$, and/or $\hat{\boldsymbol{\gamma}}$ described in this section iterate until the objective (14) ceases to improve.

To focus on the alignment task early, for now we keep $k = 1$ to ignore style and continue with the restriction to one (inlier) category using only *Civil* and *Populated Place* entities.

1) T Step: In the first stage, we precede each EM iteration with a direct optimization of $\mathcal{O}(\boldsymbol{\Upsilon})$ for $\log T$ (to keep the temperature positive) by gradient descent.

2) E Step: Given $\hat{\boldsymbol{\Upsilon}}$, the current values of the map-wide parameters to be estimated, we calculate the joint posterior of each feature, candidate offset, and text style for all words

$$\pi^i_{fos} \triangleq P\left(f^i, o^i, s^i \mid \mathbf{x}, \mathbf{W}, \hat{\boldsymbol{\Upsilon}}, \boldsymbol{\Theta}\right). \tag{16}$$

Because the Bayesian network is a polytree, the posterior is calculated in linear time and accelerated when the feature marginal is cached by combining (1) and (2) to marginalize out the unknown string $\mathbf{y}$ for each word image using the current temperature $\hat{T}$. For each word image i, the domains of features f^i and o^i are different (based on the candidate strings produced by the OCR module and the geometry of the localized word); we omit this dependence for clarity of notation when possible. With one style ($k = 1$) during the first stage, the posterior's last dimension is a singleton, so $\pi^i_{fo} \triangleq \pi^i_{fos}$.

3) M Step: We only estimate $\hat{\sigma}$ and $\hat{\mathbf{A}}$ in the first EM stage, but $\hat{\boldsymbol{\gamma}}$ is updated in all stages, as categories and styles are added to the model.

Updating $\hat{\sigma}$ requires the squared error for each feature location prediction,

$$E^i_{fo} \triangleq \left\|\mathbf{x}^i_o - \hat{\mathbf{A}}\overline{\mathbf{C}}_p\left(f^i\right)\right\|^2, \forall f \neq f_\emptyset \tag{17}$$

with a Bayesian posterior EM update (for $f \neq f_\emptyset$),

$$\hat{\sigma}^2 \leftarrow \frac{\|\mathbf{\Gamma}\|^2 + \sum_{i,f,o} E^i_{fo}\pi^i_{fo}}{|\mathbf{\Gamma}| + \sum_{i,f,o}\pi^i_{fo}}. \tag{18}$$

Intuitively, this is analogous to the standard Gaussian mixture model EM update, where squared errors are weighted by contribution and normalized by the contributors' weights. The elements of $\mathbf{\Gamma}$ serve as prior observations for the posterior inverse Gamma distribution on σ^2.

The observed positions $\mathbf{x}_o$ have a conditional linear Gaussian distribution, which depends on the discrete feature f and offset o, as well as the continuous affine transform $\mathbf{A}$, which must be updated in the M-step. We recover this latent linear factor by solving the standard weighted normal equations [19]

$$\begin{aligned}\bar{\hat{\mathbf{A}}} \;\leftarrow\; & \left(\sum_{i,f,o}\pi^i_{fo}\bar{\mathbf{x}}^i_o\overline{\mathbf{C}}_p\left(f^i\right)^\top\right)\times \\ & \left(\sum_{i,f,o}\pi^i_{fo}\overline{\mathbf{C}}_p\left(f^i\right)\overline{\mathbf{C}}_p\left(f^i\right)^\top\right)^{-1}\end{aligned} \tag{19}$$

over $f \neq f_\emptyset$, where $\bar{\hat{\mathbf{A}}} \in \mathbb{R}^{3\times 3}$ indicates the transform matrix $\hat{\mathbf{A}} \in \mathbb{R}^{2\times 3}$ augmented by row $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

In all EM stages, the category-style prior probability is updated with the total proportion of each word's contribution

$$\hat{\gamma}_{cs} \leftarrow \frac{1}{N}\sum_i\sum_o\sum_{f\in F(c)}\pi^i_{fos}. \tag{20}$$

4) Projection Selection: After the first EM stage, we have estimates $\hat{T}$, $\hat{\mathbf{A}}$, $\hat{\sigma}$, and $\hat{\boldsymbol{\gamma}}$ for each projection p. We conclude by selecting the projection $\hat{p}$ that maximizes $\mathcal{O}(\boldsymbol{\Upsilon})$ with its observed companion parameters $\boldsymbol{\phi}$.

5) GeoOCR: We now fix $\hat{T}$, $\hat{\mathbf{A}}$, $\hat{\sigma}$, and $\hat{p}$ for the rest of the process, which provides a necessary form of regularization as we expand the feature domains F to include all GNIS classes described in Section III-B1. However, the category model remains "collapsed" to the simpler inlier/outlier dichotomy.

For each word image, we accumulate additional alignment-driven string scores by re-running OCR in a "closed vocabulary" mode using a lexicon consisting only of strings from all features f for which the location-based likelihood exceeds an outlier threshold,

$$P\left(\mathbf{x}_o \mid f, \hat{\mathbf{A}}, \hat{p}, \hat{\sigma}\right)\kappa > P\left(\mathbf{x}_o \mid f_\emptyset, \hat{\mathbf{A}}, \hat{p}, \hat{\sigma}\right)(1-\kappa), \tag{21}$$

with the optimistic inlier probability $\kappa = 0.95$. Although not guaranteed to recover a parse score for strings from every feature in the vicinity, the beam search is less likely to prune out a correct string from the much-reduced lexicon.

With the newly expanded set of string candidates, we reiterate the EM update (20) for $\hat{\boldsymbol{\gamma}}$, now using features from all

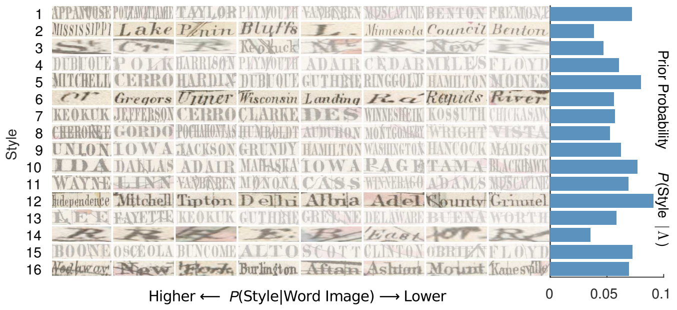

Figure 5. Example style clusters and prior probabilities, with top words sorted and transparency-weighted by the probability of the word containing that style (lighter words have lower likelihood).

GNIS classes collapsed to a single inlier category. We speed the process significantly by caching the feature likelihoods (4), (10), and (12), which do not change with γ.

6) EM Category Updates: With the expanded set of features we finally broaden γ from a simple inlier/outlier model to a prior that distinguishes among feature categories, while continuing to forego style. To estimate a prior over all categories in Section III-B1, we run another round of EM updates to $\hat{\gamma}$, initializing the outlier prior $\hat{\gamma}_{\emptyset}$ to its value from the previous stage and uniformly dividing the inlier probability among the categories.

7) EM Style Updates: As the final learning step, we associate styles with categories. For each word image i, the style prior $P\left(s^i \mid \mathbf{W}^i, \mathbf{\Lambda}, k\right)$ is a topic posterior from an unsmoothed LDA model learned with variational EM (optimizing the concentration parameter α) [13]. With these priors fixed, our E Step calculates the joint posteriors π^i_{fos}, where the style dimension s now has k entries; the M step iteratively updates with Eq. (20). We initialize $\hat{\gamma}$ for each style with values from the previous stage. While LDA's EM process could integrate with these EM updates, predictions seem unlikely to improve.

Fig. 5 illustrates the similarity of top-ranking words for each style on a map. The style topic clusters tend to capture different typefaces and the prior probability indicates the relative prevalence of each style. Because LDA learns a mixture of styles for each word, some clusters also seem to represent other graphical information, such as graticular lines at particular orientation, railroad ticks, or background hatching texture. In addition, when the number of styles k exceeds the number of fonts, sometimes styles capture the prominent or frequent characters of the words, due to the small "document" size. Fig. 6 contrasts the learned category prior with and without a style model and demonstrates how styles cluster categories.

V. Experiments

With estimates for each map's parameters $\hat{\mathbf{\Upsilon}} \triangleq \left\langle \hat{T}, \hat{\sigma}, \hat{\mathbf{A}}, \hat{p}, \hat{\gamma}, \hat{\mathbf{\Lambda}} \right\rangle$, we calculate predictive marginal posteriors for strings $P\left(\mathbf{y} \mid \mathbf{x}, \mathbf{W}, \hat{\mathbf{\Upsilon}}, \mathbf{\Theta}\right)$ and features $P\left(f \mid \mathbf{x}, \mathbf{W}, \hat{\mathbf{\Upsilon}}, \mathbf{\Theta}\right)$. Here we present results on 20 annotated maps (8,590 words), holding the rest for tuning.

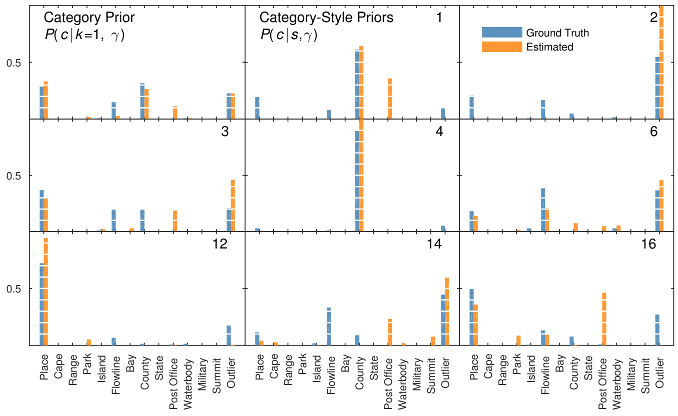

Figure 6. Comparison of example ground truth and learned style-conditional category probabilities $P\left(c \mid s, \gamma\right)$ for $k = 1$ and the $k = 16$ styles shown in Fig. 5. Omitted styles closely resemble $s = 4$, which is peaked at $c =$ County.

Table I
Text recognition results of individual maps, grouped by atlas.

Map	Num. Words	Word Err. (%) Prior	Word Err. (%) Post	Char. Err. (%) Prior	Char. Err. (%) Post	Mean Rank Prior	Mean Rank Post
1592006	285	39.65	27.02	16.34	11.05	1.507	1.335
5370006	352	53.69	41.76	27.21	22.98	1.839	1.534
5370026	210	44.29	36.19	20.95	19.68	1.553	1.385
1070001	250	40.80	27.20	24.12	17.62	1.499	1.314
1070005	216	45.83	24.07	25.84	12.63	1.509	1.242
1070006	460	52.61	38.04	34.95	25.88	1.723	1.444
1070007	322	55.28	35.71	32.53	19.87	1.768	1.442
1070010	218	37.61	19.27	17.11	9.39	1.409	1.178
1070012	214	48.13	33.64	26.88	17.60	1.651	1.398
5235001	367	37.87	28.34	17.97	15.66	1.436	1.304
5242001	391	24.04	11.25	10.14	4.31	1.227	1.100
5755018	864	5.56	1.97	1.60	0.62	1.043	1.015
5755024	1023	6.65	3.52	2.14	1.05	1.050	1.025
5755025	759	14.89	6.06	6.12	2.30	1.147	1.051
5755035	432	13.89	6.02	4.03	1.64	1.140	1.055
5755036	592	9.97	5.41	3.35	1.81	1.092	1.047
5028052	631	8.87	5.86	2.43	1.76	1.075	1.053
5028054	320	9.06	5.00	3.23	2.09	1.071	1.041
5028100	487	6.37	5.75	2.24	2.07	1.061	1.052
5028102	197	8.63	6.09	2.98	2.10	1.079	1.062

For each map, Table I reports word error, character error (total edit distance of all words divided by the number of characters annotated in the map), and harmonic mean of the correct words' ranks by probability; initial OCR predictions are compared to the final integrated model. Maps from some atlases are clearly much easier for OCR than others, but all experience significant error reduction due to georeferencing-based recognition (10–65%). On a subset of eight challenging 19th century maps, we reduce final word recognition error from an average of 55% (reported by Weinman [1]) to 31%. The performance improvement is due to both the initial OCR as well as the expanded model and parameter learning.

For the entire test corpus, Table II measures how each stage contributes to accuracy; "Map" averages error rates over the 20 maps, while "Word" aggregates errors over all annotated words in the corpus. We report baseline OCR performance of LSTM-based Tesseract 4.00 on the same RGB word images. Our

Table II
EVALUATION OF MODEL VARIATIONS ON TEXT RECOGNITION.
(*SIGNIFICANT IMPROVEMENT; ONE-SIDED SIGNED-RANK TEST $p < .01$)

		Word Error (%)		Char. Error (%)	
Pipeline Stage	Sec.	Map	Word	Map	Word
Tesseract 4.00	-	36.8 ± 5.0	28.77	15.4 ± 2.6	10.92
OCR	III-A	28.2 ± 4.2*	22.29	14.1 ± 2.6	10.19
MLESAC	IV-A	22.1 ± 3.4*	17.57	11.4 ± 2.1*	8.21
EM Align	IV-B4	19.9 ± 3.3*	15.36	10.0 ± 2.0*	7.07
EM Inlier	IV-B5	19.6 ± 3.3	15.16	10.4 ± 2.1	7.30
EM Category	IV-B6	18.9 ± 3.2*	14.58	9.9 ± 2.0*	6.95
EM Style	IV-B7	18.4 ± 3.1*	14.23	9.6 ± 1.9	6.75

Table III
GEOGRAPHIC ENTITY RECOGNITION VERSUS MODEL STAGE.

	Num.	Feature Error (%)			Category Error (%)		
Map	Lab.	Inlier	Categ	Style	Inlier	Categ	Style
1070001	198	66.40	53.28	35.61	46.70	35.86	16.92
1070007	233	64.97	49.74	40.26	55.10	38.43	26.02
5242001	333	51.03	38.98	32.31	31.60	18.05	12.13
5755035	339	42.73	34.08	35.01	24.63	13.25	14.24
5028100	377	32.05	30.49	24.39	14.50	12.49	6.13
5028102	131	46.53	40.14	36.13	28.97	20.48	17.05
Label Average		48.39	39.38	32.89	31.18	21.09	14.17

OCR boasts a 22% error reduction over Tesseract on words, though only 6.7% for characters, suggesting the importance of tuning dictionary strength. Overall, our system cumulatively reduces word error by 36% over OCR alone, and 20% over prior work [1]. Simply adjusting alignment and error scale by EM (along with choosing a best-fit projection) nets a 13% improvement; learning category priors reduces error further. While the style model also trims word error, it has a more substantial impact on geographic feature matching.

We annotated labels in six maps with GNIS feature IDs. Table III measures the accuracy of matching a label to its geographic feature f or the general category c for which $f \in F(c)$. Because a label may contain multiple words, label accuracy is defined as the accuracy among its constituent words, *excluding* any outlier predictions; if all words are predicted outliers, the label is correct when the ground truth has no GNIS match, incorrect otherwise. Explicitly associating GNIS entities with their category reduces feature error by 19% and general categorization error by 32%. Conditioning category predictions on text style reduces feature errors another 16% and categorization error by 33%. Cumulatively, the category-style model eliminates nearly one-third of the inlier model's feature errors and halves category errors.

Over half the category uncertainty is explained by style: averaged over all 20 maps, the uncertainty coefficient is Mutual Information $(c, s)/$ Entropy $(c) \approx 0.553 \pm 0.039$.

VI. CONCLUSIONS

We introduced new likelihood models for non-point features and developed adaptive category-style links that together leverage dependencies between feature label placements, categories, text styles, and known geography. Unsupervised learning of the integrated probability model parameters drastically reduces word recognition and feature-matching errors.

In the future we plan to integrate word/text detection methods (e.g., Chiang and Knoblock [20]) for comparing end-to-end processing.

ACKNOWLEDGMENTS

The author gratefully acknowledges: A. Nicolaou for helpful discussions and code support; T. Baratta, C. Nguyen, and B. Wang for pilot explorations; D. Cambronero Sanchez and L. Boateng Asante for reviewing map annotations. This material is based upon work supported by NVIDIA and the National Science Foundation under Grant No. 1526350. High performance computing resources were provided by the UMass Amherst College of Information and Computer Sciences under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative.

REFERENCES

[1] J. Weinman, "Toponym recognition in historical maps by gazetteer alignment," in *Proc. ICDAR*, 2013, pp. 1044–1048.

[2] Y.-Y. Chiang, S. Leyk, and C. A. Knoblock, "A survey of digital map processing techniques," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1:1–1:44, May 2014.

[3] J. P. Snyder, "Map projections: A working manual," U.S. Geological Survey, Washington, D.C., Tech. Rep. Professional Paper 1395, 1987.

[4] R. Yu, Z. Luo, and Y.-Y. Chiang, "Recognizing text on historical maps using maps from multiple time periods," in *Proc. ICPR*, 2016, pp. 3993–3998.

[5] A. Tarafdar, U. Pal, P. P. Roy, N. Ragot, and J.-Y. Ramel, "A two-stage approach for word spotting in graphical documents," in *Proc. ICDAR*, 2013, pp. 319–323.

[6] M. Rusiñol, R. Roset, J. Lladós, and C. Montaner, "Automatic index generation of digitized map series by coordinate extraction and interpretation," *e-Perimetron*, vol. 6, no. 4, pp. 219–229, 2011.

[7] R. Pawlikowski, K. Ociepa, U. Markowska-Kaczmar, and P. B. Myszkowski, "Information extraction from geographical overview maps," in *Proc. ICCCI*, 2012, pp. 94–103.

[8] A. Gelbukh, H. SangYong, and S. Levachkine, "Combining sources of evidence to resolve ambiguities in toponym recognition in cartographic maps," in *Proc. GEOPRO*, 2003, pp. 42–51.

[9] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild, "Toward integrated scene text reading," *IEEE Trans. PAMI*, vol. 36, no. 2, pp. 375–387, Feb. 2014.

[10] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Intl. J. Comp. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[11] A. Nicolaou, A. D. Bagdanov, M. Liwicki, and D. Karatzas, "Sparse radial sampling LBP for writer identification," in *Proc. ICDAR*, 2015, pp. 716–720.

[12] M. A. Ferrer, A. Morales, and U. Pal, "LBP based line-wise script identification," in *Proc. ICDAR*, 2013, pp. 369–373.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Machine Learning Research*, vol. 3, pp. 993–1022, Jan 2003.

[14] X. Jin and C.-L. Tai, "Analytical methods for polynomial weighted convolution surfaces with various kernels," *Computers & Graphics*, vol. 26, no. 3, pp. 437–447, 2002.

[15] J. Bloomenthal and K. Shoemake, "Convolution surfaces," in *Proc. SIGGRAPH*, 1991, pp. 251–256.

[16] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Trans. PAMI*, vol. 25, no. 2, pp. 265–270, 2003.

[17] E. Hubert, "Convolution surfaces based on polygons for infinite and compact support kernels," *Graphical Models*, vol. 74, no. 1, pp. 1–13, 2012.

[18] P. H. S. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *CVIU*, vol. 78, no. 1, pp. 138–156, Apr. 2000.

[19] K. P. Murphy, "Fitting a conditional linear Gaussian distribution," Tech. Rep., October 1998.

[20] Y.-Y. Chiang and C. A. Knoblock, "Recognizing text in raster maps," *GeoInformatica*, vol. 19, no. 1, pp. 1–27, 2015.