

Automatically and Efficiently Matching Road Networks with Spatial Attributes in Unknown Geometry Systems

Ching-Chien Chen

Geosemble Technologies
2041 Rosecrans Ave, Suite 245
El Segundo, CA 90245
jchen@geosemble.com

Cyrus Shahabi

Craig A. Knoblock
University of Southern California
Department of Computer Science
Los Angeles, CA 90089
shahabi, knoblock@usc.edu

Mohammad Kolahdouzan*

Yahoo! Search Marketing
Burbank, CA
mohammad_ysm@yahoo.com

Abstract

Vast amount of geospatial datasets are now available through numerous public and private organizations. These datasets usually cover different areas, have different accuracy and level of details, and are usually provided in the vector data format, where the latitude and longitude of each object is clearly specified. However, there are scenarios in which the spatial attributes of the objects are intentionally transformed to a different, and usually unknown, (alien) system. Moreover, it is possible that the datasets were generated from a legacy system or are represented in a native coordinate system. An example of this scenario is when a very accurate vector data representing the road network of a portion of a country is obtained with unknown coordinate. In this paper, we propose a solution that can efficiently and accurately find the area that is covered by this vector data simply by matching it with the (possibly inaccurate and abstract) data with known geocoordinates. In particular, we focus on vector datasets that represent road networks and our approach identifies the exact location of the vector dataset of alien system by comparing the distribution of the detected road intersection points between two datasets. Our experiment results show that our technique can match road vector datasets that are

composed of thousands of arcs in a relatively short time with 91% precision and 92.5% recall for the matched road feature points.

1. Introduction

With the rapid improvement of geospatial data collection techniques, the growth of Internet and the implementation of Open GIS, a large amount of geospatial data are now readily available on the web. The examples of well-known vector datasets are US Census TIGER/Line files¹ (covering most roads over the United States), NAVSTREETS from NAVTEQ,² VPF data from NGA (U.S. National Geospatial-Intelligence Agency),³ and DLG data from USGS (U.S. Geological Survey).⁴ The Yahoo Map Service,⁵ Google Map Service,⁶ Microsoft TerraService⁷ [1] are good examples of map or satellite imagery repositories. These datasets usually cover different areas, have different accuracy and level of details, and some of them are provided in the vector data format, where the latitude and longitude of each vector object is clearly specified. However, there are scenarios in which the spatial attributes of the vector objects are intentionally transformed to a different, and usually unknown, (alien) system. Moreover, it is also possible that the datasets were generated from a legacy system or are represented in a native coordinate system.

Figure 1 illustrates a scenario where we want to locate the area of a USGS raster topographic map (as shown in

* This work was done when the author was working at University of Southern California as a Post-Doc Research Associate.

¹ <http://www.census.gov/geo/www/tiger/>

² <http://www.navteq.com/>

³ <http://www.nga.mil/>

⁴ <http://tahoe.usgs.gov/DLG.html>

⁵ <http://maps.yahoo.com/>

⁶ <http://maps.google.com>

⁷ <http://terra-server-usa.com/>

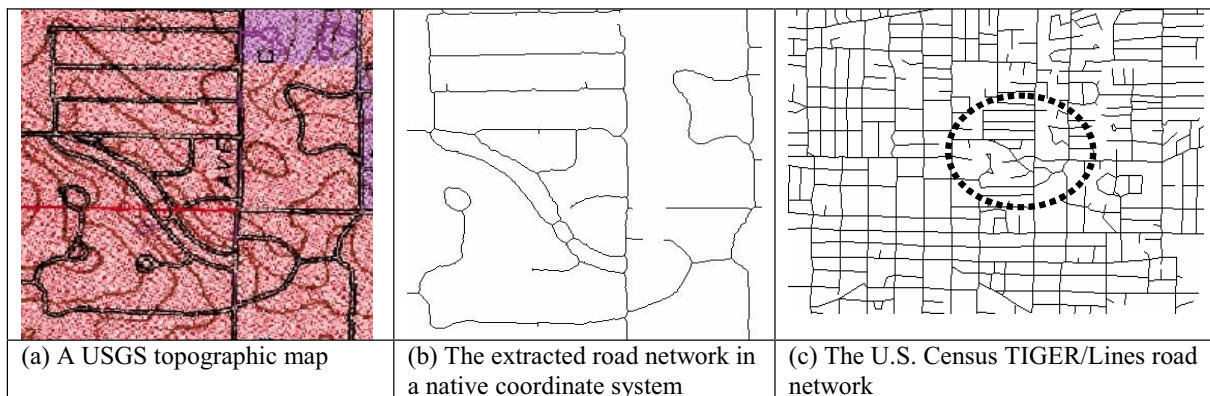


Figure 1: Two road networks cover overlapping areas

Figure 1(a). The map covers partial area of St. Louis County, MO in U.S.A., but its exact geocoordinates are inaccessible. The map is first processed to extract road network (as shown in Figure 1(b)) by using a text/graphics separation technique developed by us [2]. Obviously, the extracted road network is in a native coordinate system (i.e., raster pixel x/y in Cartesian system). Meanwhile, a larger road network covering this county is publicly available from US Census TIGER/Lines. By identifying matched features between the two road networks, the system can then automatically infer the geocoordinate of the extracted map road network (as the area highlighted in Figure 1(c)).

There have been a number of efforts to automatically or semi-automatically detect matched features across different road vector datasets [3, 4, 5, 6, 7]. Given a feature point from one dataset, these approaches utilize different matching strategies to discover the corresponding point within a predetermined distance (i.e., a localized area). This implies that these existing algorithms only handle the matching of vector datasets in the same geometry system (i.e., the same coordinate system⁸). Hence, to the best of our knowledge, no general method exists to resolve the matching of two vector data in unknown geometry systems. Furthermore, processing large vector datasets often requires significant CPU time. Our methodology, described in this paper, is able to automatically and efficiently handle the matching of diverse and potentially large vector datasets, independent of the coordinate system used. In particular, we focus on vector datasets that represent road networks.

The basic idea of our approach is to find the transformation T between the layout (with relative distances) of the feature point set on one road network and the corresponding feature point set on the other road network. This transformation achieves global matching between two feature point sets by locating the common point pattern among them. More precisely, the system can detect feature points from both road networks. The

distribution of detected feature points from each road network forms a particular (and probably unique) point pattern for the road network. In order to improve the running time, our approach exploits auxiliary spatial information to reduce the search space for the transformation T . Once the matched points across different road networks are identified, the system can then utilize this transformation to map the road network of alien geometry (coordinate) system into a known coordinate system (e.g., geodetic coordinate system).

To illustrate the usefulness of our approach, consider the matching of two road networks: one is in an unknown coordinate system but with more accurate geometry and the other has rich attributes and a known coordinate system but with poor geometry. Applying our matching algorithm to these two road networks can result in a superior road network that combines the accuracy of the road geometry from one vector dataset and rich attributes from the other. Furthermore, these matched points can be used as control points to conflate these two road network datasets [6].

The remainder of this paper is organized as follows. Section 2 describes our approach in details. Section 3 provides experimental results. Section 4 discusses the related work and Section 5 concludes the paper by discussing our future plans.

2. Proposed Approach

In this section, we first describe our overall approach to match two road networks. Then, we describe the details of our techniques.

2.1 Approach Overview

Intuitively, matching road networks relies on the process of matching the road segments from two vector datasets to find the corresponding road segments. However, this is a challenging task for two large road networks, especially when one of the road networks is in a different or unknown geometry system. To address this issue, we propose to match two datasets based on some feature points detected from the road networks. In particular, we

⁸ In this paper, we use the terms “*geometry system*” and “*coordinate system*” interchangeably.

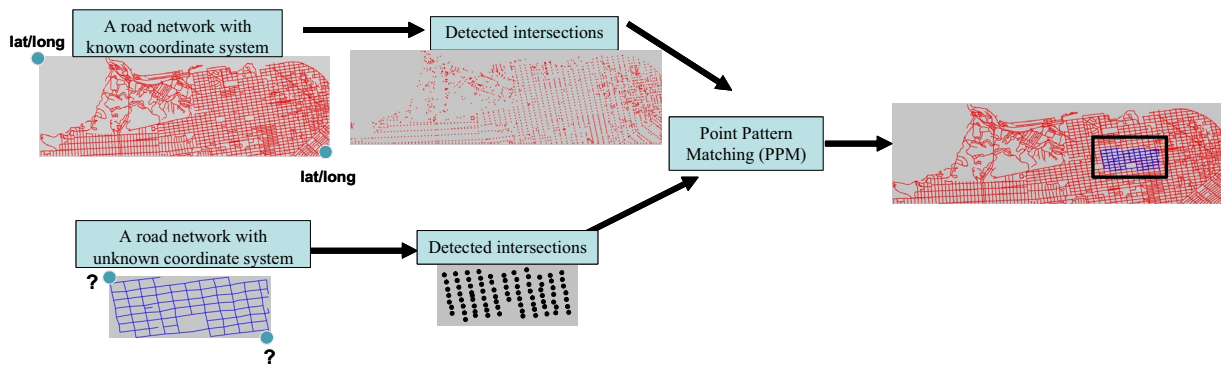


Figure 2: The overall approach

utilize road intersections as the feature points. Road intersections are good candidates for being matched, because road intersections are salient points to capture the major features of the road network and the road shapes around intersections are often well-defined. In addition, various GIS and computer vision researchers have shown that the intersection points on the road networks are good candidates to be identified as an accurate set of matched points [7, 8, 9, 10].

After detecting a set of intersection points from each road network separately, the remaining problem is how to match these intersection points effectively and efficiently to locate a common distribution (or pattern) from these intersections. Our system perceives the distribution of detected intersections from each road network as the *fingerprnt* of the road network. Then our system finds the transformation T between the layout (with relative distances) of the feature point set on one road network data and the feature point set on the other road network. This transformation achieves global alignment between two intersection point sets by locating the common point pattern among them.

Figure 2 shows our overall approach. Using detected road intersections as input, the system locates the common point pattern across these two point sets by computing a proper transformation between them. The system can then utilize this transformation to map the road network in unknown geometry system into a known coordinate system. We describe our detailed techniques in the following sections.

2.2 Finding the feature points from vector datasets

The technique to detect road intersections from road network relies on the underlying road vector data representation. Typically, there are two common ways to represent the geometry of a road vector dataset: (1). The road network is composed of multiple road segments (polylines), and the line segments are split at intersections (as the example shown in Figure 3(a)). (2). The road network is composed of multiple road segments

(polylines), but the line segments are *not* split (if not necessary) at intersections (see Figure 3(b)).

This generation of our system focus on handling vector data represented in the first way (i.e., the line segments are split at intersections), because most of the popular road vector datasets (such as US Census TIGER/Line files and NAVSTREETS from NAVTEQ) represent their datasets in such way. Based on this sort of road segment representation, the process of finding the intersection points from the road network is divided into two steps. First, the system examines all line segments in the vector data to label the endpoints of each segment as the candidate intersection points. Second, the system examines the connectivity of these candidate points to determine if they are intersection points. In this step, each candidate point is verified to see if there are more than two line segments connected at this point. If so, this point is marked as an intersection point and the directions of the segments that are connected at the intersection point are calculated. In practice, the search of road intersections from large road networks is supported efficiently by spatial access method R-tree [11].

2.3 Finding the matched feature points by Point Pattern Matching (PPM)

Now that we have described how to detect feature points from a road network, we now describe how this

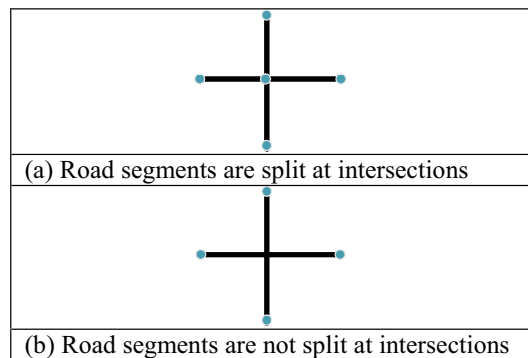


Figure 3: Different ways to represent a cross-shaped road network with one intersection

information can be used to automatically match two road networks. Let $U = \{u_i \mid u_i = (x_i, y_i)\}$, where (x_i, y_i) is the location of intersections of the first road network and $V = \{v_j \mid v_j = (m_j, n_j)\}$, where (m_j, n_j) is the location of intersections of the second road network. Our objective is to locate the set: $\{Rel_{pat} = \{(u_i, v_j) \mid \text{where } u_i \text{ is the intersection on the first vector dataset and } v_j \text{ is the corresponding intersection (if any) on the second dataset. That is, point } u_i \text{ and } v_j \text{ are formed by the same intersected road segments}\}$. Consider identifying matched point pattern between two road networks. If the system can recognize the names of road segments that meet at intersections, it can use these road names to infer the set Rel_{pat} . However, road vector data may not include the non-spatial attribute, road name. Instead, we propose our approach that relies on some prominent geometric information, such as the distribution of points, the degree of each point and the direction of incident road segments, to locate the matched point pattern. In other words, the problem of point pattern matching is at its core a geometric point sets matching problem. The basic idea is to find the transformation T between the layout (with relative distances) of the point set U and V .

The key computation of matching the two sets of points is calculating a proper transformation T , which is a 2D rigid motion (rotation and translation) with scaling. Because the majority of vector datasets are oriented such that north is up, we only compute the translation transformation with scaling. Without loss of generality, we consider how to compute the transformation where we map from a fraction α of the points of U to the points of V . The reason that only a fraction α of the points of U is considered is that one road vector dataset could be detailed while the other one is represented abstractly or there may be some missing/noisy points from each road network. The transformation T brings at least a fraction α of the points of U into a subset of V . This implies:

$\exists T$ and $U' \subseteq U$, such that $T(U') \subseteq V$, where $|U'| \geq \alpha|U|$ and $T(U')$ denotes the set of points that results from applying T to the points of U' . Or equivalently, for a 2D point (x, y) in the point set $U' \subseteq U$, $\exists T$ in the matrix form

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ T_x & T_y & 1 \end{bmatrix} \quad (S_x \text{ and } S_y \text{ are scale factors along } x \text{ and } y$$

direction, respectively, while T_x and T_y are translation factors along x and y directions, respectively), such that

$$[x, y, 1]^* \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ T_x & T_y & 1 \end{bmatrix} = [m, n, 1], \text{ where } |U'| \geq$$

$\alpha|U|$ and the 2D point (m, n) belongs to the intersection point set V on the second vector dataset. With this setting, we do not expect point coordinates to match exactly because of finite-precision computation or small errors in

the datasets. Therefore, when checking whether a 2D point p belongs to the point set V , we declare that $p \in V$, if there exists a point in V that is within Euclidean distance δ of p for a small fixed positive constant δ , which controls the degree of inaccuracy. The minimum δ such that there is a match for U' in V is called *Hausdorff distance*. Different computations of the minimum *Hausdorff distance* have been studied in great depth in the computational geometry literature [12]. We do not seek to minimize δ but rather adopt an acceptable threshold for δ . The threshold is relatively small compared to the average inter-point distances in V . In fact, this sort of problem was categorized as ‘‘Nearly Exact’’ point matching problem in [13].

Given the parameters α and δ to obtain a proper transformation T , we need to compute the values of the four unknown parameters S_x, S_y, T_x and T_y . This implies that at least four different equations are required. A straightforward (brute-force) method is first choosing a point pair (x_1, y_1) and (x_2, y_2) from U , then, for every pair of distinct points (m_1, n_1) and (m_2, n_2) in V , the transformation T' that map the point pair on U to the point pair on V is computed by solving the following four equations:

$$\begin{aligned} S_x * x_1 + T_x &= m_1 & S_y * y_1 + T_y &= n_1 \\ S_x * x_2 + T_x &= m_2 & S_y * y_2 + T_y &= n_2 \end{aligned}$$

Each generated transformation T' is thus applied to the entire points in U to check whether there are more than $\alpha|U|$ points that can be aligned with some points on V within the threshold δ . This process is repeated for each possible point pair from U , which implies that it could require examining $O(|U|^2)$ pairs in the worst case. Since for each such pair, the algorithm needs to try all possible point pairs on V (i.e., $O(|V|^2)$) and spends $O(|U| \log|V|)$ time to examine the generated transformation T' , this method has a worst case running time of $O(|U|^3 |V|^2 \log|V|)$. The advantage of this approach is that we can find a mapping (if the mapping exists) with a proper threshold δ even in the presence of very noisy data. However, it suffers from high computation time. One way to improve the efficiency of the algorithm is to utilize randomization in choosing the pair of points from U as proposed in [14], thus achieving the running time of $O(|V|^2 |U| \log|V|)$. However, their approach is not appropriate for our datasets because it is possible one vector dataset is in detailed level while other vector dataset is represented abstractly.

In fact, in our previous work [15], we utilized the similar technique to match two point sets detected from a raster map and an image. More precisely, in [15], we proposed an enhanced point pattern matching algorithm to find the overlapping area of a map and an imagery by utilizing map-scale to prune the search space of possible point pattern matches (by reducing the numbers of potential matching point pairs needed to be examined). In the following sections, we focus on finding the matching

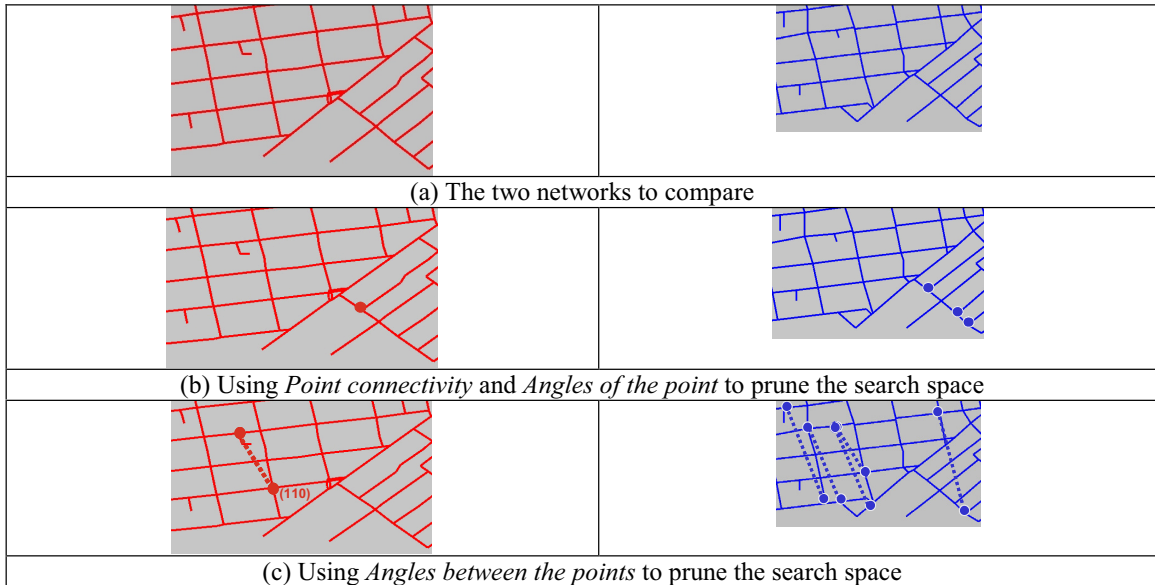


Figure 4: Comparing two road networks by using Geo-PPM

between different road networks and developing more efficient techniques by utilizing some additional spatial information that can be inferred from the road vector datasets. In addition, we also discuss how to prioritize the potential matching point pairs needed to be examined.

2.4 Enhanced PPM Algorithm: Prioritized Geo-PPM

Due to the poor performance of the brute-force point pattern matching algorithm mentioned in the previous section, PPM cannot be applied to large datasets where the number of points (or intersections) is in the order of thousands (such as the road networks covering large areas). Consequently, we utilize some auxiliary information that can be extracted from the road vector data to improve the performance of PPM for larger road networks. With the goal to reduce the numbers of potential matching point pairs needed to be examined, the intuition here is to exclude all unlikely matching point pairs. For example, given a point pair (x_1, y_1) and (x_2, y_2) in S_1 , we only need to consider pairs (x'_1, y'_1) and (x'_2, y'_2) in S_2 as candidate pairs such that the real world distance and angle between (x_1, y_1) and (x_2, y_2) is close to the real world distance and angle between (x'_1, y'_1) and (x'_2, y'_2) . In addition, (x'_1, y'_1) would be considered as a possible matching point for (x_1, y_1) if and only if they have similar connectivity and road directions. We categorize the auxiliary information we utilize to the following groups.

1. Point connectivity: We define the connectivity of a point as the number of the road segments that intersect at that point. Clearly, if datasets S_1 and S_2 have very close densities (i.e., number of intersections per one unit of area), a candidate matching point P'_1 in S_2 for a point P_1 in S_1 must have the same connectivity as P_1 . Note that if the densities of the datasets are different (i.e., one dataset

is detailed and the other one is represented abstractly), this condition will not be valid for a large portion of the intersections and may only be valid for major roads' intersections.

2. Angles of the point: The angles of a point are defined as the angles of the road segments that intersect at that point. Similar to the connectivity, a point P'_1 in S_2 can only be considered as a candidate for point P_1 in S_1 only if the two points have similar angles, or the difference between their angles is less than a threshold value. To illustrate, consider comparing two road networks as the example shown in Figure 4(a). Whenever the system chooses a point (as the point shown in the left figure of Figure 4(b)) in one road network, it only has to consider the candidate matched points with same connectivity and similar directions of intersected road segments from the other network (as some possible candidates marked in the right figure of Figure 4(b)). Note that if the densities of the datasets are different (i.e., one dataset is detailed and the other one is represented abstractly), this condition will not be valid for a large portion of the intersections and may only be valid for major roads' intersections.

3. Angle between the points: The angle between two points is defined as the angle of the straight line that connects the points. Clearly, a pair (P'_1, P'_2) can be considered as a possible candidate for the pair (P_1, P_2) only if the angle between P'_1 and P'_2 is similar to the angle between P_1 and P_2 , or the difference between their angles is less than a threshold value. Note that this feature can only be utilized when the second dataset is not rotated and has the same direction as the first dataset. Consider the example shown in Figure 4(c). Whenever the system chooses a point pair (as the point pair shown in the left figure of Figure 4(c) and the angle between these two points is about 110 degree) in one road network, it only has to consider the candidate matched point pairs with the

similar angle (as some possible candidate point pairs marked as dash lines in the right figure of Figure 4(c)).

4. Distance between the points: The distance between two points is defined as the length of the straight line that connects the points in Euclidean space. Similar to the previous case, a pair (P'_1, P'_2) can be considered as a possible candidate for the pair (P_1, P_2) only if the length of the line connecting P'_1 and P'_2 is similar to the length of the line connecting P_1 and P_2 , or the difference between the lengths is less than a threshold value. Note that this feature can only be utilized when the relationship between the geometry of the datasets is known and hence, the distances between objects in two datasets are comparable.

By applying the above conditions simultaneously, the Geo-PPM approach can be defined as a specialization of PPM where only the candidate pairs that have similar point connectivity, angles of the point, angles between the points, and distances between the points, will be considered. This will greatly reduce the size of the search space. However, this is still a very complex approach when the number of points in the datasets is in the order of thousands. Hence, we propose prioritized Geo-PPM that can dramatically reduce the complexity of Geo-PPM for large networks by examining the points that have the minimum number of candidates.

Prioritized Geo-PPM

The intuition behind prioritized Geo-PPM is to increase the possibility of examining the correct matching pair from the candidates by first examining the pairs of points that have the minimum number of candidates. Suppose that there are n_1, n_2, n_3 and n_4 points in the pool of candidates for points $P_1, P_2, P_3,$ and $P_4,$ respectively. This means that the number of possible candidate pairs for (P_1, P_2) and (P_3, P_4) that must be examined by Geo-PPM is n_1n_2 and $n_3n_4,$ respectively. Note that the values of n_1 to n_4 could be very large, especially for urban areas where the road networks follow a grid pattern and hence, a large portion of the intersections have the same connectivity and angles. Also note that from these possible candidate pairs, only (a maximum of) one pair is the correctly matching one. Hence, by first examining the combination that contains the minimum number of points, we can

significantly increase the possibility of finding the correct matching pair sooner. Consider the example shown in Figure 5. Our system can start the matching process by first examining the combination that contains the minimum number of points. As the point pair chosen in the left figure of Figure 5(b), it has less potential matching point pairs as shown in the right figure of Figure 5(b), comparing to the point pair examined in the left figure of Figure 5(a).

3. Evaluations

We performed several experiments with real world datasets to examine the performance of our prioritized Geo-PPM. We used three road networks obtained from USGS, NGA and US Census, covering the streets in the area of $(-122.5015, 37.78)$ to $(-122.3997, 37.8111)$. Figure 6(a) shows USGS road network with accurate geometry but with poor attributes. Figure 6(b) shows the US Census TIGER/Lines road network with rich attributes (e.g., road names, road classifications) but with poor geometry. Figure 6(c) shows the NGA road network with some specific attributes (e.g., road surface type). Also note that, as shown in the figure, while the data from USGS and US Census have almost similar granularity, the NGA data is an abstract level data (i.e., only major roads are stored). We manually transformed each dataset to unknown geometry systems by multiplying and subsequently adding different values to latitudes and longitudes of the vector objects in each dataset. Moreover, we filtered the south west quarter of the datasets to generate datasets with smaller sizes to examine how our approach behaves for different sizes of data.

Figure 7 shows the partial result of matched feature point sets for the USGS and US Census road networks. We also performed a quantitative analysis to measure the performance of our approach. Toward that end, we developed two metrics, precision and recall, to measure the performance of our Geo-PPM technique, since the accuracy of the matched points significantly affects the matching of the two road networks. Let the point pattern generated by Geo-PPM be defined as a set:

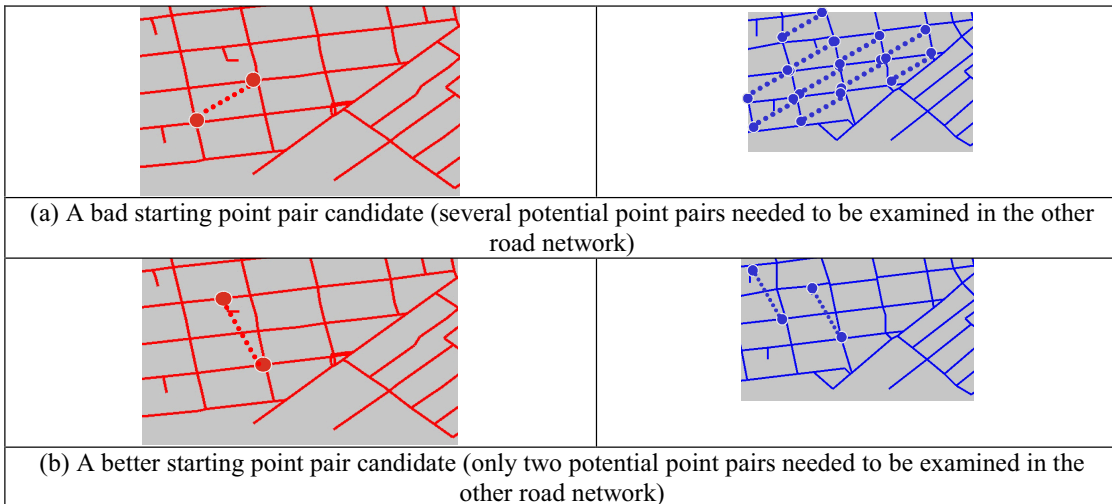


Figure 5: Picking up proper point pair by using Prioritized Geo-PPM

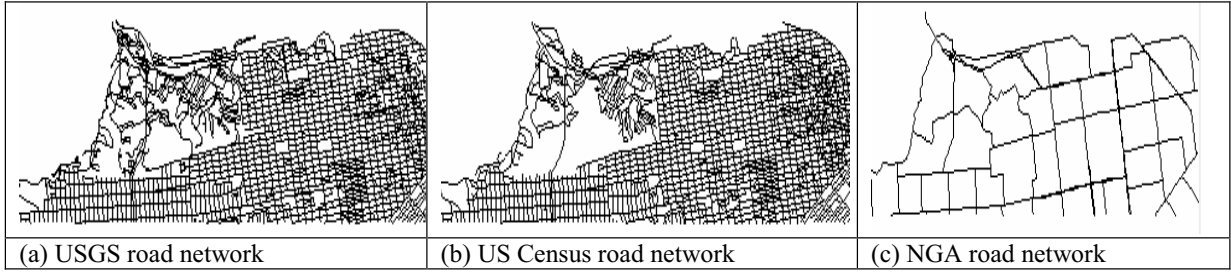


Figure 6: Different road networks used in the experiments

$Ret_{pat} = \{(m_i, s_j) \mid \text{where } m_i \text{ is the intersections on the first vector dataset and } s_j \text{ is the corresponding intersections located by prioritized Geo-PPM}\}$

To measure the performance of Geo-PPM, we need to compare the set Ret_{pat} with respect to the real matched point pattern set Rel_{pat} (defined in Section 2.3).

Using this term, we define

$$\text{Precision} = \frac{|Ret_{pat} \cap Rel_{pat}|}{|Ret_{pat}|}$$

$$\text{Recall} = \frac{|Ret_{pat} \cap Rel_{pat}|}{|Rel_{pat}|}$$

Intuitively, precision is the percentage of correctly matched road intersections with respect to the total matched intersections detected by prioritized Geo-PPM. Recall is the percentage of the correctly matched road intersections with respect to the actual matched intersections. Table 1 shows the results of our experiments for three combinations of these datasets. As shown in the table, the average number of candidates (i.e., the number of points in the second dataset with the same connectivity and angles as compared to a point in the first dataset) varies between 371 and 637. This shows that the possibility of selecting 2 pairs from the candidate pool which are exactly matched to 2 points selected from the first dataset is very low, meaning that random selection of points in Geo-PPM will result to a very large number of possibilities and hence, to a very large processing time. For example, for the USGS+US Census combination, the possibility that randomly selected pair of points from the pool of candidates is exactly matched to the pair of points

selected from the first dataset is $\frac{1}{637} \times \frac{1}{637} = \frac{1}{405769}$. However, as shown in the table, by utilizing prioritized Geo-PPM we could achieve an acceptable precision (i.e., over 80% for USGS+NGA data and over 90% for other cases) and recall (i.e., over 90%) by examining between 33 and 52 candidate pairs. This means that using the prioritized Geo-PPM, the possibility of selecting the

actual matching pair is between $\frac{3}{100}$ to $\frac{2}{100}$, which is up to 4 orders of magnitude better than that of Geo-PPM.

4. Related Work

There have been a number of efforts to automatically or semi-automatically detect matched features across different road vector datasets [3, 4, 5, 6, 7]. Given a feature point from one dataset, these approaches utilize different matching strategies to discover the corresponding point within a predetermined distance (i.e., a localized area). This implies that these existing algorithms only handle the matching of vector datasets in the same geometry systems (i.e., the same coordinate system). Hence, to the best of our knowledge, no general method exists to resolve the matching of two vector data in unknown geometry systems. In addition, various GIS systems (such as ESEA MapMerger⁹) have been implemented to achieve the matching of vector datasets with different accuracies. However, most of the existing systems require manual interventions to transform two road networks into same geocoordinates beforehand. Thus, they are not suitable for handling road networks in unknown geometry systems, while our approach can match two road networks in unknown geometry systems.

Finally, our approach discussed in this paper utilizes a specialized point pattern matching algorithm to find the corresponding point pairs on both datasets. The geometric

Datasets	USGS+ US Census	USGS+ NGA	USGS+ US Census
Number of Intersections	2367 + 2456	2367 + 133	920 + 1035
Average Number of Candidates	637	514	371
Point Pairs Examined	43	52	33
Processing Time	946 sec.	48 sec.	132 sec.
Precision	91%	82%	95.8%
Recall	92.5%	96.5%	95.8%

Table 1: Experimental Results (on a PC with 3.2GHz CPU)

⁹ <http://www.esea.com/products/>

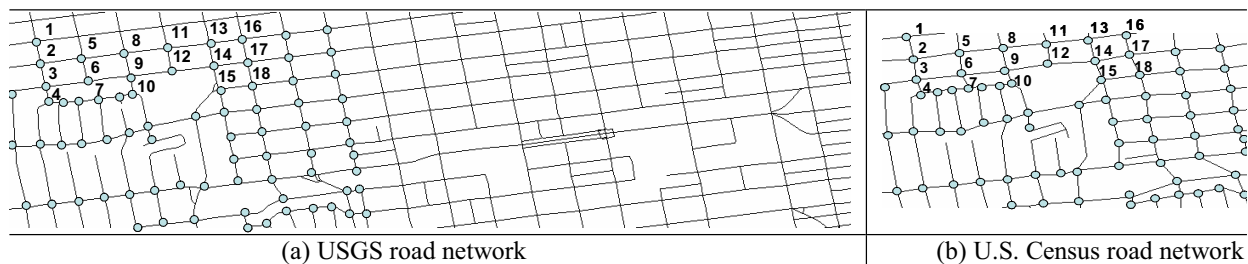


Figure 7: The partial result of matched points from two road networks (some matched points are labelled in order to show the corresponding points)

point set matching in two or higher dimensions is a well-studied family of problems with application to area such as computer vision, biology, and astronomy [12, 14].

5. Conclusion and Future Work

In this paper, we proposed an efficient and accurate technique, termed prioritized Geo-PPM, to locate the matched points between two road network datasets when the spatial attributes of the datasets are in unknown systems. In our solution, we first select pairs of points in the first dataset with the minimum number of candidates (i.e., point with similar connectivity and angles) in the second dataset, and then perform our PPM method on these pairs. Although our technique matches road networks at the point level (not at the road segment level), it takes the road connectivity, road directions and global distribution of road intersections into consideration. Our experiments show that this approach provides acceptable precision and recall values by only examining a very small number of pairs.

We plan to extend our approach in several ways. First, we plan to examine prioritized Geo-PPM for even larger road networks and for different patterns of road networks (e.g., rural roads and urban roads), and consider the orientations of the road networks as well. Second, we intend to investigate the appropriate order of utilizing the auxiliary information described in Section 2.4. Third, we would like to perform comprehensive comparisons between our approach and the related techniques described in Section 4. Finally, we also plan to use these matched points as control points to integrate different road network datasets.

6. Acknowledgements

This research is based upon work supported in part by the National Science Foundation under Award No. IIS-0324955. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

7. Reference

1. Barclay, T., Gray, J., and Stuz, D. *Microsoft TerraServer: A Spatial Data Warehouse*. in the 2000 ACM SIGMOD

International Conference on Management of Data. 2000. Dallas, TX: ACM Press.

2. Chiang, Y.-Y. and Knoblock, C.A. *Classification of Line and Character Pixels on Raster Maps Using Discrete Cosine Transformation Coefficients and Support Vector Machines*. in the 18th International Conference on Pattern Recognition. 2006.

3. Walter, V. and Fritsch, D., *Matching Spatial Data Sets: a Statistical Approach*. International Journal of Geographic Information Sciences, 1999. 13(5): p. 445-473.

4. Ware, J.M. and Jones, C.B. *Matching and Aligning Features in Overlaid Coverages*. in the 6th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'98). 1998. Washington, D.C: ACM Press.

5. Cobb, M., Chung, M.J., Miller, V., Foley, H.I., Petry, F.E., and Shaw, K.B., *A Rule-Based Approach for the Conflation of Attributed Vector Data*. Geoinformatica, 1998. 2(1): p. 7-35.

6. Saalfeld, A., *Conflation: Automated Map Compilation*. International Journal of Geographic Information Sciences, 1988. 2(3): p. 217-228.

7. Chen, C.-C., Shahabi, C., and Knoblock, C.A. *Utilizing Road Network Data for Automatic Identification of Road Intersections from High Resolution Color Orthoimagery*. in the Second Workshop on Spatio-Temporal Database Management, colocated with VLDB. 2004. Toronto, Canada.

8. Chen, C.-C., Thakkar, S., Knoblock, C.A., and Shahabi, C. *Automatically Annotating and Integrating Spatial Datasets*. in the 8th International Symposium on Spatial and Temporal Databases (SSTD'03). 2003. Santorini Island, Greece.

9. Habib, A., Uebbing, R., Asmamaw, A. *Automatic Extraction of Primitives for Conflation of Raster Maps*. 1999, The Center for Mapping, The Ohio State University.

10. Flavie, M., Fortier, A., Ziou, D., Armenakis, C., and Wang, S. *Automated Updating of Road Information from Aerial Images*. in the American Society Photogrammetry and Remote Sensing Conference. 2000. Amsterdam, Holland.

11. Guttman, A. *R-trees: a dynamic index structure for spatial searching*. in the SIGMOD Conference. 1984. Boston, MA.

12. Chew, L.P., Goodrich, M.T., Huttenlocher, D.P., Kedem, K., Kleinberg, J.M., and Kravets, D. *Geometric pattern matching under Euclidean motion*. in the Fifth Canadian Conference on Computational Geometry. 1993.

13. Cardoze, D.E. and Schulman, L.J. *Pattern Matching for Spatial Point Sets*. in the IEEE Symposium on Foundations of Computer Science. 1998.

14. Irani, S. and Raghavan, P., *Combinatorial and experimental results for randomized point matching algorithms*. Computational Geometry, 1999. 12(1-2): p. 17-31.

15. Chen, C.-C., Knoblock, C.A., Shahabi, C., Chiang, Y.-Y., and Thakkar, S. *Automatically and Accurately Conflating Orthoimagery and Street Maps*. in the 12th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'04). 2004. Washington, D.C: ACM Press.