Noname manuscript No.
(will be inserted by the editor)

Image Feature Based GPS Trace Filtering for Road Network Generation and Road Segmentation

Jiangye Yuan · Anil M. Cheriyadat

Received: date / Accepted: date

Abstract We propose a new method to infer road networks from GPS trace data and accurately segment road regions in high resolution aerial images. Unlike previous efforts that rely on GPS traces alone, we exploit image features to infer road networks from noisy trace data. The inferred road network is used to guide road segmentation. We show that the number of image segments spanned by the traces and the trace orientation validated with image features are important attributes for identifying GPS traces on road regions. Based on filtered traces, we construct road networks and integrate them with image features to segment road regions. Our experiments show that the proposed method produces more accurate road networks than the leading method that uses GPS traces alone, and also achieves high accuracy in segmenting road regions even with very noisy GPS data.

Keywords GPS · Aerial image · Road map · Segmentation

1 Introduction

Inferring road networks and segmenting road regions in high resolution aerial images are important tasks that can benefit diverse applications. Roads are the vital data layer in geospatial databases. Updated and accurate road networks and road regions are highly desired for route planning and vehicle navigation. In addition, recent research shows that knowing road regions provides contextual information that is valuable in image analysis tasks (e.g., vehicle detection) [19,12,16]. However, due to the large variations of road appearances, identifying road regions is rather challenging [18]. A large number of methods for extracting road regions have been proposed [2,7,14,17,27]. Most of those methods assume that road appearances can be modeled in terms of certain spectral, spatial, and geometric properties that differentiate road regions from other regions in images. The road model is predefined or learned from labeled data. However, this assumption can be substantially violated for high resolution images containing complex scenes, where various pavement markings, vehicles, vegetations, and shadows are visible on roads. Therefore, those methods have difficulty achieving a reliable performance.

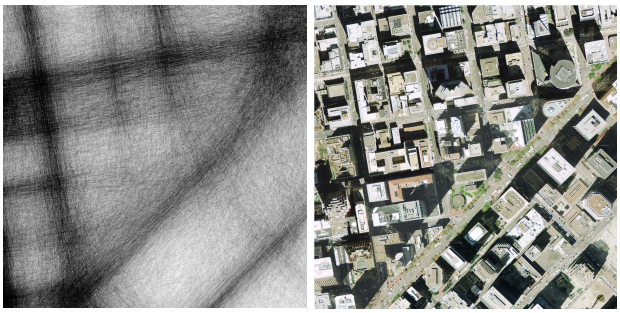


Fig. 1 Illustration of raw GPS traces with significant noise. Left: the traces are generated by taxi cabs in the downtown area of San Francisco. Darker pixels represent more traces passing. Right: the corresponding aerial image.

In this paper, we utilize GPS traces of vehicles to (i) infer road networks, and (ii) guide road segmentation using recovered road network. GPS receivers are widely deployed in various kinds of vehicles, which generate large volumes of GPS trace data. The data consists of sequences of location and time information. It

J. Yuan · A. M. Cheriyadat
Computational Sciences & Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
E-mail: yuanj@ornl.gov

has been used for creating or updating road networks [6,21]. Despite prior work, how to deal with noise in the GPS data remains a major issue. The data is often generated by low-cost devices, which gives position records of limited accuracy. The positioning errors can reach over 100 meters in areas with severe signal interference. Also, due to energy consumption and storage capacity, the data tends to have a low sampling rate (e.g., once per minute). Previous methods of using GPS traces to infer road networks mainly adopt three strategies including k-means clustering, trace merging, and kernel density estimation [3]. They perform well in the case when the traces aggregate more densely on actual roads than in other areas. Unfortunately, such a case is not guaranteed in real world data. Fig. 1 shows a dataset of taxi traces in the downtown area of San Francisco, CA , where the traces spread over the entire area because of measurement errors and low sampling frequency. Such trace data is very difficult, if not impossible, to deal with for the methods solely relying on GPS data.

We present a new method that integrates GPS data with image features. Computer vision techniques are applied to images, which identify the traces that lie in potential road regions. After the traces are filtered based on image information, we design a simple approach to obtain high quality road networks. Road networks are used along with images to produce road regions. The proposed method is built on the algorithms introduced in [23,24]. This paper provides comprehensive versions of those algorithms, a complete method to perform road segmentation in aerial images, and experiments on large real world datasets.

The rest of the paper is organized as follows. In Section 2, we give an overview of the proposed method. The components of the method are discussed in detail from Section 3 to Section 5. In Section 6, we conduct experiments on large-scale data and quantitatively evaluate the results. We conclude in Section 7.

2 Method Overview

Fig. 2 gives an overview of the proposed method for road segmentation. The aerial image and the corresponding GPS trace data are taken as input. GPS traces are discretized into the same image grid. The image is segmented using a factorization-based algorithm to produce a mid-level representation of the image, which plays an important role in the entire process. The GPS traces are filtered based on their alignments with respect to image segments and local orientations estimated from structure tensors. The filtered traces are mostly aligned with roads in the image and can be processed to generate a road network. Based on the segmented image and the road network, we examine the spatial distribution of region boundaries related to lines in the network and identify road edges that define road regions. We now give a detailed description for each step in the following sections.

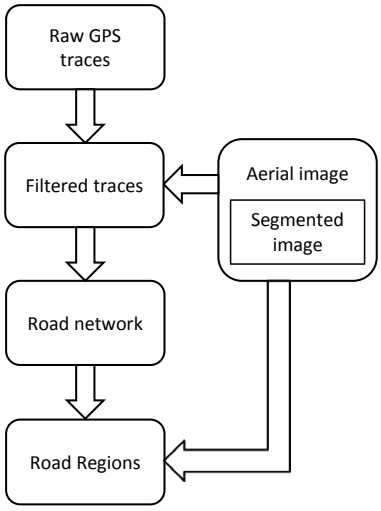


Fig. 2 Overview of the proposed method.

3 Trace Filtering

3.1 GPS Trace Data

GPS trace data consists of location records. Let $r :< x, y, t >$ denote a record, where x and y are position coordinates and t is the time. The entire trace data is organized into trips where each trip is a time sequence of location records. Let r_j^i denote the j^{th} record for trip i. Line connecting position coordinates of r_j^i and r_{j+1}^i represents a trace segment. We drop time t from location record as it is not used in our analysis and refer to r_j^i as a trace point.

GPS trace data is prone to positional errors from signal interference and device inaccuracies. Additionally, for data collected at a low frequency, the consecutive trace points can be far apart that trace segment connecting the points might span non-road regions making direct estimation of road networks from trace data challenging. Fig. 3 shows an example, where the middle trace segment does not correspond to the actual road. Although those trace points may not be noise, such trace segments can make detecting actual road networks difficult. We seek to use image features to filter out trace segments that span non-road regions

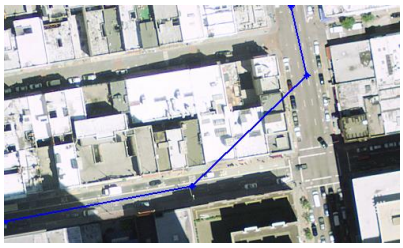


Fig. 3 An example of noisy traces caused by low sampling rate.

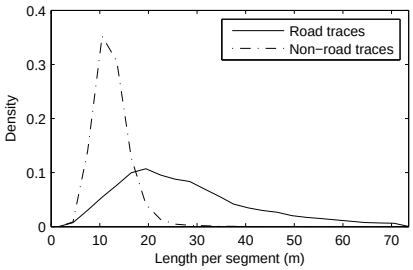


Fig. 4 Distribution of road and non-road traces with respect to length-per-segment values.

with the objective that the road network can be more accurately recovered from traces.

3.2 Initial Filtering

We begin by eliminating non-road trace segments based on the spatial density of trace points. We estimate the spatial density of trace points by applying a Gaussian kernel with σ set to 3 meters, which is determined based on expected GPS errors and road width. Pixels with density values outside one σ range are labeled as non-road regions under the assumption that spatial density of trace points on actual road regions are sufficiently high. For a trace segment, a measure T_g is defined as the length of trace segment passing through non-road regions over the total length. We remove traces with T_g values larger than 0.3. It should be noted that this filtering step use GPS data alone to remove obvious outliers but cannot deal with regions subjected to significant noise. For instance, this step removes few traces in the dataset shown in Fig. 1(a) because the point density is high over the entire area. Hence, we seek to exploit image features to further remove noisy traces.

3.3 Image based Filtering

3.3.1 Image Segmentation

The first technique is image segmentation, which partitions an image into homogeneous regions. Since road pixels tend to be grouped together to form large segments in a reasonable segmentation, trace segments on roads should span fewer segments than non-road trace segments. We define measure T_s as the ratio of trace length to total number of image segments spanned by the trace. A small T_s suggests that the trace traverses a large number of segments, which is unlikely to be a road trace segment.

Despite the large number of existing algorithms, segmenting aerial images remains a challenging task, especially for images containing various ground objects [9, 20]. We employ a factorization-based segmentation algorithm [26]. It has shown be effective to segment aerial images with great efficiency [25]. For completeness sake, we give a brief description of the algorithm. An image is convolved with a bank of filters, and a local spectral histogram [13] is computed at each pixel location, which consists of histograms of different filter responses within a square window centered at the pixel. The size of the window is called integration scale, which is a tunable parameter. Such a feature can capture the appearance of local window, and a homogeneous region has a representative feature that is similar to other features in the region. Each feature in an image can be approximated by a linear combination of representative features, and combination weights indicate the region ownership of the corresponding pixel. A factorization-based image model can be expressed in the following equation,

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{1}$$

$\mathbf{Y}$ is a feature matrix consisting of columns representing features at all pixel locations. $\mathbf{Z}$ contains columns corresponding to representative features. Each column of $\boldsymbol{\beta}$ is the combination weights at each pixel location. The largest weight in each column indicates the segment the corresponding pixel belongs to. $\boldsymbol{\varepsilon}$ represents the noise.

Based on this image model, the segmentation algorithm aims at factoring $\mathbf{Y}$ into two matrices. By applying singular value decomposition to $\mathbf{Y}$, the number of segments can be estimated, and a subspace can be revealed where all features reside. Initial representative features are estimated by analyzing the feature distribution in the subspace. A nonnegative matrix factorization algorithm [1] is then applied to obtain the factored matrices that give segmentation.

In order to show whether our segmentation result is useful for identifying traces on roads, we classify the trace segments in Fig. 1 into road and non-road based on their maximum distances to center road lines that are manually drawn, and plot the distribution of our computed measure T_s. (see Fig. 4(a)). We can see that

the majority of non-road traces have a small value for T_s. Here, we remove the non-road traces by simple thresholding. The threshold is set to 20 meters per segment for the experiments in this paper. Although some road traces are discarded in this step, we find they are mostly short traces that happen to cross segment boundaries and discarding them does not significantly affect results.

Fig. 5(b) displays the result of applying this filtering step to the trace data in Fig. 5(a). The number of noisy trace segments is clearly decreased.

3.3.2 Orientation estimation

As can be seen from Fig. 5(b), there still exist a noticeable amount of noisy traces, which happen to lie in large image segments resulting from either large regions or under-segmentation. Different types of information is required to remove those traces.

Inspired by Harris corner detector [11], we propose a structure tensor approach to examine whether the alignment of a trace segment is consistent with the image content in its surroundings. From an aerial view, one can observe that most objects on and near a road, including vegetation, pavement markings, vehicles, and buildings, spread along the road. Consequently, if we shift the image patch containing a road and compute the pixel difference, the largest difference occurs when the shift is perpendicular to the road orientation, and the smallest difference occurs when it is parallel to the road orientation. The structure tensor is used to find the orientation.

Given an image I and an image window W, we slightly shift the window by $(\Delta x, \Delta y)$, and the sum of square differences between W and the shifted window is written as

$$S = \sum_W \left[I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)\right]^2, \tag{2}$$

where (x_i, y_i) is the pixel location in the window. The second term in the equation, denoting the shifted window, can be approximated by the first order term of Taylor expansion. Then, Equation 2 can be rewritten as

$$S = [\Delta x, \Delta y]\, A \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \tag{3}$$

where A is a matrix called structure tensor and takes the following form

$$A = \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix}. \tag{4}$$

Here I_x and I_y denote the partial derivatives in x and y, which are gradients in the image sense. For corner detection, the relationship between two eigenvalues of A indicate whether the image window contains a corner.

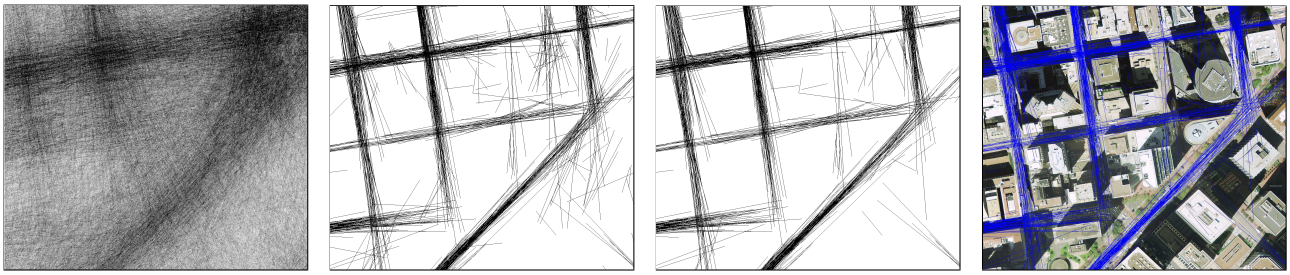
Let U denote the shift vector $[\Delta x, \Delta y]^T$. According to the Rayleigh quotient [10], $\frac{U^T A U}{U^T U}$ reaches the minimum value, which equals to the smallest eigenvalue of A, when U is the corresponding eigenvector. Likewise, the maximum value equals to the largest eigenvalue. It is reasonable to assume a fixed norm for the shift vector. Thus, the minimum value of S is given by the smaller one of two eigenvalues of A, and it is achieved when the shift vector is the corresponding eigenvector. Based on the aforementioned characteristics of road appearances, this direction should agree with the road orientation. The two eigenvectors essentially indicate the dominant direction of gradients in the image window. Note that the average of gradients does not well represent the dominant direction [5].

Now we can easily examine whether the orientation of a trace segment is consistent with the orientation estimated from the image. For each trace segment, the image patch is defined as a rectangular area around the trace that has the same length as the trace segment and a width of 30 meters. The gradients within the patch are used to construct the structure tensor, and the eigenvector corresponding to the smaller eigenvalue gives the orientation of the patch. We define measure T_o as the difference in angle between a trace segment and the eigenvector, and only retain the traces with a T_o value less than 15 degree. In the case when the image patch contains no road, the eigenvector does not reflect a road direction and is unlikely to coincide with a noisy trace. Fig. 5(c) shows the filtering result after this step. Compared with the original data in Fig. 1, noisy traces are significantly reduced, and road patterns are now clearly visible. To provide a better illustration, we overlay the traces on the aerial images, shown in Fig. 5(d). As we can see, the filtered traces are mostly located on roads.

Given input GPS trace dataset containing N trips, represented as $\{r^1, r^2, \ldots, r^N\}$, the procedure of filtering traces is described in Algorithm 1.

Algorithm 1 Trace filtering

```
for i:=1 to N do
  for each trace segment {r_j^i, r_{j+1}^i} in trip r^i do
    Compute T_g, T_s, T_o
    if (segment meets thresholding conditions) then
      Retain trace segment
    end if
  end for
end for
```



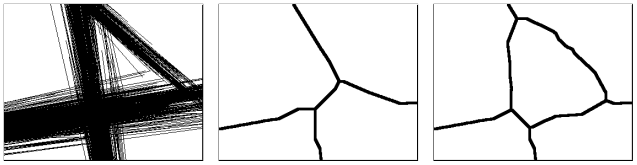
(a) (b) (c) (d)

Fig. 5 Trace filtering. (a) Traces after initial filtering. (b) Traces filtered based on the segmentation result. (c) Traces filtered by examining the alignment with respect to the image content. (d) Filtered traces (blue lines) overlaid on the aerial image.

4 Road Network Generation

We begin by mapping filtered traces onto image space. By treating the pixels covered by filtered trace segments as foreground, we obtain a binary image. Now, with the filtered traces well describing roads, we apply morphological operations to the binary image to generate a road network. A closing operation is first performed to fill small gaps among traces. Closing includes a dilation operation where each background pixel next to an object pixel is turned into an object pixel, and an erosion operation where each object pixel next to a background pixel is turned into a background pixel. Then, a thinning operation is used to extract the medial axes, or skeletons, which continuously removes boundaries pixels but preserves the extent and connectivity of foreground objects. A medial axis point is the center of a circle that touches object boundaries at two or more points. The circle radius reflects local thickness, which in our case can be interpreted as road width. The medial axes extracted from the binary image give a road network.

For roads that are close to each other, the corresponding traces can incorrectly merge after the closing operation. As a result, the medial axes do not represent the actual roads. Fig. 6 shows an example of this issue. Fig. 6(b) displays the medial axes extracted from the traces in Fig. 6(a), which fail to capture the structure of actual roads. Here we design a simple scheme to cope with this issue. Since incorrectly merged traces are from multiple roads, they tend to form thick patterns in the binary image. Thus, we locate those traces by selecting the medial axis points with radii larger than a predefined maximum road width. The foreground pixels associated with those medial axis points form a number of connected regions. The incorrect merging is caused by the traces in the areas between roads. Those traces are distributed much less densely than the traces on the roads. Therefore, we reduce the effect of those traces based on their density. For each connected region, we estimate the trace density using a Gaussian kernel. A set



(a) (b) (c)

Fig. 6 Extracting medial axis from filtered traces. (a) Filtered traces. (b) The extracted medial axes using morphological operations. (c) The extracted medial axes after corrections.

of thresholds are applied to the density map to yield different foreground regions, as well as the corresponding medial axes. The medial axes with the largest density value are selected to replace the original medial axes. The new medial axes better represent the road structure, hence a more accurate road network. Fig. 6(c) presents the result after this step, which shows a clear improvement.

Based on the extracted medial axes, we find all the intersection points and end points. We prune those points by merging intersection points close to one another and removing small branches. The Douglas-Peucker algorithm [8] is then used to reduce the path between points into line segments. A graph can be constructed, which provides a typical representation of road network.

5 Road Region Segmentation

Next, we show that the road network generated from the GPS traces can be further used to produce accurate road segmentation. A simple yet effective method is proposed in [24], which utilizes publicly available road vector data to segment road regions from aerial scenes. Given the road network obtained above, this method can be readily used to obtain road regions.

Given the road network generated above, segmenting road regions can be formulated as identifying two road edges parallel to each line in the network. We overlay the road network with a binary map representing re-

gion boundaries, which is available from factorization-based image segmentation. We scan each line in the road network. On each side of a line, a search space is defined as a rectangular area, which should be wide enough to cover potential road regions. The distance from all boundary pixels in the search space to the line are assigned to bins in a histogram. The bin width is chosen based on image resolution. The histogram reveals the spatial distribution of boundary pixels with respect to the line. In aerial images, we can observe that actual road edges generally have long detected boundaries parallel to lines in a road network. Hence, the road edge is determined as a straight line at the distance corresponding to the highest peak in the histogram. Fig. 7 shows an example, where the histogram peaks well match actual road edges. As the lines in the road network may not be at the centers of roads, the road edge on each side of a line segment is estimated separately. Previously, in [24] it was shown that this method exhibits reliable performance with manually generated road networks. Our experiments described in the next section demonstrate that this method performs well with road networks automatically generated from GPS traces.

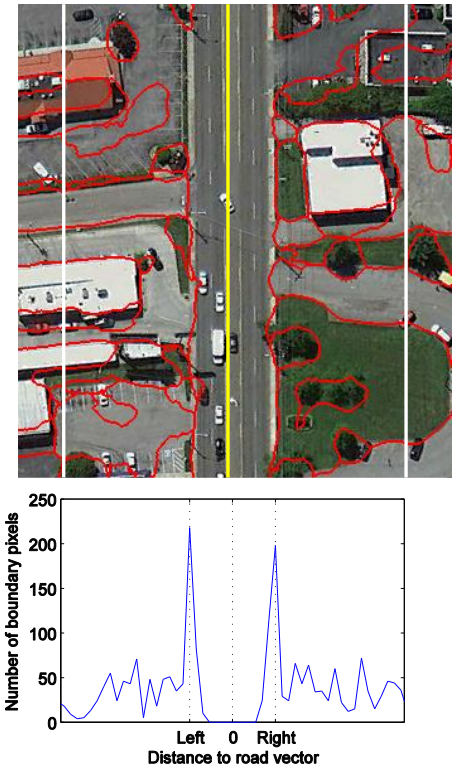


Fig. 7 Determining road edges based on the histogram of boundary pixels. The yellow line shows a line in the road network. The boundaries marked in red are generated by the factorization-based algorithm. The white lines indicate the search space on each side of the road.

6 Experiments

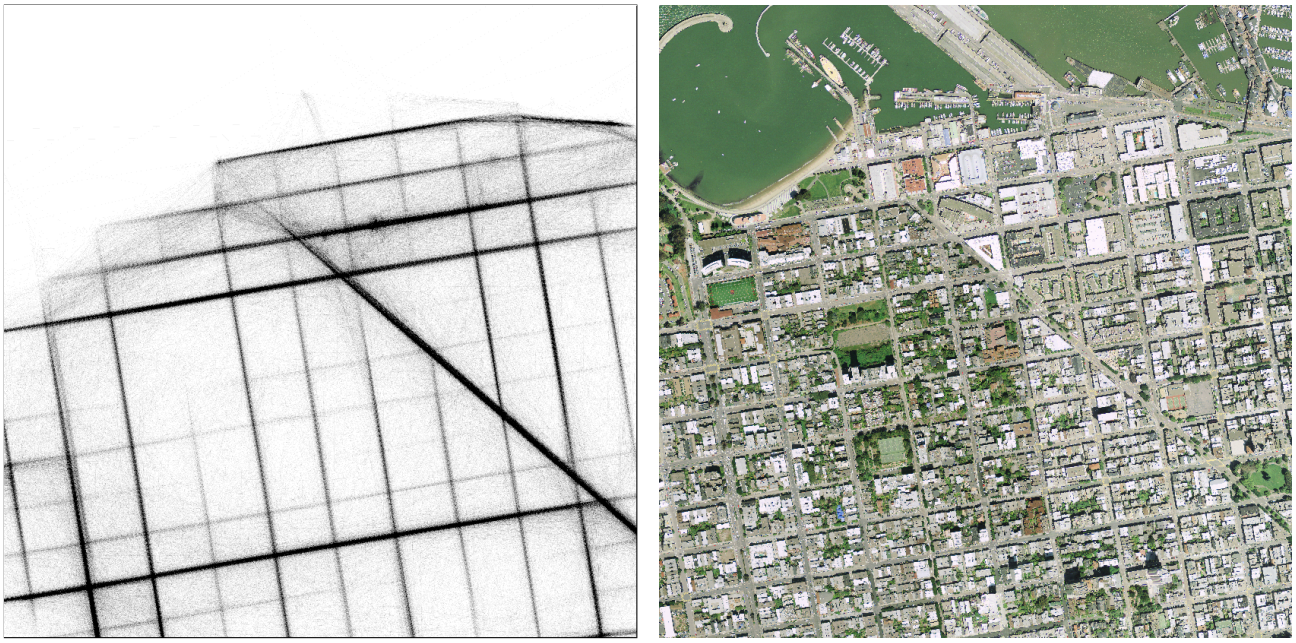
In this section, we present experimental results of applying the proposed method to a large dataset. We also quantitatively evaluate the results and compare with the leading method of inferring road maps from GPS traces.

6.1 Dataset

We use the GPS trace data of taxi cabs in San Francisco, CA [15]. It contains the GPS coordinates of over 500 taxis in one month. In the dataset, the traces in the downtown areas are highly prone to measurement errors due to tall buildings. The time intervals between two sample points are varying, most of which are 60 seconds, and a considerable amount has even longer intervals. A vehicle can pass several different roads within such long intervals. As a pre-processing to reduce noisy traces, we remove trace segments with average speed exceeding 150 km/h. We also remove trace segments shorter than 15 meters, which appear to be very noisy. We use two geo-referenced color images covering the same areas, each of which is a 5000×5000 tile. The spatial resolution is 0.3 m. Figs. 8 and 9 show the images and the corresponding traces, which respectively correspond to a residential area, where the traces are relatively less noisy, and the downtown area, where the traces are the most severely affected by noisy measurements.

6.2 Road network

We feed the datasets into the proposed method to generate road networks. The parameter values are fixed in our experiments. The results are illustrated in Figs. 10(a) and 11(a), where the road network are overlaid on the aerial images. For the dataset in Fig. 8, although the density of traces on different roads varies to a large extent, our method produces a rather complete road network. Roads in the dataset in Fig. 9 are highly difficult to extract from either GPS data or the image. Many traces corresponding to roads are completely buried in noise. Road regions in the image are occluded by vehicles and shadows. It can be seen that our method generates very promising results by exploiting the information from both data sources. We observe that some curvy roads are missing in the results. Because GPS data is collected at a low frequency, most traces on



(a) (b)

Fig. 8 The dataset covering a residential area. (a) GPS traces. (b) Aerial image.

(a) (b)

Fig. 9 The dataset covering the downtown area. (a) GPS traces. (b) Aerial image.

curvy roads lie outside road regions and hence are removed as noisy.

Previous efforts have been made in generating road networks from GPS data. Biagioni and Eriksson recently proposed a method that combines several existing techniques and report the start-of-the-art performance [4]. For comparison, we apply their method, which will be referred to as the BE method, to our dataset. We use the code distributed by the authors.

For quantitative evaluation, we use two indices, completeness and correctness [22], which are commonly used to assess road network extraction. The indices are calculated through a two-step matching. First, a buffer with a constant width is defined around the ground truth road. The extracted roads within the buffer are

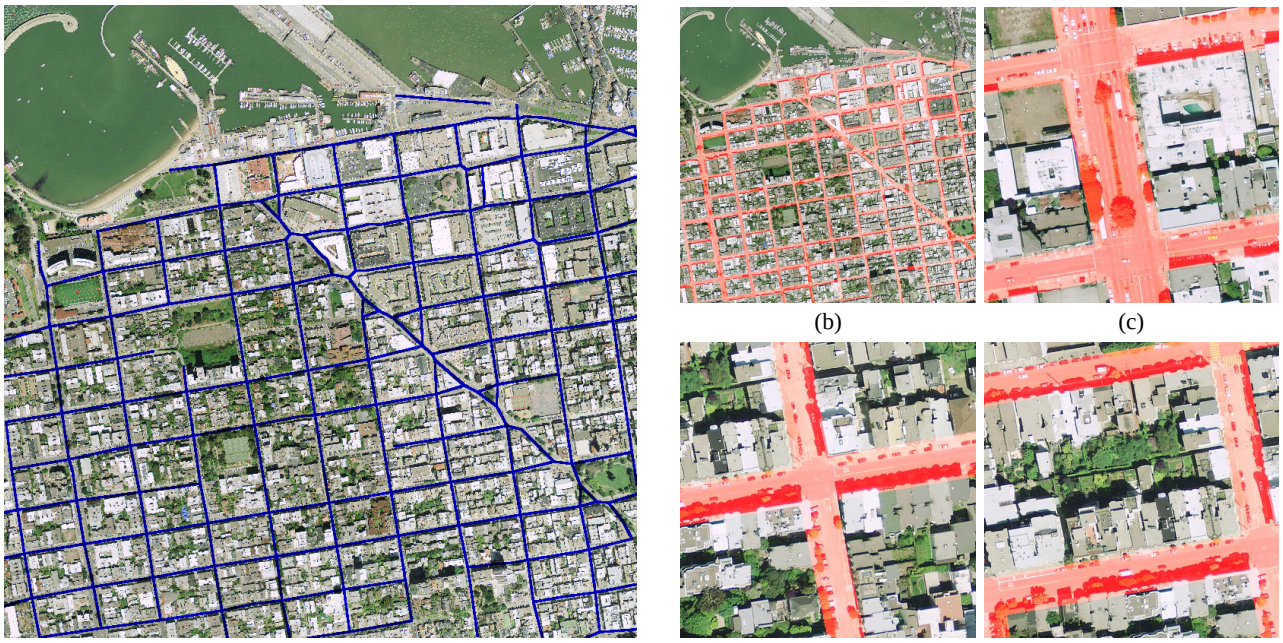


Fig. 10 Road network and road regions generated from the dataset in Fig. 8. (a) Road network (blue lines) overlaid on the aerial image. (b) Road regions marked in red. (c)-(e) Detailed views of extracted road regions.

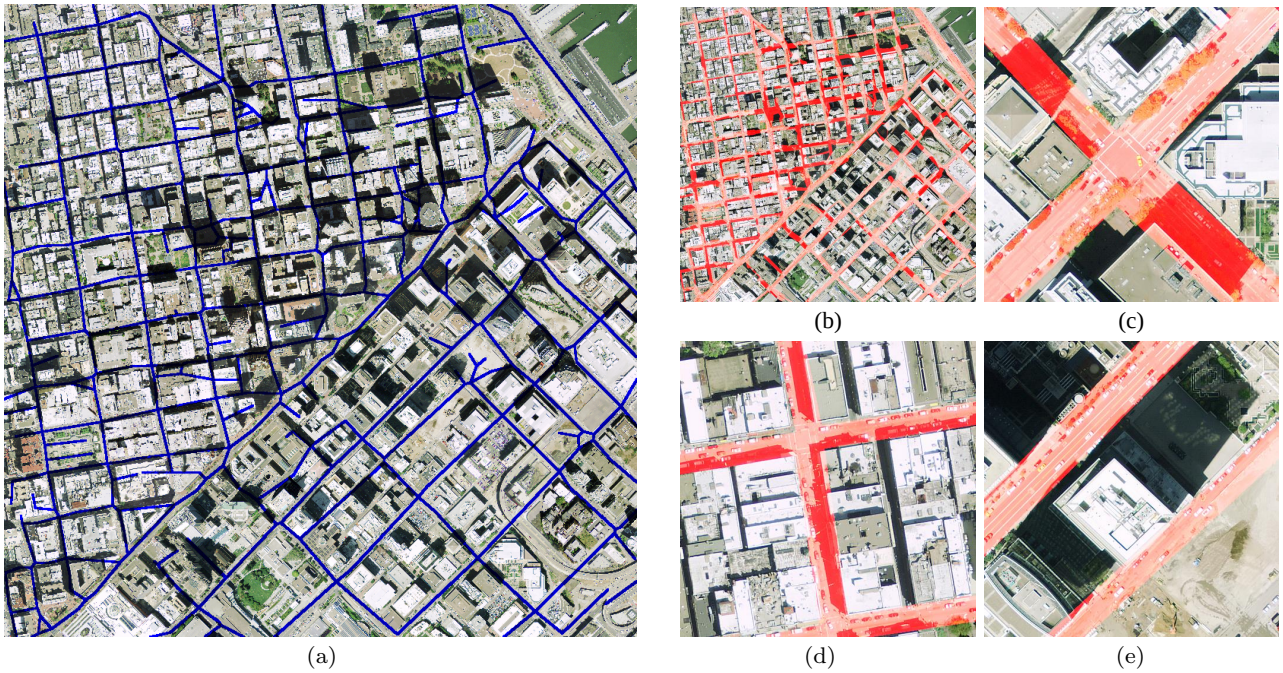


Fig. 11 Road network and road regions generated from the dataset in Fig. 9. (a) Road network. (b) Road regions. (c)-(e) Detailed views of extracted road regions.

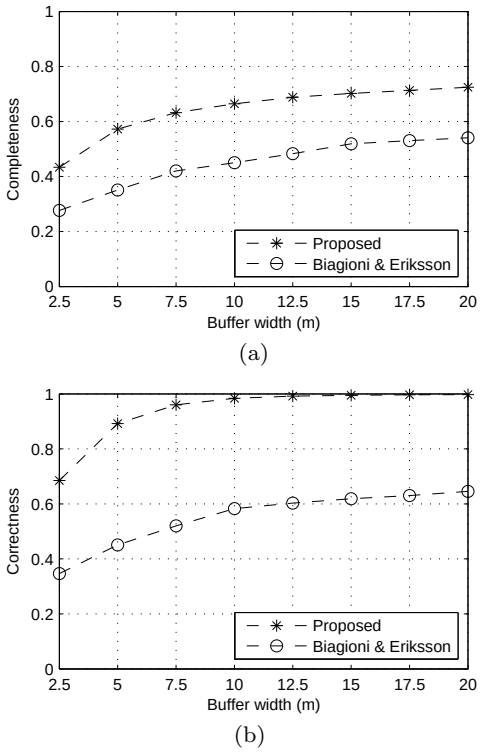


Fig. 12 Comparison between the Biagioni & Eriksson method and the proposed method on the dataset shown in Fig. 8. (a) Completeness plot. (b) Correctness plot.

denoted as true positive (TP), and the extracted roads out of the buffer false positive (FP). In the second step, a buffer is placed around the extracted road, and the ground truth road out of the buffer are denoted as false negative (FN). Completeness and correctness are defined by the following equations

$$\begin{aligned} completeness &= \frac{TP}{TP+FN} \\ correctness &= \frac{TP}{TP+FP}. \end{aligned} \tag{5}$$

For ground truth, we use the road vector data acquired from OpenStreetMap[1].

Fig. 12 shows the quantitative comparison of applying two methods to the dataset in Fig. 8. The two indices are calculated using different buffer widths. Clearly, our method achieves higher scores for both completeness and correctness. The BE method constructs an initial road network based on the trace density and applies map matching to the network to remove the edges with very few matched traces. When applied to this dataset, it confuses many crowded noisy traces as roads, which causes the low correctness rate. Also, because the initial network includes a large number of incorrect edges, the method matches the traces to those edges and misses the edges corresponding to actual roads, which decrease the completeness rate. In contrast, our method benefits from the use of image information and identifies much more roads with very few false detections. Note that some roads are not detected due to the lack of quality GPS traces on those roads. The correctness rate of our result is particularly high. The reason is that the detected roads are actually verified by both GPS and image information. We have also applied the BE method to the dataset in Fig. 9, but found that it fails to produce a reasonable result.

[1] http://www.openstreetmap.org/

6.3 Road regions

Based on the road networks, we produce road regions for both datasets, as shown in Figs. 10(b) and 11(b). Several patches in both figures are also presented to give detailed views of extracted road regions. We can see that the road regions are accurately delineated. To quantitatively measure accuracy, we create ground truth by manually labeling road regions on images with assistance of road vector data. As a typical binary classification task, we measure the extraction accuracy using precision and recall. Precision is the percentage of the correctly detected road pixels among those detected by the algorithm, and recall the percentage of the correctly detected road pixels among those in ground truth. The average precision and recall for the result in Fig. 10(b) are 0.81 and 0.73, respectively. For the result in Fig. 11(b), they are 0.78 and 0.68, respectively.

We implement the entire method using MATLAB. On a 3.2-GHz Intel processor, the running time for processing the two sets of data is 8 minutes and 12 minutes, respectively.

6.4 Visualization applications

The results of our methods can be transferred to useful visualization. One simple example is to assign a constant road surface color to road regions, which creates a view of empty streets, as shown in Fig. 13.

By incorporating trace data, we can also provide traffic pattern visualization. After obtaining the road network and road regions, we project each GPS sample onto the closest line in the road network. We compute the shortest path between two consecutive samples and accumulate the trips on the road network. The 1-D representation is then dilated to the segmented road regions. Fig. 14 illustrates the results of counting the taxi trips within three time intervals in each day. Three cropped areas are shown, where the roads colored based

Fig. 13 Empty street view. (a) Original image. (b) Road regions filled with a single color.

on trip numbers are overlaid on gray-level images. This provides a clean and informative visualization of taxi traffic, where further analysis can be conducted, e.g., travel time estimation and hotspot detection.

7 Conclusions

We have presented a new method that integrates aerial images with GPS trace data for road network inference and road segmentation. Applying computer vision techniques to aerial images offers valuable information to remove noisy traces while preserving the useful ones. The road network generated from filtered traces provides guidance to accurately delineated road regions. Our experiments demonstrate that our method can use GPS data with high level noise to segment road regions in highly complex scenes. Moreover, we show that our method leads to useful visualization.

There are several directions for future work. First, the current method does not consider fine details of road networks, such as separated lanes, ramps, and overpasses. One reason is that the coarse-grained trace data cannot provide such detailed information, which, however, may be available from high-resolution images. It is interesting to explore how to utilize images to refine generated road networks. Secondly, this paper focuses on the GPS data of vehicles. In future research, we plan to investigate the use of other types of GPS data, e.g., smartphone location data, which may be capable of characterizing semantic regions other than roads when combined with images.

Acknowledgment

This work was supported in part by U.S. Department of Energy/National Nuclear Security Administration under Grant DOE-NNSA/NA-22. This manuscript has been authored by employees of UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. Accordingly, the United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

1. Albright, R., Cox, J., Duling, D., Langville, A., Meyer, C.: Algorithms, initializations, and convergence for the nonnegative matrix factorization. NCSU Technical Report Math 81706 (2006)
2. Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., Heinrich, E.: Automatic road extraction based on multi-scale, grouping, and context. Photogrammetric Engineering & Remote Sensing **65**, 777–785 (1999)
3. Biagioni, J., Eriksson, J.: Inferring road maps from GPS traces: survey and comparative evaluation. In: Transportation Research Board, 91st Annual (2011)
4. Biagioni, J., Eriksson, J.: Map inference in the face of noise and disparity. In: ACM SIGSPATIAL GIS (2012)
5. Brox, T., Weickert, J., Burgeth, B., Mrázek, P.: Nonlinear structure tensors. Image and Vision Computing **24**, 41–55 (2006)
6. Cao, L., Krumm, J.: From GPS traces to a routable road map. In: ACM SIGSPATIAL GIS (2009)
7. Doucette, P., Agouris, P., Stefanidis, A.: Automated road extraction from high resolution multispectral imagery. Photogrammetric Engineering & Remote Sensing **70**, 1405–1416 (2004)
8. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer **10**, 112–122 (1973)
9. Gaetano, R., Masi, G., Poggi, G., Verdoliva, L., Scarpa, G.: Marker-controlled watershed-based segmentation of multiresolution remote sensing images. IEEE Transctions on Geoscience and Remote Sensing **53**, 2987–3004 (2015)
10. Golub, G., Loan, C.V.: Matrix Computation. The Johns Hopkens University Press (1996)
11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conferencel (1988)
12. Heitz, G., Koller, D.: Learning spatial context: using stuff to find things. In: ECCV (2008)
13. Liu, X., Wang, D.L.: A spectral histogram model for texton modeling and texture discrimination. Vision Research **42**, 2617–2637 (2002)
14. Mena, J.B., Malpica, J.A.: An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. Pattern Recognition Letters **26**, 1201–1220 (2005)
15. Piorkowski, M., Sarafijanovoc-Djukic, N., Grossglauser, M.: A parsimonious model of mobile partitioned networks with clustering. In: The First International Conference on COMmunication Systems and NETworkS (COMSNETS) (2009)
16. Porway, J., Wang, K., Zhu, S.C.: A hierarchical and contextual model for aerial image understanding. In: CVPR (2008)
17. Poullis, C., You, S.: Delineation and geometric modeling of road networks. ISPRS Journal of Photogrammetry and Remote Sensing **65**, 165–181 (2010)

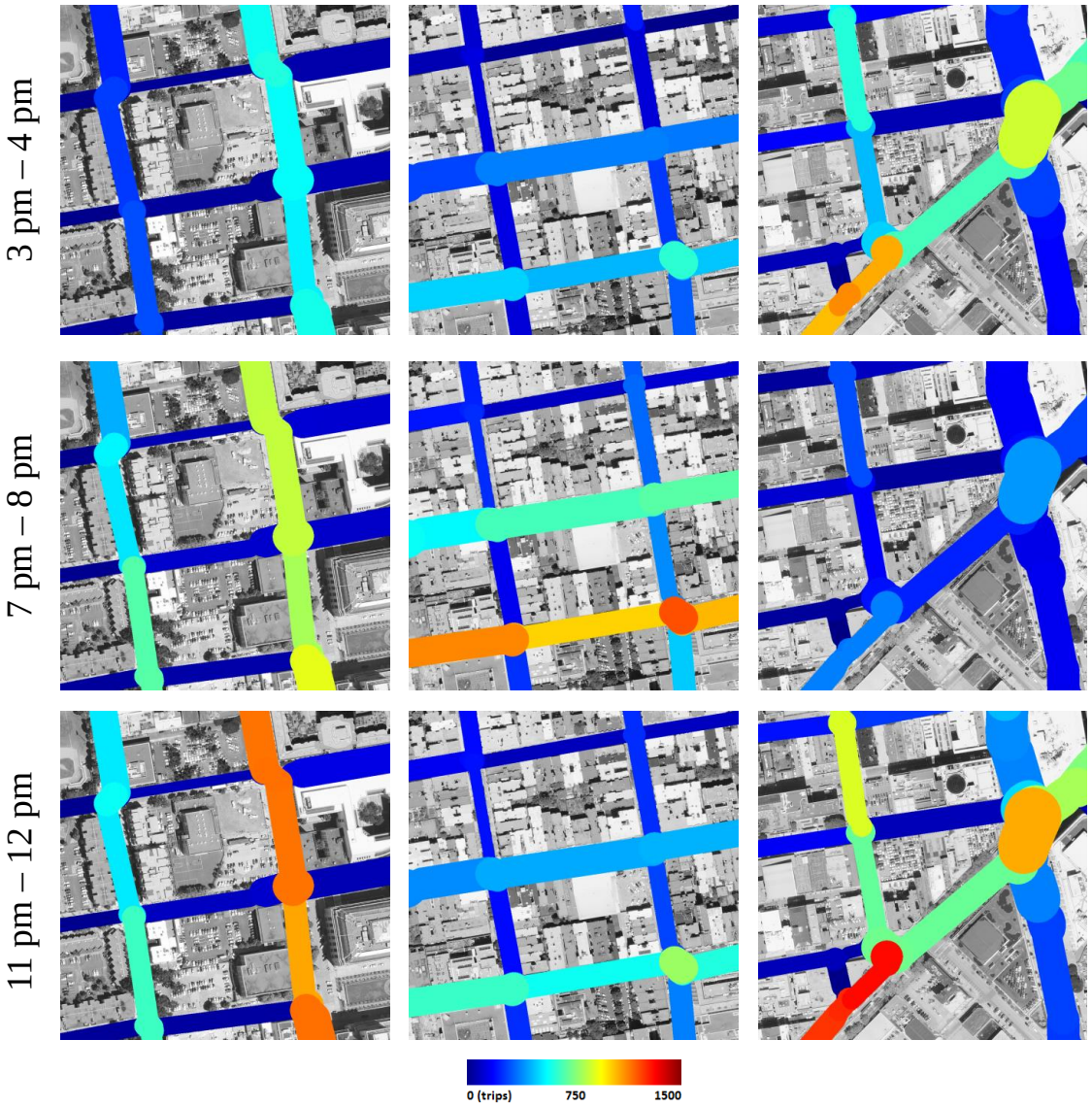


Fig. 14 Visualization of taxi traffic at 3 different locations in San Francisco city. Columns and rows correspond to locations and time intervals, respectively. The number of taxi trips is indicated by different colors on road regions, which is superimposed on the gray-level aerial image. Road network and region extraction enable effective visualization of GPS trace signals for better understanding of traffic patterns.

18. Seo, Y.W., Urmson, C., Wettergreen, D.: Exploiting publicly available cartographic resources for aerial image analysis. In: ACM SIGSPATIAL GIS (2012)
19. Seo, Y.W., Urmson, C., Wettergreen, D., Lee, J.W.: Augmenting cartographic resources for autonomous driving. In: ACM SIGSPATIAL GIS (2009)
20. Shi, Q., Du, B., Zhang, L.: Spatial coherence-based batch-mode active learning for remote sensing image classification. IEEE Transactions on Image Processing **24**, 2037–2050 (2015)
21. Torre, F., Pitchford, D., Brown, P., Terveen, L.: Matching GPS traces to (possibly) incomplete map data: bridging map building and map matching. In: ACM SIGSPATIAL GIS (2012)
22. Wiedemann, C., Heipke, C., Mayer, H., Jamet, O.: Empirical evaluation of automatically extracted road axes. In: Empirical Evaluation Techniques in Computer Vision (1998)
23. Yuan, J., Cheriyadat, A.M.: Image driven GPS trace analysis for road map inference. In: ACM SIGSPATIAL GIS (2013)
24. Yuan, J., Cheriyadat, A.M.: Road segmentation in aerial scenes by exploiting road vector data. In: International Conference on Computing for Geospatial Research and Applications (2013)
25. Yuan, J., Gleason, S.S., Cheriyadat, A.M.: Systematic benchmarking of aerial image segmentation. IEEE Geoscience and Remote Sensing Letters **10**, 1527–1531 (2013)

26. Yuan, J., Wang, D.L.: Factorization-based texture segmentation. Technical Report OSU-CISRC-1/13 -TR01 (2013)
27. Yuan, J., Wang, D.L., Wu, B., Yan, L., Li, R.: LEGION-based automatic road extraction from satellite imagery. IEEE Transctions on Geoscience and Remote Sensing **49**, 4528–4538 (2011)