An Uncertainty Aware Method for Geographic Data Conflation

Haowen Lin
Computer Science Department
University of Southern California
Los Angeles, CA
haowenli@usc.edu

Yao-Yi Chiang
Spatial Sciences Institute
University of Southern California
Los Angeles, CA
yaoyic@usc.edu

ABSTRACT

With a significant amount of spatial data archives online, data conflation is becoming more and more critical in the domain of Geographical Information Science (GIScience) because of its broad applications such as detecting the development of road networks and the change of river course. Existing conflation approaches usually rely on the vector data of corresponding features in multiple sources to have an approximate location. However, they commonly overlook the uncertainty produced during the vector data generation process in the data sources. In previous work, we presented a Convolutional Neural Networks (CNN) recognition system that automatically recognizes areas of geographic features from maps and then generates a centerline representation of the area feature (e.g., from pixels of road areas to a road network). In this paper, we propose a method to systematically quantify the uncertainty generated by an image recognition model and the centerline extraction process. We provide an end-to-end evaluation method that exploits the distance map to calculate the uncertainty value for centerline extraction. Compared with methods that do not consider uncertainty value, our algorithm avoids using a fixed buffer size to identify corresponding features from multiple sources and generate accurate conflation results.

CCS CONCEPTS

• **Information System** → Geographic Information System;

KEYWORDS

Historical maps, Vector data conflation, Uncertainty

1 INTRODUCTION

Because of the high capability of computing systems and storage as well as the Internet, there is a rapid growth of geospatial data online. A great amount of organizations have scanned historical maps and archived them in digital format. For example, the United States Geological Survey (USGS) database,[1] as one of the most significant online sources for topographic maps, contains thousands of scanned map documents ranging from 1884 to 2006, covering entire United States states. Other map sources like OpenStreetMap[2]offer not only free-downloaded maps but also other kinds of geospatial data like locations of highways and restaurant names. Besides map sources from organizations, remote sensors, usually carried by satellites, also provide numerous images about physical surfaces on the Earth with real-time and long-time series information on large geographic features. For instance, as the earliest satellites image dataset covering the entire territory of the United Kingdom, Land Cover Map 2000 (LCM2000) [3] collects different kinds of landforms and plants variation across seasons. All those spatial images are essential information sources for various disciplines such as biology, history and social sciences[5].

Due to the availability of a large quantity of spatial data, many studies [14, 16, 18, 20] are focusing on extracting geographic features from multiple sources and integrating them for future analysis. The process of integrating information from multiple sources is called "conflation." Conflation on spatial data is important for several reasons. First, because different datasets provide different types of features, we can add up all features of the same location to obtain a comprehensive evaluation about this region (eg., the satellites images provide real-time information of the earth surface while maps offer location names). Second, the differences of the same geographic or human-made features across long time dimensions imply a specific pattern about the development of biography and human activities inside this area. For instance, a user can compare vector data of railroads in the same area but from different time periods for change analysis to identify the development of railroads over time.

One limitation in current approaches of integrating vector data for analysis is that most approaches do not incorporate the uncertainty arises during the vector data generation process. However, the uncertainty value is necessary in the conflation stage. For geospatial data, uncertainty is generated in many places including data collection and image processing. This paper focuses on quantifying the two major sources of uncertainty in vector-to-vector geospatial data conflation. Specifically, one of the vector data in the conflation process is the centerline representation of geographic features(e.g., road networks) extracted from images.

The first source of uncertainty comes from the feature recognition model. A common procedure in vector-to-vector data conflation is to first extract geographic features from raster images and then compare the features with vector data from another data source. Because it is impossible to manually extract features from raster images considering existing millions of maps and satellites images, we need an automatic geographic feature recognition system to separate features from background pixels. In [8], we built an

[1]https://www.usgs.gov/products/maps/overview

[2]https://www.openstreetmap.org/

[3]https://www.ceh.ac.uk/services/land-cover-map-2000

ACM SIGSPATIAL, November 2018, Seattle, Washington, USA

ACM ISBN 978-1-4503-6041-8/18/11…$15.00
https://doi.org/10.1145/3282834.3282842

automatic recognition system that utilizes Convolutional Neural Networks (CNN) to handle the challenges of extracting geographic features from historical maps. In the feature recognition step, each predicted pixel has an associated uncertainty value. Typically the recognition system discards pixels with uncertainty higher than 50% probability before extracting the centerline representation of the feature. In this paper, we pass through the uncertainty generated by the recognition system to the centerline extraction process for an uncertainty-aware conflation process.

The second type of uncertainty comes from the centerline extraction process. After we acquire linear features from raster data, a common practice is to extract the centerline representation and use this unit resolution vector data to represent the extracted linear features (See in Figure 1). There exist a great number of =skeleton algorithms which extract the structure information of the curves or strokes [1, 2, 9]. However, these algorithms do not preserve the original shape of the features, and some pixels of centerlines representation have deviated from the center. For example, in Figure 2, we can see after applied the thinning algorithm, the variance of the line width in the original image is lost, and the algorithm produces jagged line pieces for irregular line conjunctions.

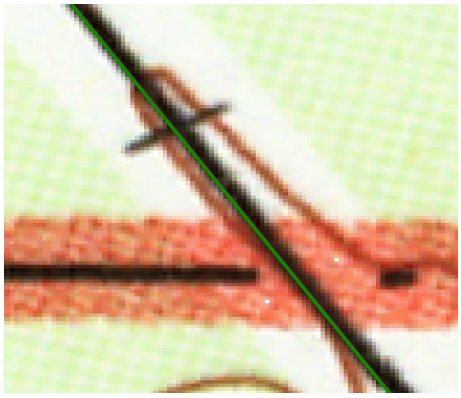

Figure 1: Using vector data to represent railroad features on the map

In this paper, we present an approach that systematically quantifies the uncertainty of geographic features produced during the recognition and centerline extraction process, and then we use the uncertainty to improve the conflation result. We first record the uncertainty of pixels in recognition model and then evaluate the thinning algorithm and generate the buffer size of centerline by the distance map. We compare the mismatched pixels between buffered output and recognition result to further quantify the uncertainty value. Finally, we present how to use the pixels uncertainties in the conflation process.

In the remainder of this paper, Section 2 discusses the related work, Section 3 presents the overall procedure for measuring the uncertainty, Section 4 presents the experiment result, and Section 5 discusses the future work.

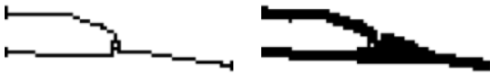

Figure 2: Left: Original image file with irregular junctions. Right: The centerline representation of left pictures after applied thinning algorithm

2 RELATED WORK

Thinning Algorithms. In order to get the vector data of image objects, one significant step is to generate the centerline representation of linear features. A large number of theoretical and practical methods have been investigated. Generally, the thinning algorithms are categorized based on their thinning methods. Classical approach to extract centerlines usually relies on digital morphological erosion [3, 12, 15]. In 1980, Theo has already proposed a skeleton algorithm that iteratively erodes the boundary of the pixels while preserving the connectivity of the original objects at the same time [15]. In his paper, Theo explores in depth the geometry principle of the image objects and defines the edge-strength by using a linear diffusion equation that helps to deal with grayscale noise in the images [15]. Another popular method is based on the Voronoi diagram of boundary that helps to generate the medial axis. The medial axis is the point sets where each point has more than one closest points on the boundary (2D) or more than two points on the surface (3D) [7, 10]. This feature makes it possible to efficiently regenerate the original objects. For example, Giesen et al. propose an improved medical axis transform that not only enforces topology of the object but also removes insignificant brunches in traditional medical axis method [10].

However, there is hardly a standard criterion to measure the performance of these thinning algorithms because they are serving for different purposes. Some thinning algorithms aim to avoid corner distortions under different image artifacts while the others are trying to reduce the errors in the reconstruction of original image objects. The shapes of the thinned centerlines in the output also vary a lot in different methods, and the variances can significantly affect the vector data and thus influence conflation result.

Data Conflation Conflation in geospatial datasets has always been a spotlight in the research community and commercial software development of GIS because of its wide range of applications. For example, The JUMP project has developed many powerful commercial software tools for GIS data analysis and conflation [17]. However, the differences in image resolution make it challenging to integrate features from two independent datasets accurately [18]. As a result, a great amount of effort has been put into automatic feature alignment, and traditionally they fall into two categories: vector-to-vector alignment and vector-to-raster alignment. For vector-to-vector alignment, researchers commonly use point-based methods and line-based methods to link the entities. In [19] Volz integrated street road data by topological splitting the street objects to add additional nodes for a precise matching and iteratively calculating the similarity between nodes and edges. The algorithm considers the emanating edges, distance, direction as the criterion

to calculate the similarity. For vector-to-raster integration, vector data plays an important role in locating the features on the images. Chen et al. propose an integration method that map vector data onto street maps. They first identify the road intersection points on roads recognized on images that have been pre-processed by the thinning algorithm and then link them with the intersection points on the vector data by geographic rules [4].

As can be seen in all these work, no matter what approach the researchers use for information integration, the location of vector data points is an important criterion to measure the similarity between the vector data do not always correctly align with corresponded entities in the other dataset. To deal with this problem, their work either use a fixed distance to search neighborhood points [19] or use a fixed size to buffer vector data [4] in order to link the corresponding features.

In contrast, our algorithm provides a solution to measure the uncertainty generated by the recognition model and thinning algorithms. With the uncertainty, the algorithm is able to use a more precise matching metric compared with previous methods. Besides, our algorithm avoids using a fixed buffer size for the vector data that may cause incorrect conflation result especially when the width of the feature varies a lot on the images.

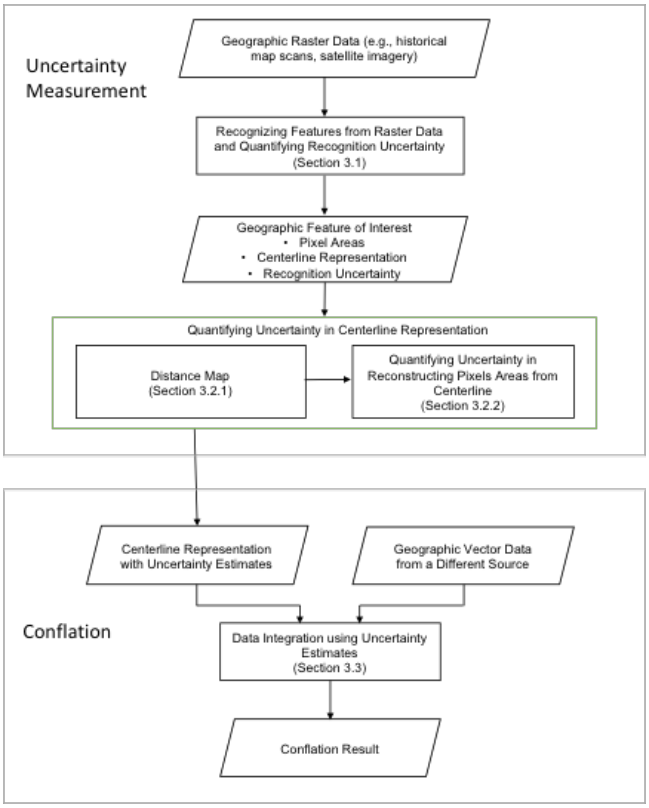

Figure 3: The workflow of uncertainty quantification and conflation process

3 OVERALL APPROACH

In this paper, as a case study to illustrate our approach, we perform an analysis to determine changes of railroads in a USGS historical map and contemporary vector data of the same region. We first describe the procedure of generating uncertainty given the image recognition result and the centerline extraction algorithm. After we run the thinning algorithm to obtain the centerline of the geographic feature, we record the recognition uncertainty and use the "distance map" to evaluate the uncertainty generated in the thinning process. In our case study, we use the Guo-Hall two-subiteration thinning algorithm as an example because of its wide usage and high computing efficiency [11]. We then show the steps of recording the uncertainty value of each pixel in the extracted centerline. The uncertainty values represent how accurate the centerline is if we want to reconstruct the exact pixel area of the geographic feature on the map. We finally present the procedure of using the uncertainty values to improve the conflation result. Figure 3 shows the workflow of the uncertainty quantification process in our approach and the conflation process that utilizes uncertainty value.

In the following sections, we denote the raster output from the recognition model as **recognition uncertainty map**, the centerline of recognition result with uncertainty value as **centerline representation**, , the intermediate buffered output using distance map value as **reconstruction map**,the final raster buffered output adjusted by distance map and centerline uncertainty as **probability map**.

3.1 Uncertainty in Recognition System

In [8], we developed a geographic feature recognition system using the Fully Convolutional Network (FCN). FCN was proposed by Long et al. in 2015 and was trained in pixel-to-pixel, which showed great success in spatial dense prediction tasks [13]. Inspired by the recent work of [13], our recognition system trains the FCN network and takes advantages of the skip architecture that combines information from the intermediate layer with the information from final layer to refine the spatial resolution in the result. Finally, the system produces a grayscale raster image with each pixel taking a value between 0 and 1, which represents the uncertainty of this pixel to be classified as the feature of interest (e.g., 0 means the network is very confident that this pixel is a part of the feature of interest because it has zero uncertainty). Figure 4 shows a part of the recognition uncertainty map of our recognition model. In previous work, the recognition system removes the pixels with uncertainty value higher than 0.5 in the final prediction output because these pixels have a great chance to be the false positives and classify all remaining pixels to be the true positives. In the experiment of extracting geographic features on a variety of historical maps, the system achieves correctness of 84.74% and completeness of 97.46% [8]. However, discarding uncertainty value potentially loses useful information and therefore in this paper, we record the recognition uncertainty value σ_i for pixel $p_i \in C$ where $\sigma_i \in [0, 1]$ and C is the extracted centerline representation to improve the performance of matching railroads across multiple data sources. We assume the median of the uncertainty values in the centerline pixel's neighbors can better represent the uncertainty of this pixel than the value

of itself because there is much "noisy objects" intersecting the features, the median value is effective in filtering uncertainty value generated by the noise background. Assuming p_1 is the current pixel to be processed, $p_2, p_3 ... p_9$ are its neighborhood pixels in the recognition result (Figure 5), and the corresponded uncertainty value is σ_1.

$$\sigma_1 = \tilde{X}, X = \{I(p_1), I(p_2), ...I(p_9)\} \quad (1)$$

where $\tilde{X}$ is the median of X which contains the foreground neighborhood pixels and $I(p_i)$ is the uncertainty value of pixel p_i in the recognition result.

3.2 Uncertainty in Centerline Representation

After we obtain the recognition uncertainty map (i.e., pixel areas of the geographic feature on interest), it is common to extract the centerline representations to represent original linear features in the raster images and save the centerline as vector data to save data storage. However, because this step of data compression causes information loss on the width (area) of the extracted feature pixels, we exploit the distance map to obtain the distance between pixels in the centerlines and pixels in the nearest boundary of the extracted feature pixels. This distance information help to describe the shape of the recognition result (Section 3.2.1). Next, we use the distance information to reconstruct the the recognition result and record the reconstruction errors as the uncertainty of the centerline representation (Section 3.2.2).

3.2.1 Distance Map. The distance map is useful in many image processing applications such as skeleton calculation and expansion of image objects. In [6], Danielsson proposed and analyzed an eight-point sequential Euclidean distance mapping (8SED) algorithm[6]. According to the algorithm, we view each pixel in the image as a vector of two positive integer components consisting of the x-direction and y-direction. Initially, the distance of each pixel is set to be

$$L(x,y) = (z, z) \text{ if } (x,y) \in S$$
$$L(x,y) = (0, 0) \text{ if } (x,y) \in \overline{S}$$

where L is the distance mapping procedure, z represents an infinite integer, S represents the geographic features and $\overline{S}$ represents the background. Figure 6 shows an example of the integer component output by using the 8SED-algorithm. The pixels outside the boundary are first initialized to infinite, and pixels inside the boundary are initialized to zero. Suppose we are at the boundary background pixel with integer components (0,0), and we want to propagate distance information to the feature pixel in the west. We add 1 to the x-axis integer components and update the integer component of the feature pixel from (z,z) to (1,0). Likewise, if we want to propagate distance information to the feature pixel in the south, we add 1 to the y-axis integer component and update the integer component to (0,1). We propagate distance information to eight directions and retain the integer component pairs with the minimum real distance where the distance between a pixel to the nearest area boundary is computed as

$$| L(x,y) | = \sqrt{L_x(x,y)^2 + L_y(x,y)^2} \quad (2)$$

To make use of these properties, we process the grayscale recognition result into binary images with values 0 and 1 (0 represents the background pixel, and 1 represents the feature pixel) and then apply the 8SED-algorithm because the distance to the boundary pixel is independent of the model uncertainty value. For each pixel $p_i \in C$, we record the distance value γ_i, which can be used for reconstructing the original feature area in the later stage (Figure 7).

3.2.2 Quantifying Uncertainty in Reconstructing Pixels Areas from Centerline. One significant factor that affects the final conflation result is the thinning algorithm's quality for reconstructing the original object area in the image. With the help of distance map, we can generate the object area from the centerlines using the morphological dilation and represent the object area as the binary reconstruction map. For each pixel $p_i \in C$ with distance value γ_i, let the structuring element K_i be the squares with size $(\lfloor 2\gamma_i - 1 \rfloor) * (\lfloor 2\gamma_i - 1 \rfloor)$. We obtain the binary reconstruction map I_b (the object area) by morphological dilation

$$I_b = p_i \oplus K_i, \forall p_i \in C \quad (3)$$

Figure 9 shows a small region of binary reconstruction map compared with original image objects (recognition uncertainty map). However, as seen in the Figure 9, binary reconstruction map still has mismatches because the extracted skeletons are not always at the center of the linear feature. A small loss in the detail of the branches can incur significant distortion and lead to incorrect matching output later in conflation. To address this problem, we measure the number and location of mismatched pixels based on the comparison between the original images and the reconstruction map by four error values $d_{i1}, d_{i2}, d_{i3}, d_{i4}$. We have four direction values because once the algorithm has found the mismatched pixels, it searches the centerline pixels in four directions: north, south, east and west. We use d_{i1} to represent north, d_{i2} to represent south direction and etc. For notational convenience, we denote the set of pixels in the buffered image as I_b and the set of pixels in recognition result as I_s. We initialize the four direction values of every pixel in the centerline representation to zero. For each unmatched pixel q_i between buffered output and original image object, the algorithm searches centerline pixels along four directions and chooses the direction with shortest shift distance. The algorithm then updates the corresponded error value of the centerline pixel by the following equation

$$d_{ij} = \begin{cases} d_{ij} + 1 & \text{if } p_i \in I_s \cap \overline{I_b} \\ d_{ij} - 1 & \text{if } p_i \in \overline{I_s} \cap I_b \end{cases} \quad (4)$$

where $j = \in \{1, 2, 3, 4\}$ is the associated direction index and $\overline{I_b}$ is the background pixels in buffered output and $\overline{I_s}$ is the background pixels in the recognition result. Figure 10 is an example showing how the error value is recorded.

3.3 Data Integration using Uncertainty Estimates

This step describes that once the uncertainty values for centerlines are recorded, the procedure of using the uncertainty value to identify common line segments between the recognized centerlines and another vector dataset of the same feature but from a different source. Assuming one input is contemporary vector data of geographic features. The other input is the extracted centerline representation of the geographic features C from a map edition at

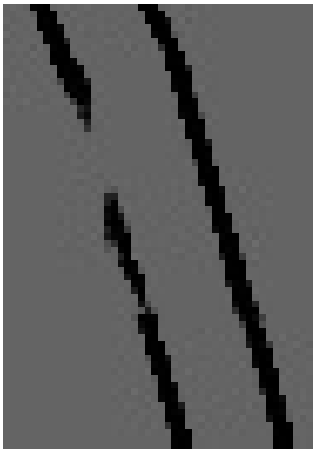

Figure 4: Original image file with irregular junctions.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Figure 5: The centerline representation of Figure 1a after applied thinning algorithm

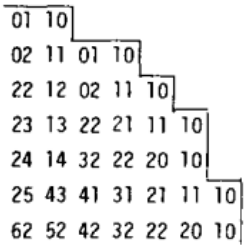

Figure 6: Distance map example. Pixels out side the black line are background pixels.

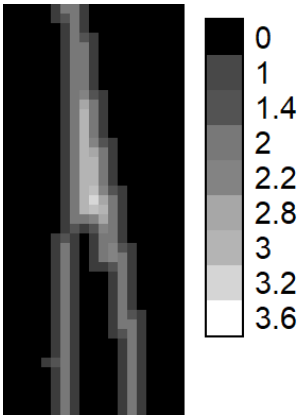

Figure 7: Distance map for part of the feature image

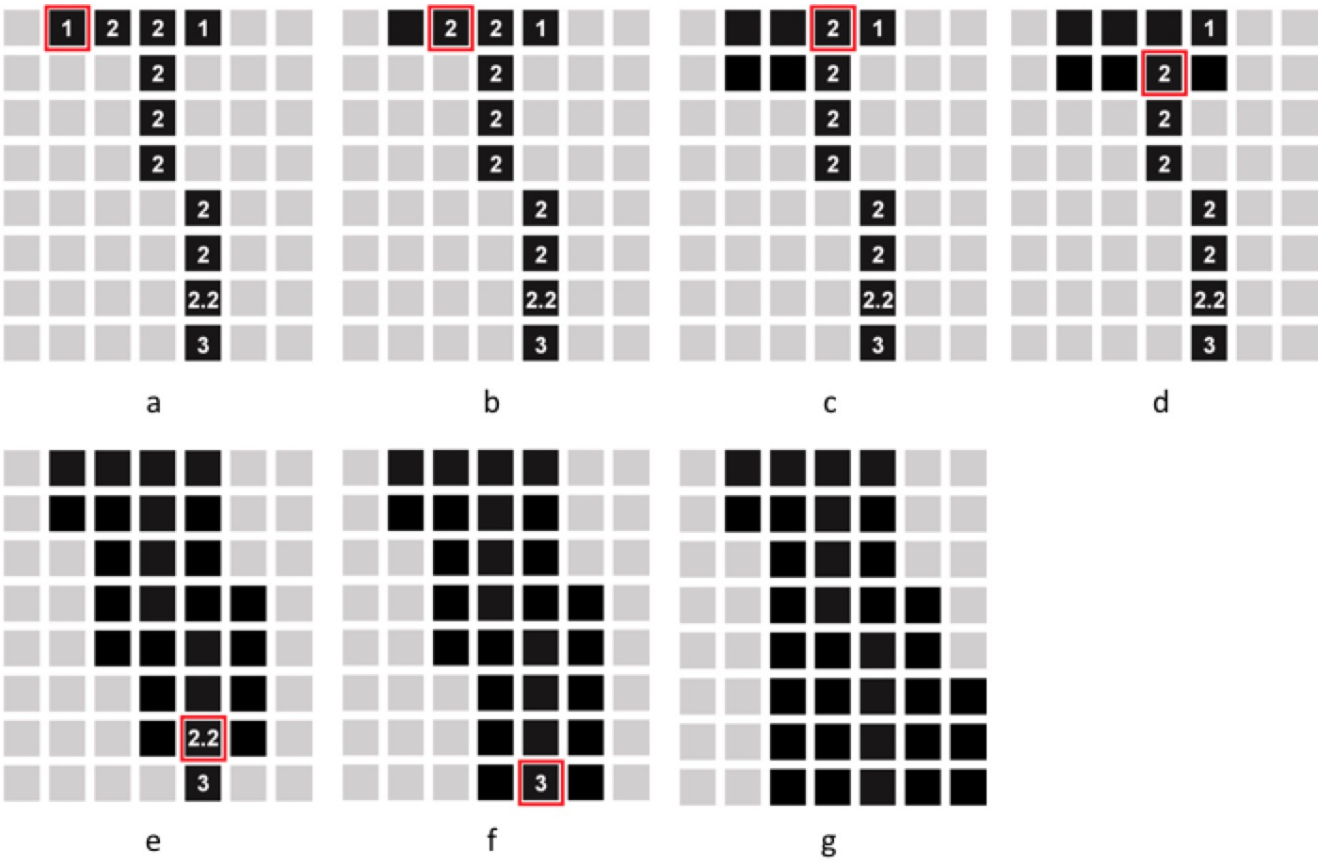

Figure 8: The reconstruction process of buffering a centerline by the distance values. The black squares are the foreground pixels, and the gray squares represent background pixels. The value on the centerline is the distance value. The pixel inside the red rectangle is the next pixel to be buffered.

a different time. Each pixel in the centerline representation[4] is associated with an uncertainty tuple $<\sigma_i, \gamma_i, d_{i1}, d_{i2}, d_{i3}, d_{i4}>$ where σ_i denotes recognition model uncertainty, γ_i denotes the distance value and d_{i1} to d_{i4} denotes the four-direction values. The output

[4]Here we assume the extracted centerlines are still in raster format for illustrating the idea. In practice, the extracted centerline is typically converted to vector format.

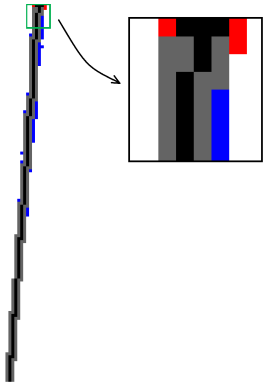

Figure 9: An example of comparing the extracted feature image with the buffered output using the distance map. The black pixels represent the centerlines using Guo-Hall algorithm. The gray pixels exist both in the extracted feature images and in buffered images. The red pixels exist in the extracted feature images but not in buffered images. The blue pixels exist in the buffered image but not in the extracted feature images.

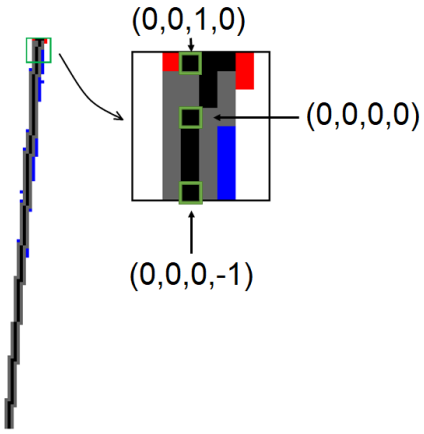

Figure 10: An example of the uncertainty values for three pixels.

is a "matchability" score $m(l_s) \in [0, 1]$ where l_s represents a line segment in vector data and the higher the score is, the less likely l_s has a correspondent entity in C. The procedure consists of two parts. First, we generate the probability image denoted as I_p by the uncertainty triple where each pixel value z_i represents the probability of current pixel in as the targe features. We then rasterize the contemporary vector data and overlay the rasterized results with I_p to identify the same entity by using the probability of overlapping area of each line segment.[5]

To generate the probability of each pixel, we first buffer the centerline representation C into the reconstruction map I_p using the distance value and the recognition uncertainty value. We buffer the centerline pixels by dilation as described in Section 3.3. We denote the set of pixels in the buffered result as A. However, since this dilation process is designed for the binary image with value 0 and 1, we also need to find a way to propagate the uncertainty value. If $p_i \in A \cap C$, the probability of p_i in the feature of interest is its associated model uncertainty value γ_i. If $p_i \in A \cap \overline{C}$, the probability is the maximum centerline pixels' recognition uncertainty that propagated through the morphological dilation. Therefore, we obtain the equation to calculate the probability of pixel $I_p(p_i)$ as follows

$$I_p(p_i) = \begin{cases} \gamma_i & \text{if } p_i \in A \cap \overline{I_b} \\ \max\{\sigma_1, \sigma_2 ... \sigma_n\} & \text{if } p_i \in A \cap \overline{C} \end{cases} \tag{5}$$

where $\sigma_1, \sigma_2 ... \sigma_n$ is the uncertainty of the set of centerline pixels q_i that satisfies $p_i \in q_i \oplus K_i$

Once the reconstruction map with uncertainty value is obtained, the algorithm adjusts the probability value of each pixel by the error values d_{i1} to d_{i4}. Assuming p_i is the current pixel, d_{ij} is the associated error value where $j \in 1, 2, 3, 4$ and $I_p(p_i$ is the probability value after buffering the centerlines. We move the point towards the corresponded direction to the boundary of reconstruction object area. If d_{ij} >0 which means there are $\mid d_{ij} \mid$ pixels missing from the reconstruction object area, the algorithm replaces the $\mid d_{ij} \mid$ pixels' probability along the direction from 0 to σ_i. Conversely, if $d_{ij} < 0$ the algorithm substitute the pixel's probability with 0.

Finally, we convert contemporary vector data into a rasterized image with two values 0 and 1 (0 if it is a background pixel). For each line segment with starting point (x_1, y_1) and end point (x_2, y_2), we crop the rectangle with four corner points four points $(\min\{x_1, x_2\}, \min\{y_1, y_2\})$, $(\max\{x_1, x_2\}, \min\{y_1, y_2\})$, $(\min\{x_1, x_2\}, \max\{y_1, y_2\})$, $(\max\{x_1, x_2\}, \max\{y_1, y_2\})$ on both the probability image and rasterized image to include the geographic feature in the sub-regions. We denote that the cropped sub-images of probability image and raster image as P_{sub} and R_{sub}, respectively. We calculate the probability score of the line segment l_s vector data $m(l_s)$ by the following equations.

$$m(l_s) = \frac{\sum I_p(q_i \in l_s)}{L(l_s)} \tag{6}$$

where $I_p(q_i)$ represents the probability value of each pixel q_i in the line segment l_s and $L(l_s)$ represents the length of line segment l_s. If $m(l_s)$ is greater than a tunable parameter T_{sim}, we consider that this line segment does not have a corresponded entity on the map layer.

4 PRELIMINARY EXPERIMENT RESULT

We evaluated our approach using railroad data from two geographic datasets: a USGS historical topography map, Bray in California (circa 2001) and US Census railroad vector data (circa 2016). Our goal is to show that the uncertainty evaluation algorithm described in Section 3 can improve the effectiveness of the geographic feature conflation process and identify corresponding features from the two datasets.

In the experiment, we first ran the FCN model to extract railroads from the map and used 1) our proposed method and 2) a baseline method to integrate the extracted railroads with the US Census railroad vector data. We call it a match if either our approach or

[5]In practice, this conflation process is usually performed using vector data of the two datasets. The rasterization process here is to illustrate the idea.

the baseline method could correctly identify whether the railroad from vector data has an equivalent entity on the map. We call it a wrong match either 1) the railroad from vector data doesn't exist on the maps but still matches to an entity in the extracted railroads or 2) one line segment of railroad maps to two entities on the map. We manually checked whether the matched entity was correct. We calculated the accuracy based on the length of the correctly matched line segments divided by the overall line segments length in the vector data. For each line segment l_s, we used Euclidean distance $||l_s||_2$ in the image coordinates as its vector length.

To test our overall approach, we ran our algorithm on the entire image of the extraction output (i.e., the probability image of whether or not a pixel belongs to railroads) using T_{sim} =0.5. If the similarity between two line segments from different sources is greater than 0.5, we considered that these two line segments did not map to the same entity. For the baseline method, we simulated a traditional conflation process. After we obtained the recognition result from the recognition model, we filtered pixels with an uncertainty value greater than 0.5 and the remaining pixels were classified as the railroad pixels because tradition classification algorithms ignore pixels with uncertainty value higher than 0.5 to decrease false positive. We then ran the thinning algorithm and buffered the centerline with buffer size p=3-pixel, p=5-pixel and p=7-pixel. For fairness, we also applied T_{sim} =0.5 to evaluate the vector data similarity. Since the buffered outputs were binary images, the baseline method considered the line segment had an equivalent entity on the map if the similarity value was lower than 0.5 which means half of its pixels were matched with the recognition result.

Overall we have 309 line segments with a total length 24,158 pixels. Figure 11 shows the experiment results. The x-axis of Figure 11 represents the buffer size, the y-axis of Figure 11 shows the accuracy. The red line presents the accuracy of our uncertainty aware approach, which does not require a manually set buffer size.

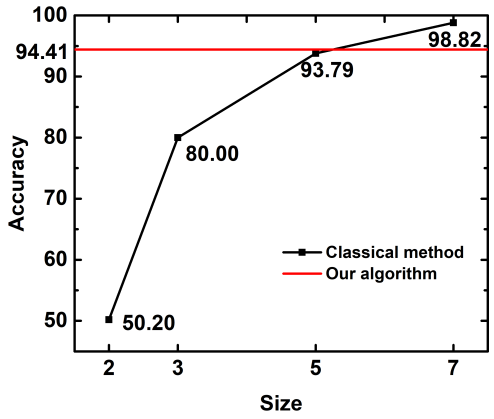

Figure 11: The accuracy chart of the results from the baseline method and our approach. The red line is the conflation accuracy of our approach. The black line shows the conflation accuracy of using different fixed buffer sizes in the baseline method.

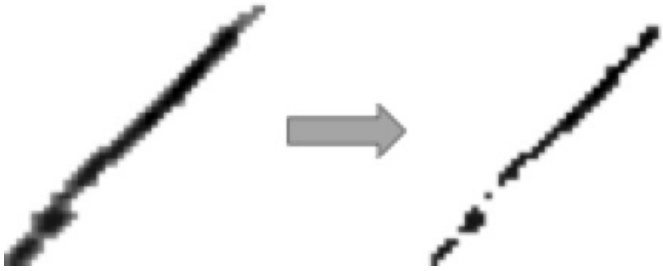

Figure 12: Left: A line segment of extracted railroads on the recognition uncertainty map. Right: The filtered output of the left image. Pixels with an uncertainty value greater than 0.5 were filtered out. Therefore, the baseline method that using a fixed buffer size with 2-pixel could not generate the correct conflation result.

We can see the performance of our algorithm compared with the method using a fixed buffer size. We obtained 44.21% improvement when the buffer size p = 2-pixel, 14.41% improvement when the buffer size p = 3-pixel and 0.62% improvement when the buffer size p = 5-pixel. The accuracy increased as the buffer size increased for the baseline method. When the buffer size was greater than 3-pixel (the width of railroads on the maps), the upward trend of accuracy decreased.

For the baseline method, when the buffer width is too small, there are not enough pixels to be compared with the line segments from the contemporary vector data. Figure 12 shows the example that our algorithm correctly identified the corresponding line segment on the map while the classic method using the buffer size of 2-pixel could not. Since the recognition model incorrectly extracted railroad pixels, pixels in noisy areas were predicted with high uncertainty and filtered out in the first step in the baseline method. In contrast, with the help of retaining the recognition uncertainty, our approach successfully determined a match between the railroads on the map and the contemporary data.

For the baseline method, when the buffer size increases, there is a higher chance for the pixel from the two sources to be matched. However, increasing the buffer width could introduce incorrect matches because a high number of the background pixels would be included in the buffered images. For example, when the buffer width is too large, two separate line segments could overlap with each other and thus one line segment could match with two entities in the map (Figure 13). In our experiment, because the railroads are very sparse, increasing buffer size to 7-pixel (more than twice as wide as the railroads) still does not decrease the conflation accuracy. In practice, the geographic features could align close with each other, and the user needs to manually set the buffer width for each of the input datasets and even for each line segment. Hence the baseline method does not scale well to process a large number of datasets (e.g., integrating thousands of historical maps with various scan resolutions and map scales).

For both our approach and the baseline method, if the FCN model can not extract the correct features, we always obtained an incorrect integration result. For example, in Figure 14 because the background was noisy, there was a gap in the recognition result. Although

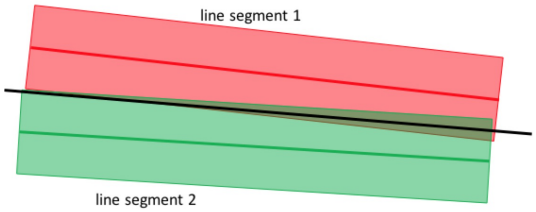

Figure 13: The black line represents the railroad from contemporary vector data. When the buffer width is too large, this part of the railroads map to two entities in the map.

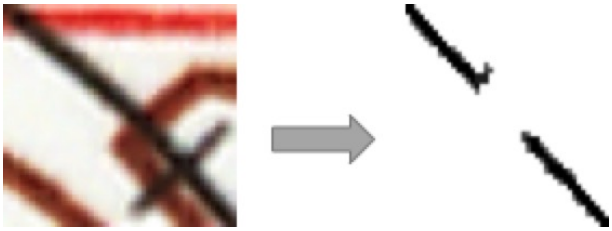

Figure 14: Left: Part of the map layer of Bray in California circa 2001 junctions. The FCN model cannot correct extract the railroads in this area due to the noisy background. Right: The recognition output of the left image. The is a huge gap in the middle of the predicted line segment.

our algorithm was able to combine uncertainty information from pixels, the recognition model gave poorly extracted features at some locations, so our algorithm could not correctly match the vector data. In this case, because there was a huge gap in the middle of the predicted railroads, the uncertainty for this line segment was over 0.6. One potential solution is to use semantic knowledge of the geographic features. If there is a small gap between two lines on the recognition images, we could calculate the in-between values based on a spline curve model to connect the lines.

5 DISCUSSION AND FUTURE WORK

We presented an approach to quantify the uncertainty during the recognition and thinning processes as well as a vector-to-vector conflation process using the uncertainty values. Future work includes solving the error mentioned in Section 4. One possible solution is to use a spline model with neighborhood pixel uncertainty to globally adjust the pixel uncertainty. We also plan to conduct extensive experiments to test the proposed method with more types of geographic features and sources.

ACKNOWLEDGMENTS

This research is supported in part by the National Endowment for the Humanities (Grant No.: NEH PR-253386-17), in part by the USC Undergraduate Research Associates Program, and in part by the National Science Foundation under Grant No. IIS 1564164 (to the university of Southern California). We gratefully acknowledge the support of Microsoft Corporation with the Azure for Research Award and NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Cagri Aslan, Aykut Erdem, Erkut Erdem, and Sibel Tari. 2008. Disconnected skeleton: shape at its absolute scale. *IEEE Transactions on pattern analysis and machine intelligence* 30, 12 (2008), 2188–2203.

[2] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. 2009. Stability and computation of medial axes-a state-of-the-art report. In *Mathematical foundations of scientific visualization, computer graphics, and massive data exploration*. Springer, 109–125.

[3] Vassilios Chatzis and Ioannis Pitas. 2000. A generalized fuzzy mathematical morphology and its application in robust 2-D and 3-D object representation. *IEEE Transactions on Image Processing* 9, 10 (2000), 1798–1810.

[4] Ching-Chien Chen, Craig A Knoblock, Cyrus Shahabi, Yao-Yi Chiang, and Snehal Thakkar. 2004. Automatically and accurately conflating orthoimagery and street maps. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems*. 47–56.

[5] Yao-Yi Chiang. 2016. Unlocking Textual Content from Historical Maps-Potentials and Applications, Trends, and Outlooks. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*. Springer, 111–124.

[6] Per-Erik Danielsson. 1980. Euclidean distance mapping. *Computer Graphics and image processing* 14, 3 (1980), 227–248.

[7] Tamal K Dey and Wulue Zhao. 2004. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica* 38, 1 (2004), 179–200.

[8] Weiwei Duan, Johannes H. Uh Yao-Yi Chiang, Crang A Knoblock, and Stefan Leyk. 2018. Automatic generation of precisely delineated geographic features from georeferenced historical maps using deep learning. In *Proceedings of the AutoCarto*.

[9] Alessandro Farina, Zsolt M Kovacs-Vajna, and Alberto Leone. 1999. Fingerprint minutiae extraction from skeletonized binary images. *Pattern recognition* 32, 5 (1999), 877–889.

[10] Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. 2009. The scale axis transform. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*. 106–115.

[11] Zicheng Guo and Richard W Hall. 1989. Parallel thinning with two-subiteration algorithms. *Commun. ACM* 32, 3 (1989), 359–373.

[12] Louisa Lam, Seong-Whan Lee, and Ching Y Suen. 1992. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence* 14, 9 (1992), 869–885.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[14] Patrick Lüscher, Dirk Burghardt, and Robert Weibel. 2007. Matching road data of scales with an order of magnitude difference. In *Proceeding of XXIII International Cartographic Conference*.

[15] Theo Pavlidis. 1980. A thinning algorithm for discrete binary images. *Computer graphics and image processing* 13, 2 (1980), 142–157.

[16] Juan J Ruiz, F Javier Ariza, Manuel A Urena, and Elidia B Blázquez. 2011. Digital map conflation: a review of the process and a proposal for classification. *International Journal of Geographical Information Science* 25, 9 (2011), 1439–1466.

[17] Stefan Steiniger and Erwan Bocher. 2009. An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science* 23, 10 (2009), 1345–1370.

[18] E Lynn Usery, Michael P Finn, and Michael Starbuck. 2009. Data layer integration for the national map of the United States. *Cartographic Perspectives* 62 (2009), 28–41.

[19] Steffen Volz. 2006. An iterative approach for matching multiple representations of street data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36, Part 2/W40 (2006), 101–110.

[20] Xiaqing Wu, Rodrigo Carceroni, Hui Fang, Steve Zelinka, and Andrew Kirmse. 2007. Automatic alignment of large-scale aerial rasters to road-maps. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. 17.