Road Segmentation in Aerial Images by Exploiting Road Vector Data

Jiangye Yuan

Computational Sciences and Engineering Division

Oak Ridge National Laboratory

Oak Ridge, Tennessee 37831

Email: yuanj@ornl.gov

Anil M. Cheriyadat

Computational Sciences and Engineering Division

Oak Ridge National Laboratory

Oak Ridge, Tennessee 37831

Email: cheriyadatam@ornl.gov

we don't have this data available

Abstract—Segmenting road regions from high resolution aerial images is an important yet challenging task due to large variations on road surfaces. This paper presents a simple and effective method that accurately segments road regions with weak supervision provided by road vector data, which is publicly available. The method is based on the observation that in aerial images road edges tend to have more visible boundaries parallel to road vectors. A factorization-based segmentation algorithm is applied to an image, which accurately localizes boundaries for both texture and nontexture regions. We analyze the spatial distribution of boundary pixels with respect to the road vector, and identify the road edge that separates roads from adjacent areas based on the distribution peaks. The proposed method achieves on average 90% recall and 79% precision on large aerial images covering various types of roads.

I. INTRODUCTION

This paper presents a new method to segment road regions in aerial images. The segmentation is supervised by publicly available road vector data. As shown in Fig. 1, the proposed method accurately delineates the road regions in a complex aerial scene.

With the advances made in remote sensing data acquisition, large volumes of high-resolution aerial images have been collected, which pose a significant challenge to image analysis and understanding. One important analysis task is to segment road regions from images, which has a wide range of valuable applications. The resulting road map can be used for establishment and update of geographic information systems. Knowledge of road regions also provides contextual information that benefits many image analysis tasks. It has been shown that incorporating the information of road extent gives a clear improvement on detecting vehicles [8] and capturing spatial relations among objects [15].

However, it is challenging to accurately identify road regions from high resolution images. In images with sub-meter resolutions, road region appearances vary vastly. In addition to the pavement materials and markings that cause the appearance variations, road regions can be largely covered by vehicles, vegetations, and shadows, especially in urban and suburban scenes. A multitude of methods for extracting road regions have been proposed. 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 [2], [5]. The road model can
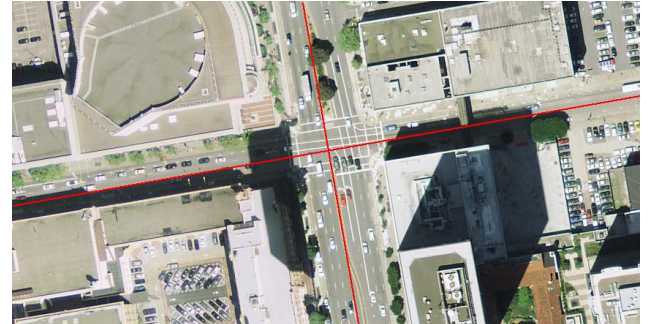
(a)
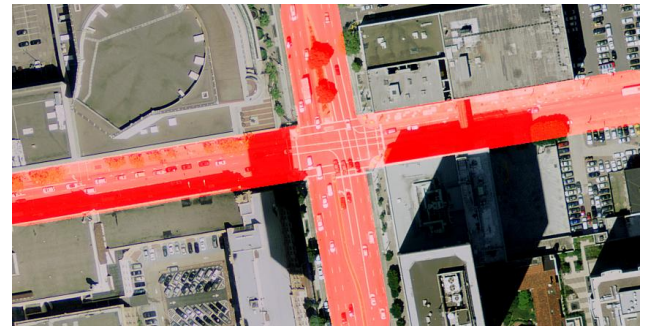
(b)

Fig. 1. An example of our road segmentation results. (a) An aerial image with road vectors overlaid. (b) The segmented road region.

be predefined or learned from labeled data. However, this assumption can be substantially violated in complex scenes, and thus those methods have difficulty achieving a reliable performance on large real-world datasets [11].

While one purpose of segmenting road regions from remote sensing images is to generate road vector data [12], the importance of this purpose is decreasing as road vector data are now widely available from various online cartographic resources such as Google Maps[1] and US Census TIGER/Line files[2]. Moreover, with the emergence of Volunteered Geographic Information (VGI), which allows common users to create and edit geographic data, the availability of road vector data is growing rapidly. For example, one of the most extensive VGI

---

[1]http://maps.google.com/

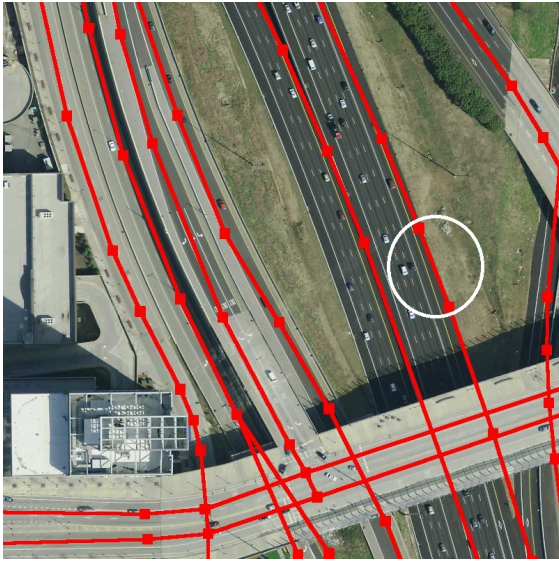[2]http://www.census.gov/geo/maps-data/data/tiger.html

Fig. 2. An example of road vector data from OSM. Road vectors are overlaid on the aerial image. Squares are the end points of line segments. The white circle marks a line segment that largely deviates from the road centerline

sources is OpenStreetMap (OSM), which has millions of contributors [7]. Since the vector data can be considered as labeled data, where the pixels corresponding to road vectors are known to be road, they can provide supervision for segmenting road regions. However, how to utilize vector data for supervised road segmentation is not a trivial problem.

In vector data, a road vector specifies the end points of line segments passing through road centerlines. Connected line segments forms road networks. Fig. 2 shows an example of OSM road vector data overlaid on the aerial image. Due to various types of errors, the vectors often deviate from the actual road centerlines in images, which can be observed in the figure. Given the noise in the data together with varying road width, setting a predefined width along vectors cannot correctly identify road regions. Another possible solution is to select pixels that have similar features to labeled road pixels. A straightforward approach is to segment an image and select the segments that overlap with vector data. Fig. 3 presents the results of applying such a approach to the image in Fig. 1(a), where we adopt a leading superpixel segmentation algorithm [9] to generate over-segmentation with each segment representing a compact and homogeneous region. Two results are shown, obtained from segmentations with different numbers of superpixels. We can see that the results either miss large road regions or contains many non-road regions due to the highly varying appearances.

Instead of constructing or learning a road model, the method proposed in this paper relies on a very basic but distinctive feature of roads – parallel road edges. The distribution of the boundary locations with respect to road vectors are exploited to identify road edges. We adopt a recently proposed factorization-based segmentation algorithm [18] to provide segment boundaries that account for both texture and nontexture regions. We find that this seemingly oversimplified method produces highly accurate results for large images containing complex road structures.
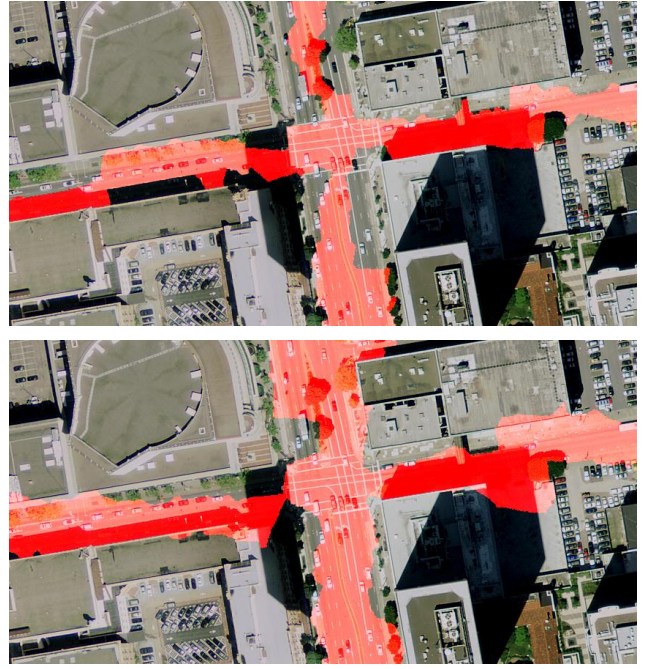

Fig. 3. Illustration of a straightforward approach for supervised road segmentation. The road regions are marked in red. The top image shows the result from a 200-superpixel segmentation, the bottom from a 100-superpixel segmentation

II. PREVIOUS WORK

Several attempts have been made in using road map data to assist road extraction from remote sensing images. Mnih and Hinton [14] generate labeled datasets from road vector data and satellite images, which are used to train a neural network to detect road pixels. Their approach can detect roads with moderate occlusions thanks to the availability of large training datasets and the learning ability of neural networks. However, as their results show, the method fails when the occlusions are large. The work presented in [17] has a problem formulation similar to our work. Instead of road vector data, screen shots of Google Maps are used in that work for supervised road region segmentation. A superpixel image is first generated. Based on the superpixels that overlap with the roads in the map, a probabilistic classifier is learned, which is combined with Markov Random Field (MRF) to identify road regions. Because this method models road appearances from narrow areas in the roads, it can misclassify those non-road regions with similar appearances. In addition, this method is computationally expensive due to parameter estimation in MRF.

It is a common practice for road extraction methods to start from contour detection or segmentation outputs [13], [16], [19]. Using the detected boundaries or segments as basic units offers several benefits, including reduced spectral variability and more spatial and contextual information. However, extraction results can be largely affected by the quality of the results from this step, which is often difficult to ensure for large images. Besides, more accurate results require advanced algorithms, which tend to have a high computational cost. In this paper, we employ a factorization-based segmentation
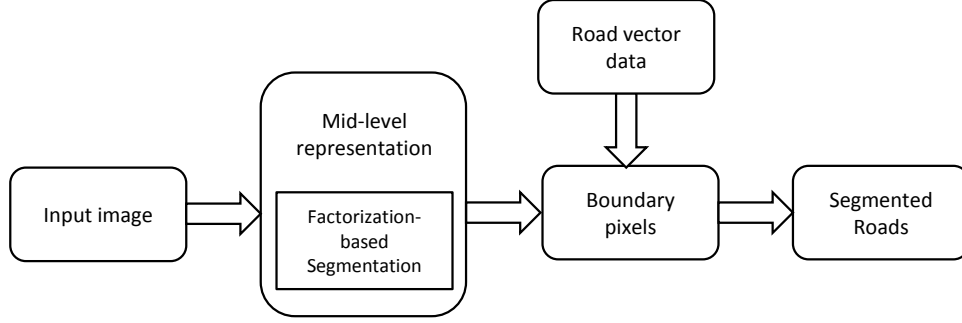
Fig. 4. Flow diagram of the proposed method.

algorithm [18], which can efficiently produce segments for both texture and nontexture regions with well-localized boundaries. Moreover, in contrast to those methods that directly build on the resulting boundaries or segments, we find road edges based on the spatial distribution of detected boundaries, which reduces the dependence on segmentation accuracy.

III. SUPERVISED ROAD SEGMENTATION

This section presents the proposed method for road segmentation, which is summarized in Fig. 4. The segmentation algorithm is first applied to an image to generate a mid-level representation that gives object boundaries. Given road vector data, the relative locations of boundary pixels are examined to determine two road edges that define road regions.

A. Boundary detection

The first step in our method is to find boundaries in the images that should include most boundaries separating roads and adjacent regions and but contain as few as possible the noisy ones that appear inside meaningful regions. We utilize a factorization-based segmentation algorithm [18]. This algorithm uses local spectral histograms [10] as features. At each pixel location, the feature is a concatenated histogram of different filter responses within a local window. The size of the window is referred to as integration scale. Each feature can be regarded as a linear combination of several features representative of different regions, and combination weights indicate the region ownership of the corresponding pixel. Consequently, a feature matrix $\mathbf{Y}$ can be expressed as a product of two matrices,

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \qquad (1)$$

Here each column in $\mathbf{Y}$ is the feature at each pixel location, each column of $\mathbf{Z}$ is a representative feature for a region, and each column of $\boldsymbol{\beta}$ is the combination weights at each pixel location. $\boldsymbol{\varepsilon}$ is the noise. Segment labels are given by $\boldsymbol{\beta}$, where the largest weight in each column indicates the region the corresponding pixel belongs to.

In such a formulation, the segmentation algorithm seeks to factor $\mathbf{Y}$. By applying singular value decomposition to $\mathbf{Y}$, the number of segments can be estimated, and a subspace can be constructed that contains all features. Clustering features in the subspace produces initial representative features, which are fed into a nonnegative matrix factorization algorithm [1] to obtain the factored matrices. Fig. 5 shows the result of applying this algorithm to a 1000×1000 aerial image. We can see that the major objects like buildings and roads are segmented with well localized boundaries. It takes 10 seconds to segment this image on a 3.2-GHz Intel processor.

B. Road segmentation supervised by road vectors

The road regions are confined by two road edges parallel to road centerlines. Therefore, given the vector data, segmenting road regions can be solved by determining two parallel road edges. Since in most cases road vectors do not lie on the exact road centerlines in images, two road edges need to be determined separately.

From an aerial view, we can observe parallel road edges formed by the contrast between roads and other neighboring objects. This is also one of the most important cues a human operator would use to delineate roads. Based on this observation, we define road edges as lines parallel to road vectors and aligned with most segment boundaries. This definition, overlooking many other cues that could be exploited for road extraction though, leads to an effective approach for segmenting road regions.

On each side of a road vector, we define a search space, which is a rectangular area that is sufficiently large to cover potential road regions. We compute the distance from each boundary pixel in the search space to the line segment in vector data, and construct a histogram by assigning all the distances to equal-width bins. The bin width is chosen based on image resolution. The road edge, a straight line, is at the distance corresponding to the highest peak in the histogram. Fig. 6 shows an example of constructing a histogram of boundary pixels. The boundaries in the image are produced by the segmentation algorithm. The yellow line represents the road vector. We can see that the histogram peak on each side reveals the location of a road edge. Note that this method does not require a highly accurate boundary detection with each meaningful region delineated. Because it is based on the distribution of boundary pixels, the result is not sensitive to boundary detection errors, as long as sufficient major boundaries are identified.
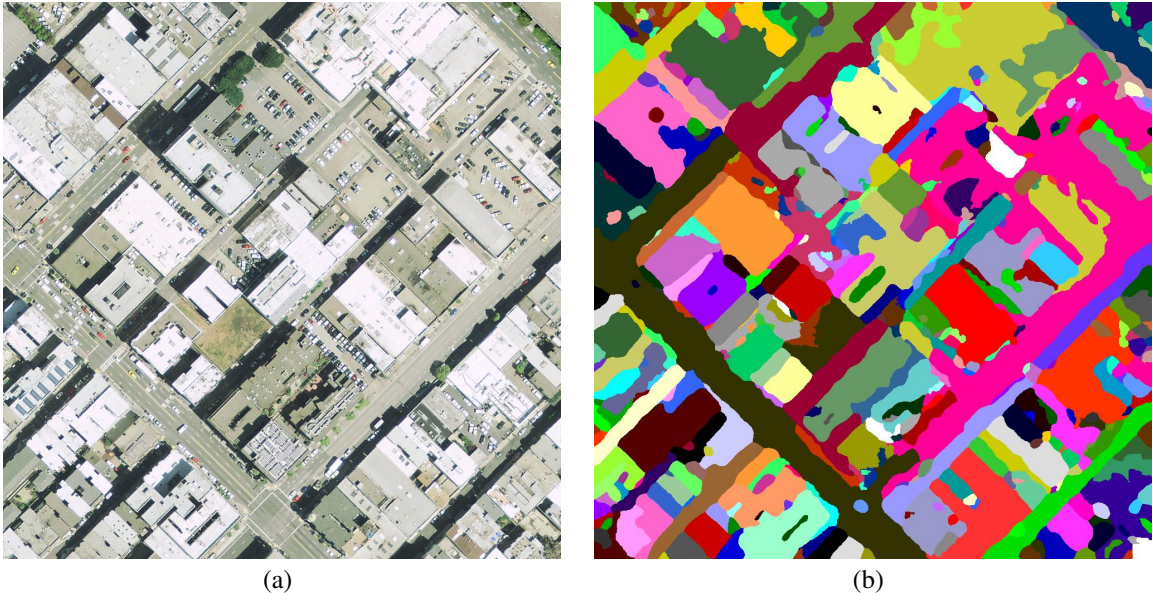
(a)　　　　　　　　　　　　　　　(b)

Fig. 5.　Example of segmentation results. (a) An aerial image of size 1000×1000 pixels. (2) The segmentation result with each segment labeled by a random color.
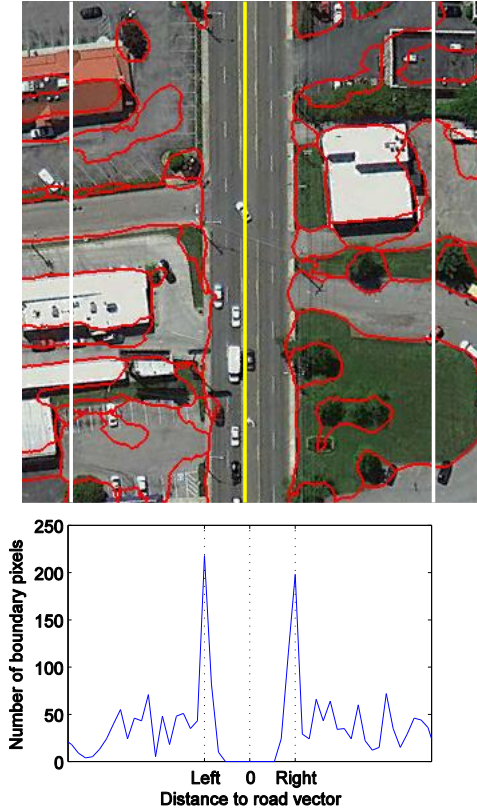

Fig. 6.　Illustration of computing the histogram of boundary pixels. The yellow line shows the road vector. The boundaries marked in red are generated by the factorization-based algorithm. The white lines indicate the search space on each side of the road.

For each line segment, the detected road edges have the same length, which give a rectangular road region. On both ends of the road, we include the semicircular region within the diameter equal to the rectangle width, in order to smoothly connect roads. In some cases, aggregated noisy boundaries can result in a histogram peak at a place that does not correspond to actual road edges. To address this problem, we compute the orientation of each boundary pixel and exclude the boundary pixels with orientations significantly different from the vector orientation. The orientation of a boundary pixel is determined by the direction most neighboring pixels reside in.

IV. EXPERIMENTS

We test our method on two 5000×5000 geo-referenced color images covering different cities with a spatial resolution of 0.3 m. Each image contains various types of roads with highly diverse appearances. The corresponding road vector data are acquired from OSM, which are stored as shapefiles.

For all experiments, we use a fixed set of parameter values. In the segmentation algorithm, we apply two Laplacian of Gaussian filters to the red band and use the filter responses together with three color bands to compute local spectral histograms. The integration scale is set to 21×21. The bin width for computing histograms of boundary pixels is set to 5 pixels, corresponding to 1.5 meters on the ground. The search space needs to be sufficiently large. However, as the road width varies in a wide range, a large search space that accommodates multi-lane highways may include more than one road in residential areas, which causes inaccurate detection. To alleviate this problem, we use two sizes of search spaces for different road classes. In the OSM data, each road has a classification tag. The search space is 30 meters from road vectors for the classes of 'motorway', 'truck', 'primary', 'secondary', and 'tertiary', and 15 meters for the other classes. Note that a search space is defined on each side of the road vector.

Fig. 7. Road segmentation results. The segmented roads are marked in red in the original images. The areas in white rectangles will be displayed in a larger view.

(a)   (b)   (c)

(d)   (e)   (f)

Fig. 8. Image patches corresponding to the rectangular areas in Fig. 7. The left column presents the image patches from the top image, and the right column from the bottom image. The size of each image is 400×600. This figure is best viewed in color.

Fig. 7 show the results of applying our method to two images, where the road regions are shown in red and overlaid on the original images. As we can see, the images cover complex scenes that are very challenging for road extraction. The resulting road regions are highly accurate from a visual inspection. We implement the method using MATLAB. The average running time for processing an image is 12 minutes.

To better show the quality of results, we display six image patches from the images in Fig. 7, where the corresponding areas are indicated by white rectangles. These examples demonstrate that our method is capable of segmenting roads with heavily shaded areas (Fig. 8(a) and (d)), dense vehicles (Fig. 8(a)), large vegetation coverage (Fig. 8(b) and (e)), and complex road markings (Fig. 8(f)). In Fig. 8(c), the roads are bordered by a large parking lot, which is a difficult situation for segmenting road regions because of similar spectral characteristics. In our result, two regions are accurately separated based on the boundary cue in the surroundings. In Fig. 8(e), builds and roads are highly occluded by trees. This scene is very difficult for segmentation algorithms to obtain meaningful results. Fig. 9 shows the search space of the horizontal road in the middle of the image. Although the boundaries from segmentation are rather noisy, the peaks in the histogram clearly indicate the locations of road edges.

To quantitatively evaluate the results, we generate ground truth by manually labeling road regions on the images. Since identifying road regions is a typical binary classification task, we use the precision and recall measures. Precision is the percentage of the true positive among the road regions detected by the algorithm, and recall the percentage of the true positives in the ground truth. The average precision and recall for the results in Fig. 7 are 0.79 and 0.90, respectively.

There are some cases where the method does not perform well. Some examples are illustrated in Fig. 10. The misplaced
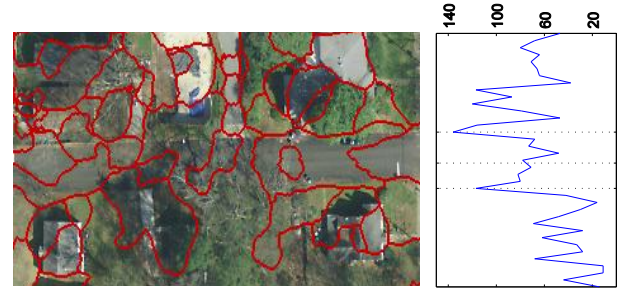

Fig. 9. Boundary pixel histogram of the middle road in Fig. 8(e).

road edges are mostly caused by elongated buildings or shadows that happen to show long boundaries that parallel to roads. Such misplacements cannot be easily corrected based on low-level information. Fortunately, these situations do not occur very often in an image.

Because our method analyzes the segment boundaries to extract roads, a very large number of contour detection and segmentation methods can be used to provide boundary pixels. For comparison purpose, we investigate three methods detecting boundaries at different levels, including Canny edge detector [4], straight line extraction [3], and the graph-based region merging algorithm (Felz-Hutt) [6].

Canny edge detector is a classic approach to produce boundaries. Due to its simplicity and efficiency, it is still widely used for boundary detection. The results based on Canny detector can be considered as a baseline.

In the straight line extraction method, spatially contiguous pixels with the same quantized orientations form line supporting regions. The orientations are quantized with different bin
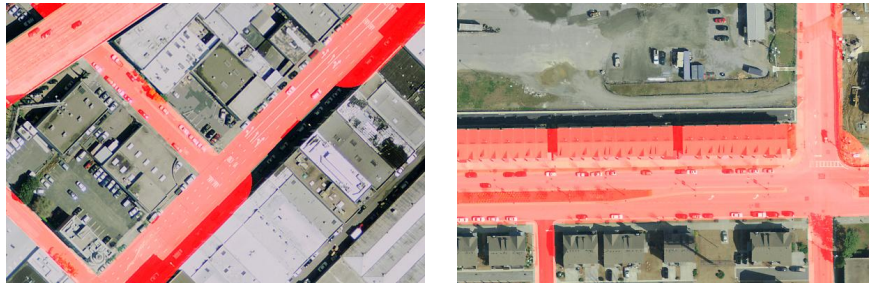
Fig. 10. Examples where the proposed method does not perform well due to multiple parallel edges.

centers to reduce the chance of incorrectly partitioning regions. Different line region partitions are integrated to extract straight lines by using a pixel voting scheme. Since road edges are supposed to be straight, this method is expected to generate candidate boundary pixels with less noise.

Based on the view that a segment is a connected component in a graph, the Felz-Hutt algorithm defines the differences within a component (*internal difference*) and between two components (*difference between*), and iteratively merges components whose difference between is smaller than their internal differences. This algorithm is very efficient and has shown promising performance.

Fig. 11 shows the average precision and recall of the results from each method. The F-measure is also presented as a summary score, which is the harmonic mean of precision and recall. Canny detector gives the highest recall but with a very low precision. It often fails to detect low contrast boundaries on actual road edges and thus results in placing road edges at some off-road objects with high-gradient edges. In spite of a gradient-based method, straight line extraction gives much improved results because it finds straight lines on road edges and at the same time eliminates many noisy boundaries that can distort the histogram of boundary pixels. The Felz-Hutt algorithm attains a slightly higher F-measure. It removes more noise boundaries but still generates boundaries of aligned vehicles or certain road markings that can negatively affect the final result. The factorization-based algorithm achieves the best F-measure thanks to the effective use of texture information that helps identify the boundaries with high saliency.

V. CONCLUSIONS

We have presented a new method of supervised road segmentation. The supervision comes from road vector data, which are easily accessible. Despite the simple strategy, the method makes effective use of the vector data and accurately segments road regions. The method works reliably on two large datasets of challenging aerial scenes.

The current method requires a very weak supervision, lines on the road regions. In addition to road vector data, other forms of road position data can be employed. One candidate is the GPS traces from vehicles. Coupling our methods with such GPS data results in a system that can generate road maps in real-time without involving any manual work. However, to achieve this goal, we need to address the issues such as generating lines from raw GPS data and the high level of GPS noise occurring in urban areas. We are currently investigating these issues.
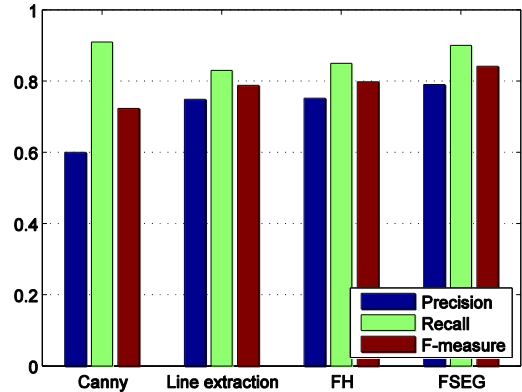

Fig. 11. A comparison of road segmentation results. FH stands for the Felz-Hutt algorithm, and FSEG the factorization-based algorithm.

REFERENCES

[1] R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. *NCSU Technical Report Math 81706*, 2006.

[2] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, and E. Heinrich. Automatic road extraction based on multi-scale, grouping, and context. *Photogrammetric Engineering & Remote Sensing*, 65:777–785, 1999.

[3] J. B. Bums, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *PAMI*, 8:425–455, 1986.

[4] J. Canny. A computational approach to edge detection. *PAMI*, 8:679–698, 1986.

[5] P. Doucette, P. Agouris, and A. Stefanidis. Automated road extraction from high resolution multispectral imagery. *Photogrammetric Engineering & Remote Sensing*, 70:1405–1416, 2004.

[6] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59:167–181, 2004.

[7] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7:12–18, 2008.

[8] G. Heitz and D. Koller. Learning spatial context: using stuff to find things. In *ECCV*, 2008.

[9] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, , and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, 7:12–18, 2008.

[10] X. Liu and D. L. Wang. A spectral histogram model for texton modeling and texture discrimination. *Vision Research*, 42:2617–2637, 2002.

[11] H. Mayer. Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63:213–222, 2008.

[12] J. B. Mena. State of the art on automatic road extraction for GIS update: a novel classification. *Pattern Recognition Letters*, 24:3037–3058, 2003.

[13] J. B. Mena and J. A. Malpica. 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.

[14] V. Mnih and G. Hinton. Learning to detect roads in high-resolution aerial images. In *ECCV*, 2010.

[15] J. Porway, K. Wang, and S. C. Zhu. A hierarchical and contextual model for aerial image understanding. In *CVPR*, 2008.

[16] C. Poullis and S. You. Delineation and geometric modeling of road networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65:165–181, 2010.

[17] Y.-W. Seo, C. Urmson, and D. Wettergreen. Exploiting publicly available cartographic resources for aerial image analysis. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2010.

[18] J. Yuan and D. L. Wang. Factorization-based texture segmentation. *Technical Report OSU-CISRC-1/13-TR01*, 2013.

[19] J. Yuan, D. L. Wang, B. Wu, L. Yan, and R. Li. LEGION-based automatic road extraction from satellite imagery. *IEEE Transctions on Geoscience and Remote Sensing*, 49:4528–4538, 2011.