Automated Georeferencing Based on Topological Point Pattern Matching

Yan Li
Geographic Information Sciences Program
School of Social Sciences
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
E-mail: yanli@utdallas.edu
Tel: 972-883-4943
Fax: 972-883-6297

Ronald Briggs
Geographic Information Sciences Program
School of Social Sciences
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
E-mail: rbriggs@utdallas.edu
Tel: 972-883-6877
Fax: 972-883-6297

Automated Georeferencing Based on Topological Point Pattern Matching

Yan Li and Ronald Briggs
GIS Program, The University of Texas at Dallas
E-mail: yanli@utdallas.edu, rbriggs@utdallas.edu

Abstract

In this paper, we develop a new approach for automated georeferencing of a raster image to a vector road network. Our approach improves existing solutions by: (1) eliminating the same scene constraint between an image and the vector road network; (2) requiring no pre-knowledge of the image's placement in the vector road network; (3) necessitating only a few points from the image; (4) tolerating point location distortion, missing points, and spurious points; (5) providing high performance and scalability. Our key contribution relies on the use of the topology of point patterns, which we call topological point pattern (TPP) analysis, along with a set of corresponding matching algorithms, to automatically link image and vector data sets. The biggest advantage of TPP is its flexibility to capture the spatial information of any portion of a point set. The TPP matching algorithms can identify control point pairs between an image and a vector road network by systematically searching the vector road network. The automated scheme is very efficient and highly scalable.

Keywords: Automated georeferencing, Image registration, Topological point pattern matching, Similarity transformation

1. Introduction

In order to exploit the full benefit of geospatial information from multiple data sources, it is essential to integrate a variety of datasets in a consistent and precise way. One important research area in the GIS data integration literature is georeferencing. Georeferencing involves correctly aligning a raster image with a map coordinate system. In this paper, we focus on image to vector automated georeferencing.

If a raster image does not contain correct positional information, or if the information is not available, we need to align it to a spatially referenced vector dataset such as a road network. Georeferencing now is a standard feature of GIS software. The typical georeferencing process includes the following steps: (1) identifying a set of control point pairs (CPPs) that link locations on a raster image with corresponding locations in a correctly positioned vector dataset; (2) calculating a transformation function from the raster image to the vector map based on CPPs; (3) transforming and resampling the image. Steps (2) and (3) normally are implemented by software. Step (1) is mainly a manual process. It requires the user to have substantial knowledge about the geographical location of the raster image in order to manually establish CPPs. This process is time consuming, error prone, and sometimes impossible because raster images usually are transformed, rotated and scaled, and the vector maps may be too big to navigate without a priori information about the images' location. For example, Fig. 1 is a satellite image that covers a small, unknown area in the City of Dallas, TX. Fig. 2 is the road network of the City of Dallas which has more than 58,000 road segments. It is difficult, if not impossible, to manually identify CPPs between them without a priori information about the image's approximate location.

There is a plethora of legacy raster images available and new raster images are generated at an accelerating rate. In order to overcome the limitations of the manual georeferencing method and meet the growing need, it is essential to develop automated georeferencing methods that do not rely on users to carry out key steps, such as finding and establishing control point pairs.

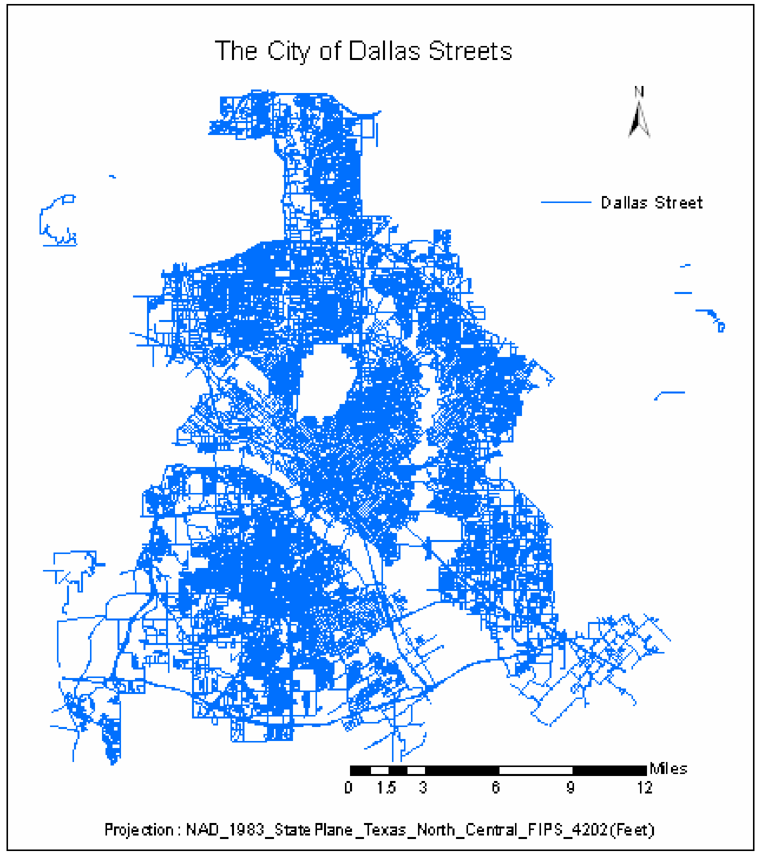

Fig. 1: A satellite image from Google map

Fig. 2: Road network of the City of Dallas, TX

There has been little progress in the area of automated georeferencing between a raster image and a vector road network. The existing studies mostly focus on aligning different datasets of the same scene (for example, Chen, 2005). Assuming the datasets are of the same geographical area effectively limits the size of the searching space and hence helps reduce the problem complexity. However, it significantly confines the scope of georeferencing and hinders the general applicability of the automated solution. To the best of our knowledge, no automated solution has been developed to address the georeferencing problem between a raster image and a very large vector road network, such as in Fig. 1 and Fig. 2.

In this paper, we develop a new approach for automated georeferencing from a raster image to a vector network under a similarity transformation. Our solution addresses the following issues:

(1) eliminates the same scene constraint between an image and the vector road network;
(2) requires no pre-knowledge of the image's placement in the vector road network;
(3) necessitates the use of only a few points from the image;
(4) tolerates point location distortion, missing points, and spurious points;
(5) provides high performance and scalability.

Our key contribution is a unique point pattern-based methodology, which we refer to as topological point pattern (TPP) analysis, and a corresponding set of matching algorithms. The biggest advantage of TPP is its flexibility to capture the unique spatial information of any portion of a point set. The TPP matching algorithms identify control point pairs between an image and a vector road network by systematically searching the vector road network for corresponding patterns of network intersections between the image and the network. The automated scheme is very efficient and highly scalable.

This paper is organized as follows. In Section 2, we will briefly review previous works in related areas. In Section 3, we discuss our general strategies to solve the automated georeferencing problem. The key algorithms and methods are introduced in Section 4, including TPP and its matching algorithms, as well as transformation verification. In Section 5, we present results of several experiments to demonstrate the feasibility, accuracy, efficiency and scalability of our approach. Section 6 concludes the paper.

2. Related works

Research related to image georeferencing can be divided into two main groups: image to vector georeferencing, and image to image registration.

2.1 Image to vector georeferencing

Depending on the input data, there are two types of image to vector georeferencing. The first type assumes there exists a transformation function between the two datasets. The automated solutions to this problem are limited to datasets of the same scene (Hild and Fritsch 1998). If both an image and a vector map can be properly represented by two sets of point patterns, then point pattern matching (PPM) algorithms can be used to find CPPs. Various PPM algorithms have been studied (Li 1992, Chang 1997, Cheng 1996). However, they typically assume point sets are about the same size and none of the existing matching algorithms can address all of the issues, including translation, rotation, scaling, and point data distortion, at the same time without significant limitations in either the complexity or size of the problem.

In the second type of image to vector map georeferencing, a practical transformation function does not exist between image and vector datasets. One such application is image and map conflation in which a slightly offset image or map is to be overlaid and aligned to a vector map. The automated conflation algorithms are studied in Chen (2005), Filin and Doytsher (2000), and Yuan and Tao (1999).

2.2 Image to image registration

Image to image registration is the process of overlaying multiple images of the same scene taken at different times, from different viewpoints, or by different sensors (Brown 1992). For an extensive review on image to image registration methods, readers are referred to Zitova's 2003 survey paper. There is a connection between image to image registration and image to vector georeferencing: if both images can be georeferenced to a common vector map, then the image to image registration problem can be reduced to the image to vector georeferencing problem.

3. System model

In this section, we discuss our general strategies to address the key issues raised in previous sections. We assume the use of road intersections for georeferencing since in both raster images and vector maps road intersection points are usually well defined, plentiful, and relatively easy to identify. The first step in georeferencing requires the extraction of the points from the image and vector data sets. The second step requires the identification of a set of CPPs linking raster points to corresponding vector points. The transformation function to convert the raster data set's positioning to that of the vector data set is then derived based upon the CPPs. In conventional georeferencing, the point extraction and the CPP identification are specified by users manually.

3.1 Automated point extraction

Points from an image can be extracted through image processing. For a road network, the extraction process involves two steps. In the first step, roads are extracted. In general, roads in a raster image have well-defined geometrical properties and are extracted as line segments. There are a wide variety of image processing algorithms available for road extraction depending on image resolution, image condition, and noise (Fortier et al 2000). In the second step, intersection points are calculated as intersections of multiple extracted roads.

Extracting all roads from a raster image can be difficult if the graphical features of interest are blocked, split, or obscured by other spatial objects such as trees, buildings, cars, etc., thus producing noisy information. Obviously, it is unrealistic to expect all intersection points to be found immaculately. Therefore any schemes assuming the same set of intersection points to be extracted from a raster image and a vector map are impractical.

It is much easier for extraction algorithms to focus only on the quality of the extracted intersection points. Most algorithms have thresholds available to control the quality of point extraction and filter out questionable points. Based on this fact, our proposed scheme is designed to use only "a few good intersection points" from a raster image. The number of points required from an image is not based on its size and the good points are used to manage point errors in other points. Consequently, our scheme can use any reasonably good intersection point extraction algorithm by setting their thresholds properly, for example, the algorithms discussed in Chiang et al (2005), Heipke (1997), and Sebok et al (1981).

3.2 Automated georeferencing

Automated georeferencing uses point pattern matching (PPM) algorithms to establish CPPs. Due to the complex nature of the matching problem, PPM algorithms typically assume the two point sets are about the same scene, i.e., *symmetric georeferencing*. The problem is, even with symmetric georeferencing, it is infeasible to assume point sets have the same size since extracting all points correctly from an image is not an easy task. Furthermore, most of the image to road network georeferencing is *asymmetric*, in which the raster image covers only a part of the area of the road network.

Any practically useful automated georeferencing scheme must address the above limitations effectively. Since developing a perfect point extraction algorithm is extremely difficult, if not impossible, we focus on how to remove the limitations on the point pattern matching algorithms so that point sets don't have to be the same size. If this is possible, we can greatly simplify the point extraction process and solve the asymmetric georeferencing problem simultaneously. This is also very important to the speed and scalability of the automated scheme since we no longer require the use of all points contained in the image. The solution we will discuss in Section 4 is a special case of point pattern matching, called topological point pattern (TPP) analysis, in which we rely on the correspondence between scaled distances and angles between pairs of points in the image and vector networks.

Due to the possibility of image distortion and misidentification by point extraction algorithms, points extracted from images may contain errors. It is very unlikely that point patterns will match exactly. Therefore, without point error tolerance, it is quite possible that good CPPs may be missed. However, with too much tolerance on point errors, it is possible to mistake bad CPPs as good CPPs. The solution is to evaluate the performance of the transformation to filter out bad CPPs. This is done through CPP and transformation verification, which is the second key component of our process.

3.3 Automated scheme

According to the above discussions, the proposed automated georeferencing scheme consists of the following key steps. The underlying algorithms are discussed in Section 4. Our contribution lies in the topological point pattern and matching algorithms.

1. extract intersection points from the vector road network and calculate TPPs (a one-time task)
2. select intersection points from a raster image and calculate TPPs
3. use matching algorithms to compare image TPPs with vector TPPs to find candidate CPPs
4. verify CPPs and transformations
5. transform and resample the raster image

4. Algorithms

4.1 Definitions

The goal of georeferencing is to find the right transformation from a raster image to a vector road network. In this paper, we assume the transformation function is a similarity transformation. A similarity transformation is defined by $ST = (A, B, C, D)$ or (s, θ, t_x, t_y) as

$$x_i' = A\,x_i + B\,y_i + C = s\,x_i \cos(\theta) + s\,y_i \sin(\theta) + t_x$$
$$y_i' = -B\,x_i + A\,y_i + D = -s\,x_i \sin(\theta) + s\,y_i \cos(\theta) + t_y$$

where s is the scale change (same in x and y directions), θ is the rotation angle measured counter-clockwise from the x-axis, t_x is the translation in x direction, and t_y is the translation in y direction. The similarity transformation of point p is denoted by $p' = (x_i', y_i') = ST(p)$. CPPs are used to solve for A, B, C, and D.

Let $V = \{\, v_i = (x_i, y_i) \mid i = 1, \ldots, n \,\}$ be a point set and $d_{ij} = dist(v_i, v_j)$ be the distance between points v_i and v_j, for point $v_i \in V$, we define $n(v_i)$ to be a point in V that is the nearest point to v_i and $d_i = dist(v_i, n(v_i))$.

4.2 Topological point pattern algorithm

All the conventional automated georeferencing methods define one point pattern for a raster image and one for a vector map. Such point patterns are only suitable for matching an image with a vector map in whole, but are not effective for asymmetric georeferencing. Another big disadvantage is the scalability problem: as the sizes of images and vector maps grow, so does the size of the point patterns. To overcome these limitations and, more importantly, to support automated asymmetric georeferencing, we propose a new point pattern that is flexible, efficient, and scalable, i.e., the *topological point pattern* (TPP).

A TPP is defined based on a special point, called an *anchor point*. As a matter of fact, a TPP represents a unique view of the point set from its anchor point. Every point in a point set V can be an anchor point and hence have its corresponding TPP. Therefore, the TPP for V is not just one point pattern, but instead, a set of n TPPs, i.e.,

$$tpp(V) = \{\ tpp(v_i) \mid for\ v_i \in V\ \}$$

where $tpp(V)$ is the TPP for point set V and $tpp(v_i)$ is the TPP for anchor point v_i.

Let $v_i \in V$ be the anchor point. For the simplicity of discussion, assume $n(v_i) = v_{i'}$ is unique. For each point $v_j \in V$, $j \neq i$, define $r_{ij} = \frac{d_{ij}}{d_i}$ and let $0 \leq \theta_{ij} < 2\pi$ be the angle from vector $\overrightarrow{v_i v_{i'}}$ to vector $\overrightarrow{v_i v_j}$ in counter-clockwise direction. Set $r_{ii} = 0$ and $\theta_{ii} = 0$, then $tpp(v_i)$ is defined by

$$tpp(v_i) = \left\{\ v'_j = \left(x'_j, y'_j\right)\ \middle|\ v_j \in V\ \right\} \quad \text{where} \quad x'_j = r_{ij}\cos(\theta_{ij}) \ \text{and}\ y'_j = r_{ij}\sin(\theta_{ij})$$

In the other words, $tpp(v_i)$ is the point set V in a new coordinate system whose origin point is v_i, x-axis coincides with vector $\overrightarrow{v_i v_{i'}}$, and unit distance is d_i. Fig. 3b shows the TPP for v_1 in V.

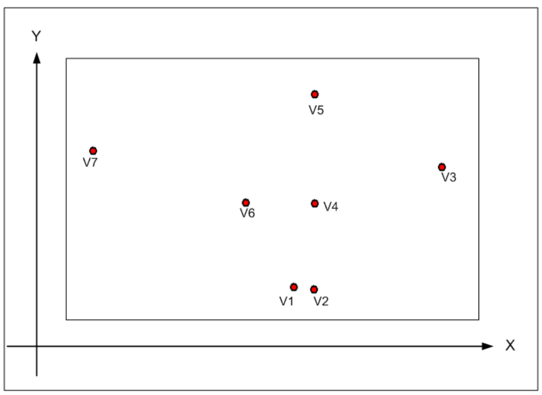

Fig. 3a Extracted point set V

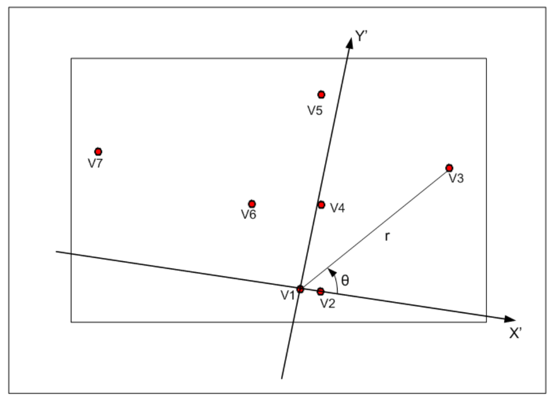

Fig. 3b TPP example: $tpp(v_1)$

Since each $tpp(v_i)$ of $tpp(V)$ includes all points of V, a matching for $tpp(v_i)$ is also a matching for V. It is also very easy to derive the point pattern for points in any surrounding area of point v_i using $tpp(v_i)$. For example, the following subset of $tpp(v_i)$,

$$tpp(v_i, d) = \left\{(r_{ij}\cos(\theta_{rj}), r_{ij}\sin(\theta_{ij}))\ \middle|\ r_{ij} \leq d\right\} \subseteq tpp(v_i)$$

is a point pattern for only those points that are within the distance of d from v_i. The local point pattern information captured by the definition of $tpp(v_i)$ provides the foundation for automating the asymmetric georeferencing problem. To georeference an image to a vector map V, we only need to find the right v_i and d and use the points specified by $tpp(v_i, d)$. In fact, we can view each point set as being covered by many small $tpp(v, d)$. This is very difficult to achieve in conventional point patterns.

Another feature of TPP is that we can sort all points in $tpp(v_i)$ based on their polar coordinates (r_i, θ) according to $(r_i, \theta_i) \prec (r_j, \theta_j)\ if\ (r_i < r_j)\ or\ (r_i = r_j\ \&\ \theta_i < \theta_j)$. Then, any two sorted TPPs and their subsets can be matched quickly. More importantly, this order is very efficient for confining point matching within a region and avoiding unnecessary point comparisons. For example, we can effectively limit the georeferencing region to a specific area around a point *v* by only using the first *m* points in $tpp(v)$. Since solving a similarity transformation requires only a few CPPs and the sorted TPPs allow points to be matched in a small region, there is no need for our scheme to use all the points in a raster image for georeferencing. In fact, even when image size increases, the number of points needed from the image can remain the same. This explains why our proposed scheme is highly scalable.

4.3 Matching algorithm

Let R and V be the set of extracted points from a raster image and a vector road network respectively. The matching algorithm searches for candidates of CPPs between R and V. Each candidate set of CPPs is called a candidate matching (CM). We will first describe the basic matching algorithm and then discuss its improvements.

```
Algorithm: basic algorithm for finding candidate CPPs
Input: R for a raster image and V for a vector road network
Output: CCM: collection of CMs
SET CCM to be an empty set;
select an anchor point r ∈ R such that n(r) in R also is r's nearest neighbor in the image
compute tpp(r) in R and sort it based on the polar coordinates
FOR each point v in V DO
   compute tpp(v) in V and sort it based on the polar coordinates
ENDFOR
FOR each v in V DO
   IF tpp(r) ⊆ tpp(v) THEN
      SET acm to be the set of matching point pairs between tpp(r) and tpp(v)
      IF (there are enough point pairs in acm) THEN
         add acm to CCM
      ENDIF
   ENDIF
ENDFOR
return CCM
```

In the above algorithm, we assume there are two intersection points in R that are both nearest in R and in the original image. This is because if r in R and v in V form a correct CPP then for $tpp(r)$ to match a subset of $tpp(v)$ they must use the same unit distance. Due to space limitations for this paper, we will not discuss how to modify the algorithm to drop this assumption.

The matching between $tpp(r)$ and $tpp(v)$ is based on their sorted lists. When comparing $p = (r_i, \theta_i) \in tpp(r)$ with $q = (r_j, \theta_j) \in tpp(v)$, due to point distortion in image and extraction error, it is unlikely that p and q will match exactly even though (p, q) is a correct CPP. Therefore, we define that p and q are *matched* and they form a CPP if $|r_i - r_j| \le \Delta r$ and $|\theta_i - \theta_j| \le \Delta\theta$. Δr and $\Delta\theta$ are predefined thresholds for error tolerance. If there are multiple points in V that match p, the algorithm selects the closet q to form a CPP with p.

The performance of the basic matching algorithm can be improved in several ways. First, we could use all the extracted points in the basic matching, but this is inefficient and unnecessary. To avoid coincidental mismatching, we need sufficient points in R; on the other hand, we need no more than what is necessary to verify CPPs. This means that, even if the size of the image gets bigger, the number of points in R does not have to grow proportionately. For the same reason we do not have to use all the points in $tpp(v)$; using $tpp(v,d)$ with proper d will be sufficient. By keeping the size of $tpp(r)$ and $tpp(v)$ stable, the total computation time for the matching algorithm is stabilized and, therefore, the algorithm is highly scalable. Another factor affecting the number of points required for matching is the tolerance on point errors. We need to control the percentage of unmatched points due to errors under a threshold. In our experiments, using 7 to 14 points from an image is quite sufficient.

For asymmetric georeferencing, the complexity of the matching algorithm is dominated by the size of *V*. For the basic algorithm, the complexity for computing all $tpp(v)$ in *V* is $O(N^2 \log N)$ and the complexity of matching $tpp(r)$ with all $tpp(v)$ is $O(N \log N)$, where $N = |V|$. If the vector road network is fixed, $tpp(v)$ only needs to be pre-computed once. Therefore, Lines 4 through 6 in the algorithm can be skipped. This is typical when different images are georeferenced against the same vector map repetitively. If we limit $tpp(v,d)$ to only include the nearest *m* points of *v*, where *m* is constant and big enough for $tpp(v,d)$ to cover all points in *R*, then the complexity of matching is reduced to *O(N)*.

4.5 Verification algorithm

The output of the basic matching algorithm contains all candidate sets for CPPs. The question is which set is correct. The verification algorithm verifies the quality of each set of CPPs by measuring the performance of the ST derived from it. For a correct set of CPPs, its *ST* function should map each intersection point in *R* to an intersection point in *V*. Although this is only a necessary condition, if *R* consists of sufficient points, the chance of each point being coincidentally mapped to a mismatched point is negligible. Therefore, if a set of CPPs produces a close matching between intersection points, then it is considered to be correct.

Let $CM = \{(r_i, v_i) | r_i \in R, v_i \in V\}$ be a candidate matching found by the above matching algorithm and denote $v_i = CM(r_i)$. For a similarity transformation *ST*, $ST(r_i)$ is the transformed point. The pair-wise residual error for point r_i under a CM and a *ST* is defined by the distance between v_i and $ST(r_i)$, i.e., $\varepsilon(CM, ST, r_i) = dist(CM(r_i), ST(r_i))$. The total error under *CM* and *ST* is defined to be the root mean square (RMS) of all pair-wise residual errors, i.e.,

$$\varepsilon(CM, ST) = \sqrt{\frac{1}{|CM|} \sum_{(r_i, v_i) \in CM} \varepsilon^2(CM, ST, r_i)}$$

We define the matching error of *CM* to be the minimum RMS under the optimal *ST*, i.e.,

$$\varepsilon(CM) = \min_{ST} \varepsilon(CM, ST)$$

We denote the optimal *ST* by $ST_{opt}(CM)$. For each CM, both $\varepsilon(CM)$ and $ST_{opt}(CM)$ can be solved using the least squares method (Press 1992). The verification algorithm is as follows.

Algorithm: verification algorithm
Input: *CCM*: a collection of *CM*s
Output: the best *CM* and its optimal *ST*
1. FOR each *CM* in *CCM* DO
2. calculate $\varepsilon(CM)$ and $ST_{opt}(CM)$
3. ENDFOR
4. Find CM' such that $\varepsilon(CM') = \min_{CM \in CCM} \varepsilon(CM)$
5. return CM' and $ST_{opt}(CM')$

Finally if $ST_{opt}(CM')$ is good, $\varepsilon(CM')$ should be very small. By combining the matching algorithm with verification algorithm, our proposed automated scheme can find the set of correct control point pairs as well as the similarity transformation to align the raster image to the vector road networks. This completes the process of automated georeferencing.

5. Experiments and discussion

In this section, we describe the experiments conducted upon the satellite image of Fig. 1 and the vector road network of Fig. 2 and present example results from the automated georeferencing algorithms. We use the City of Dallas, TX as our study area because it represents a very large urban area for testing the feasibility and performance of our approach. The satellite image is saved from a screen snapshot of Google maps in JPG format. It is intentionally rotated, translated, and scaled. And we do not have its actual location in the city. The vector dataset used in the experiment is a road network file for the City of Dallas in ESRI shapefile format.

In our experimental system, the algorithms are implemented in VBA for ArcObjects and C++, running in ESRI ArcGIS Desktop 8.3 or above. The hardware used for all the experiments is a DELL PC with Pentium IV 3.0GHz processor, 1G memory, and windows XP professional.

The TPPs for the vector road network need only be computed once and are then used repeatedly for georeferencing any image in the region covered by the network. To obtain these vector TPPs, we wrote a VBA application to automatically extract a total of 24,522 intersection points from the road network of the City of Dallas (Fig. 5). We then pre-computed TPPs for nine point subsets of various sizes; it took 247 seconds and 767 seconds to pre-compute the TPPs for the point sets of 5,029 and 10,008 points respectively, for example.

For the satellite image we used two sets of arbitrarily chosen points: one with six intersection points plus one fake point, and the other with twelve intersection points plus two fake points, as shown in Fig. 4 and Fig. 6. We then used the pre-computed TPPs for the eighteen experiments. The matching and verification time for each experiment is listed in Table 1. Most of the experiments only take a few seconds to complete matching and verification. The total RMS error is 8.466 for the cases of 14 image points, and 7.064 for the cases of 7 image points. The georeferencing results for 7 image points are shown in Fig. 7.

To summarize, the results of our experiments have shown that the proposed automated georeferencing scheme is very efficient and accurate in identifying the correct CPPs and producing good georeferencing results. The scheme has achieved all the goals stated in Section 1.

6. Conclusions and future work

In this paper, we proposed a new scheme to solve the automated georeferencing problem under a similarity transformation. The proposed scheme dramatically improves the speed and reliability of georeferencing and can align a raster image with a vector road network without any priori knowledge of the image's location. Our approach relies upon topological point pattern (TPP) matching. It effectively solves the automated asymmetric georeferencing problem and reduces the number of points required to be extracted from the raster image. It is highly scalable, and robust in handling the errors in extracted points through the combination of a matching and a verification algorithm. The experimental results confirm the efficiency and scalability of the new scheme with large datasets.

In the future, we plan to continue to improve and extend the proposed solution by addressing the automated georeferencing problem under affine transformation, by incorporating automated point selection from the image, and by exploring the general robustness of the system relative to such issues as image resolution.

Fig. 4: 7 selected image points. 1 circled in yellow is false.

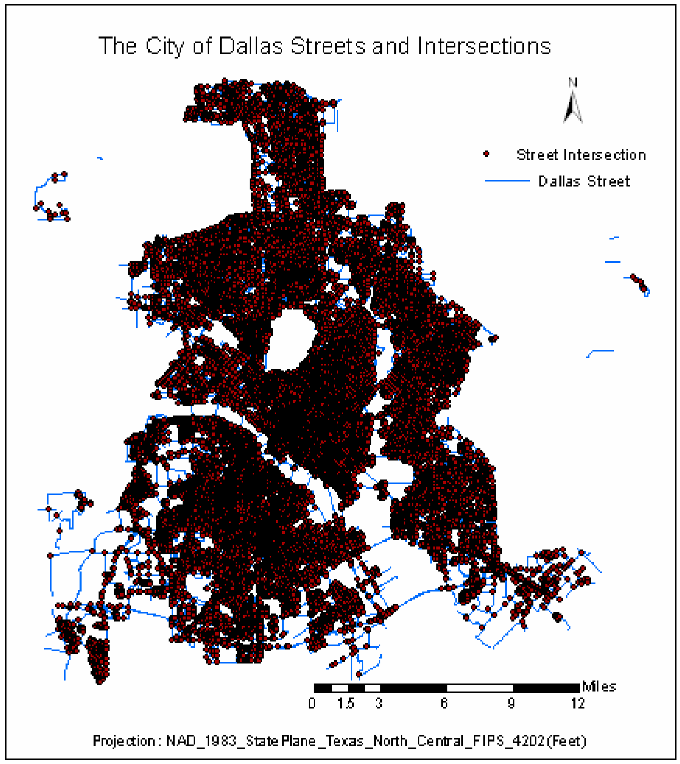

Fig. 5: Dallas street intersections. (24522 points)

Fig. 6: 14 selected image points. 2 circled in yellow are false.

Matching and verification time (seconds)		Number of intersection points in the road network								
		518 points	1506 points	2508 points	5029 points	7511 points	10008 points	15150 points	20004 points	24522 points
Points number in the image	7 Points	0.188	0.609	1.109	2.328	3.531	4.674	7.391	11.125	11.982
	14 points	0.234	0.75	1.344	2.797	4.219	6.234	9.031	11.609	13.078

Table 1: Execution time for matching and verification with respect to different point sizes

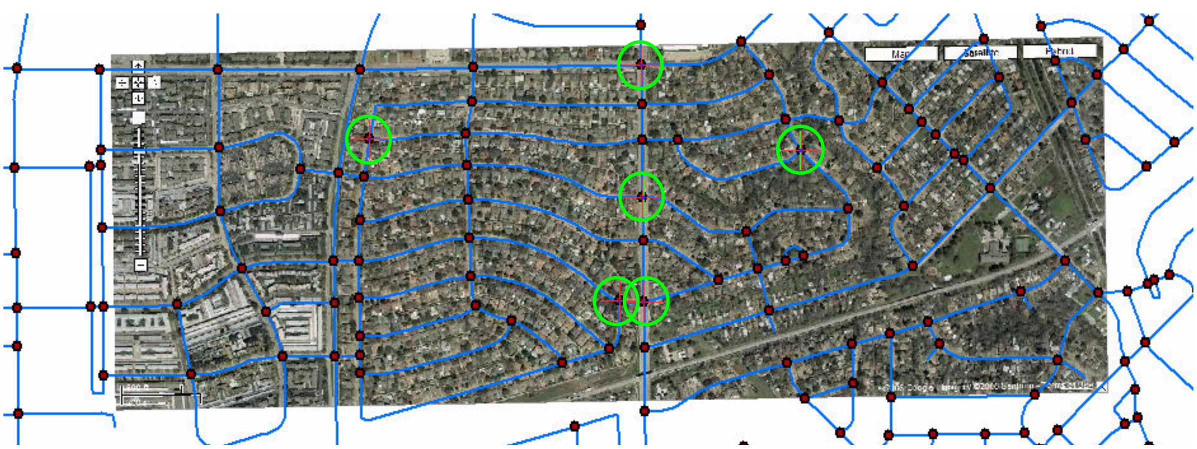

Fig. 7: Georeferencing result for 7 image point tests. Total RMS error is 7.064. 1 false point is filtered out.

References:

Brown, Lisa Gottesfeld. 1992. A survey of image registration techniques. *ACM Computing Surveys* 24 (4): 325-376.

Chen, Ching-Chien. 2005. *Automatically and accurately conflating road vector data, street maps and orthoimagery*. Dissertation. University of Southern California.

Chang, Shih-Hsu, Fang-Hsuan Cheng, Wen-Hsing Hsu, and Guo-Zua Wu. 1997/2. Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes. *Pattern Recognition* 30 (2): 311-320.

Cheng, Fang-Hsuan. 1996. Point pattern matching algorithm invariant to geometrical transformation and distortion. *Pattern Recognition Letters* 17 (14): 1429-1435.

Chiang, Yao-Yi, Craig A. Knoblock , Ching-Chien Chen. 2005. Automatic extraction of road intersections from raster maps. *Proceedings of the 13th annual ACM International Workshop on Geographic Information Systems,* Bremen, Germany.

Filin, S. and Y. Doytsher. 2000. "A Linear Conflation Approach for the Integration of Photogrammetric Information and GIS Data". *International Archives of Photogrammetry and Remote Sensing*, 33: 282-288.

Fortier, A., D. Ziou, C. Armenakis, and S. Wang. 2000. *Survey of Work on Road Extraction in Aerial and Satellite Images.* Technical Report, Universite de Sherbrooke.

Heipke, C., H. Mayer, and C. Wiedemann. 1997. Evaluation of Automatic Road Extraction. *International Archives of Photogrammetry and Remote Sensing* 32: 47-56.

Hild, Heiner and Dieter Fritsch. 1998. Integration of vector data and satellite imagery for geocoding. *International Archives of Photogrammetry and Remote Sensing* 32: 246--251.

Li, S. Z. 1992. Matching: Invariant to translations, rotations and scale changes. *Pattern Recognition* 25 (6): 583-594.

Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. 1992. *Numerical Recipes in C : The Art of Scientific Computing.* New York: Cambridge University Press

Sebok, T.J., L.E. Roemer, and J. Malindzak. 1981. An Algorithm for Line Intersection Identification. *Pattern Recognition* 13 (2): 159-166.

Sprinzak, Josef, and Michael Werman. 1994. Affine point matching. *Pattern Recognition Letters* 15 (4): 337-339.

Umeyama, Shinji. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (4): 376–380.

Zhang, Lihua, Wenli Xu, and Cheng Chang. 2003. Genetic algorithm for affine point pattern matching. *Pattern Recognition Letters* 24 (1-3): 9-19.

Zitova, Barbara, and Jan Flusser. 2003. Image registration methods: a survey. *Image and Vision Computing* 21 (11): 977-1000.

Yuan, Shuxin and Chuang Tao. 1999. Development of Conflation Components. *Proceedings of Geoinformatics and Socioinformatics*, Ann Arbor, Michigan, USA.