

Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints

Jiuxiang Hu, Anshuman Razdan, John C. Femiani, Ming Cui, and Peter Wonka

Abstract—In this paper, a new two-step approach (detecting and pruning) for automatic extraction of road networks from aerial images is presented. The road detection step is based on shape classification of a local homogeneous region around a pixel. The local homogeneous region is enclosed by a polygon, called the *footprint of the pixel*. This step involves detecting road footprints, tracking roads, and growing a road tree. We use a spoke wheel operator to obtain the road footprint. We propose an automatic road seeding method based on rectangular approximations to road footprints and a toe-finding algorithm to classify footprints for growing a road tree. The road tree pruning step makes use of a Bayes decision model based on the area-to-perimeter ratio (the A/P ratio) of the footprint to prune the paths that leak into the surroundings. We introduce a lognormal distribution to characterize the conditional probability of A/P ratios of the footprints in the road tree and present an automatic method to estimate the parameters that are related to the Bayes decision model. Results are presented for various aerial images. Evaluation of the extracted road networks using representative aerial images shows that the completeness of our road tracker ranges from 84% to 94%, correctness is above 81%, and quality is from 82% to 92%.

Index Terms—Bayes decision rule, road extraction, road footprint, road tracking, road tree pruning.

I. INTRODUCTION

THE ABILITY of new sensors to provide fine-resolution imagery of urban areas increases the potential for automatic interpretation tools to extract and identify road networks. Road network detection can be used for several applications such as automated correction and updating for geographic information systems (GIS) from aerial images [1], [2], registration with multitemporal images for change detection [3]–[5], automatically aligning two spatial datasets [6]–[8], etc. These applications not only require detecting road networks but also need to identify road intersections [1], [6]. A road network is a graph-based representation of the road structure in an image. Vertices in the graph corresponding to sections of roads can be categorized into linear segments, corners (L), or intersections (T , X , etc.) according to the number and turning

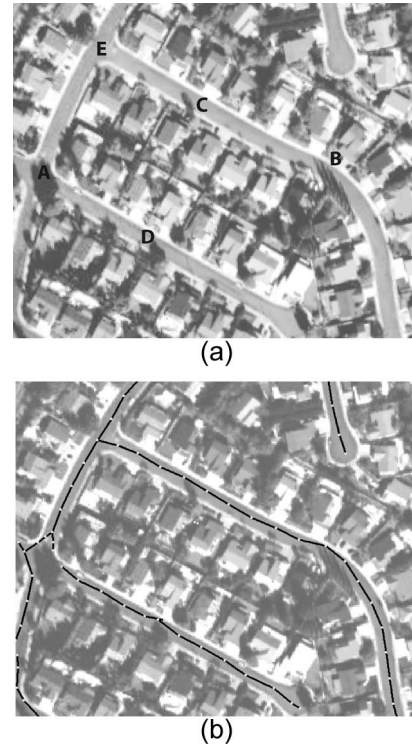


Fig. 1. Road extraction using the proposed method. (a) Original image shows tree shadows on roads labeled A–D and intersections labeled by A and E. (b) Result of road extraction by our proposed method. Black curves are the inscribed lines of roads, and white dots are the vertices of the inscribed lines.

angle of adjacent road segments. Road networks are generally reliable and stable landmarks and are very useful features for solving matching problems. There are four issues related to road network extraction and road intersection identification: 1) the variety of roads with different features (width, intensity, shadows) in one image; 2) multiple types of intersections; 3) variable width along road segments due to shadows or cars (see Fig. 1), and 4) leakage from roads to the surroundings because of the imaging noise or the physical connection between a road and its surroundings.

Our main contributions are as follows.

- 1) We utilize road footprints to detect road directions at a pixel instead of searching the maximum or minimum response to a function of image intensity.
- 2) The proposed road tracker is more adaptive to roads with high curvature and sharp turns. We present a method to detect and classify road intersections.
- 3) We present an automatic method to prune the branches resulting from overextraction and leakage.

Manuscript received February 9, 2007; revised June 21, 2007. This work was supported by the National Geospatial-Intelligence Agency under Grant HM1582-05-1-2004.

J. Hu, A. Razdan, and J. C. Femiani are with the I^3DEA Lab, Division of Computing Studies, Arizona State University's Polytechnic Campus, Mesa, AZ 85212 USA (e-mail: hu.jiuxiang@asu.edu; razdan@asu.edu; john.femiani@asu.edu).

M. Cui and P. Wonka are with PRISM Lab, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: mcui3@asu.edu; peter.wonka@asu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2007.906107

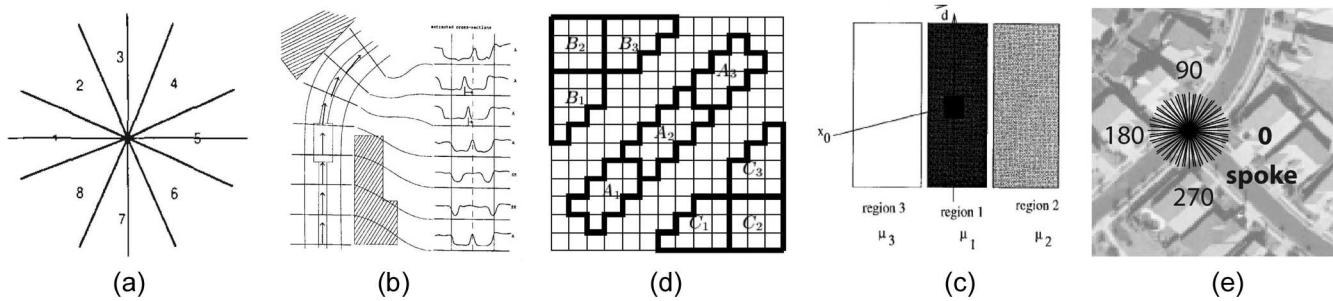


Fig. 2. Techniques for road detection at a pixel. (a) Edge directions for edge detectors from [10]. (b) Road cross-section profile matching for road tracker from [15]. (c) Ratio edge detector from [18]. (d) Road mask filter from [20] includes the central homogeneous road regions A_i and both sides B_i and C_i of the road ($i = 1, 2, 3$). (e) Spoke wheel operator (see Section II-B).

A. Previous Work

Research on extracting roads from aerial and satellite images can be traced back to the 1970s [9]. Many of these methods are based on a bilevel analysis: a local analysis involving the use of local operators to detect road direction at a pixel, and a road network grouping analysis that applies a higher level road model to improve the performance of these methods. We organize previous work according to the use of *road detection* and *road network grouping* criteria to extract road networks.

1) *Road Detection Techniques*: Road detection techniques include edge detectors [10]–[12], morphological operators [13], [14], intensity cross-section profile matching [15]–[17], ratio edge detectors [18], [19], road mask filters [9], [20], [21], and line segment matching [22].

The *edge-based method* is the most common method to detect roads. The goal of edge detection is to mark the points in an image at which the intensity sharply changes. Edge-based methods apply the first and second derivatives to detect the edges in images [10] [see Fig. 2(a)]. Steger [23] made use of a Hessian matrix to find directions normal to a road. However, aerial images usually include such complex scenes that edge detectors extract too many extra edges to recognize where the roads are.

Morphological methods use morphological operators such as grayscale dilation or erosion to identify roads. Chanussot *et al.* [13] presented a directional morphological approach to detect roads in synthetic aperture radar (SAR) images. Based on the authors' road geometrical model, a series of morphological operators were designed to represent linear features with specific widths. The roads are extracted by simple thresholding after applying the morphological operators. Unfortunately, this method yields incomplete detection of the road networks and several spurious detections.

Intensity cross-section profile matching. The intensity profile is a 1-D cross section of the image intensities taken orthogonal to the direction of a road segment. Profile matching compares a reference profile with the road profile at a pixel to predict the road directions. McKeown and Denlinger [15] used profile matching along with their road surface model to follow the direction of roads [see Fig. 2(b)].

Ratio edge detectors were introduced by Bovik [19]. Tupin *et al.* [18] performed local detection of a road based on the fusion of the results from a ratio edge detector and a cross-correlation line detector, taking into account the statistical

properties of speckle in SAR images [see Fig. 2(c)]. However, the authors' speckle filters are not necessarily the most suited operator to improve road detection, although they can reduce the image noise.

Mask matched filters compare a vertex's neighborhood against a set of exemplars to categorize it. Bajcsy and Tavakoli [9] used 52 templates to find the likely road points. Huber and Lang [20] applied two road mask filters [one of them is shown in Fig. 2(c)] to characterize a road by a central homogeneous region adjacent to two homogeneous regions on both sides of the road. Gamba *et al.* [21] also applied 16 road masks to determine road direction at a pixel. However, it is very difficult to enumerate all possible road masks.

The road detectors discussed above were designed to find the direction of a road at a pixel. The road direction obtained by local road detectors usually corresponds to the maximum or minimum of the cost function of the intensity in a neighborhood located at the pixel. These local detectors detect only a single tangent direction of the road and are not able to detect the singularities that occur at the intersections of the roads. In addition, these local operators were designed to detect roads with specific width. The morphological operators [13] and the ratio edge detector [18] are designed to extract roads in low-resolution images, where the widths of roads are usually less than five pixels. Edge detectors [24] and road mask filters [21] are good for extracting the edge of roads in median-resolution images. The filters based on intensity cross-section profile matching are designed to detect highways in high-resolution satellite images, where the width of roads is often more than 15 pixels [16].

2) *Road Network Grouping Techniques*: The performance of the local analyses can be greatly improved by techniques that introduce some global constraints on the line segments detected at road detection level. These techniques minimize a global cost function by applying tracking methods [15]–[17], Markov random field (MRF) techniques [14], [18], [25], [26], or dynamic programming [24].

Road tracking is an iterative line growing process; a spanning tree of the road network is formed starting with a seed point, and local information is used to add new segments to the graph based on the pixel intensities of the image. McKeown and Denlinger [15] presented a road tracking method based on the intensity profile matching of road cross sections to follow the direction of a road [see Fig. 2(b)]. Their profile matching technique compares a reference profile with the road profile at

a pixel predicted to be on the road. The differences between the two profiles are measured by identifying two geometric parameters (shift and width) and two radiometric parameters (brightness and contrast). These parameters are estimated by minimizing the squared sum of the gray value differences between the profiles. Vosselman and Knecht [16] improved the road tracking technique by using the Kalman filter, and Zhou *et al.* [17] applied particle filters to solve the profile matching problem for road tracking. Baumgartner *et al.* [27] manually enhanced road tracking based on profile matching with a graphical user interface that guides an operator through the whole data acquisition process. However, they often fail to go around the sections of the road narrowed by cars or shadows and lose their directions at road intersections or segments with high curvature. Manual seeding also makes it difficult to fully automatically extract road networks.

MRF-based models have been used to identify road networks. *Simulated annealing* was used by Tupin *et al.* [18], [25], [26] to solve an MRF over road segments and all possible segments connected to them. *Dynamic programming* was used by Barzohar and Cooper [24] to find roads by first partitioning an image into windows and using dynamic programming to find the MAP estimation in each window. They then select the windows containing high confidence estimates as seeds. They finally apply dynamic programming again to obtain optimal global estimates of the roads present. Global techniques based on dynamic programming are sensitive to image noise [28].

All of the road trackers and detectors mentioned above make use of the anisotropic nature of roads and detect the direction of a single road tangent. However, few of these methods are able to detect the road intersections. They are mostly limited to extraction of the roads with low or bounded curvature. There are several road intersection detectors [1], [7], [26], [29], [30]. Deschenes and Ziou [29] developed a line junction detector based on curvatures between the directions of line pixels within a given neighborhood; however, their method yielded extra detections of intersection that are not related to roads at all. There are other approaches of intersection identification by using fusion of other types of data such as road vector data [7].

B. Proposed Method

In aerial and satellite images, roads are often modeled as continuous and elongated homogeneous regions with nearly constant width [16], [31]. Geometric shapes of road sections also play a crucial role in road recognition. In this paper, we present a new two-step approach based on road footprint classification to automatically extract road networks and detect intersections from aerial images. Our algorithm consists of three steps.

Automatic road seeding. We analyze the neighborhood of all pixels in the image to decide if the pixel is a valid starting point for a road segment. Details are described in Section II-C.

Road tracking. Starting from all the seeds,¹ we iteratively grow road segments in one, two, or more directions. We iter-

atively analyze the local neighborhood of all active points in the road network to decide whether to stop, to continue in one direction, or to split in two or three directions. This algorithm grows a road tree. See Section II-E for details.

Road tree pruning. The previous steps produce a road network that contains almost all road segments, but suffers from overextraction and leakage. We use a Bayesian decision rule to remove portions of the extracted network that do not appear to be roads. Details are given in Section III.

The remainder of this paper is organized as follows. In Section II, an automatic road tracking method based on road footprint classification is presented. Several concepts and efficient algorithms are discussed. Section III presents an original road pruning method based on Bayes decision rule. In Section IV, the results of applying these algorithms to real aerial images are presented. Finally, Section V provides a summary.

II. ROAD TRACKING BASED ON ROAD FOOTPRINT CLASSIFICATION

A. Roads in Aerial Images

In aerial images, roads are long and thin linear structures with the following geometric and radiometric features.

- *Directional rectangularity.* Each road pixel has a local homogeneous region, which is anisotropic and directionally rectangular. That is, along some directions, the branches of the local homogeneous region, called toes, are approximately rectangular [see Fig. 7(a)].
- *Bounded width.* Roads are thin elongated structures with a bounded width. Width of a road is constant or piecewise constant.
- *Contrast.* Roads are usually contrasted to their surroundings and are either darker or brighter structures with respect to their neighbors.

Generally, the intensity within the road is not constant because cars, shadows, and lane markers can produce rapid changes in intensity. These factors make it difficult to determine the local homogeneous region around a road pixel.

We now describe our road tracking method. The input to this method is one or more valid seeds. Each seed is a pair of pixels that represent a road segment in the image. The output of the algorithm is a set of line segments representing road network in the given image. We begin the discussion of our method by introducing road footprint.

B. Footprint

Many methods have made use of directional angular operators to detect local linear structures on geospatial images. For example, Tupin *et al.* [18] considered eight directions to find the minimum response of a ratio edge detector on both sides of a linear structure. In [32], major roads in a satellite image are approximated by allowing three angles between successive line segments: left turn, no turn, and right turn. Chanussot *et al.* [13] applied 40 directional closing operations to detect local linear structures. Jin and Davis [31] utilized 72 directions to generate the spatial signature of a road pixel. Negri *et al.* [26]

¹A road tracker based on automatic road seeding generates one seed and spans a road tree from the seed completely before searching another one, whereas road trackers on manual road seeding can take several seeds as input that can be grown at the same time (see Sections IV-B and C).

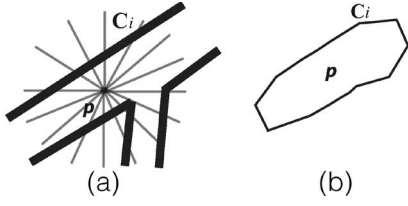


Fig. 3. Concept of a footprint obtained using a spoke wheel operator. (a) Spoke wheel is shown by 16 gray spokes centered at \mathbf{p} . (b) Footprint around pixel \mathbf{p} .

made use of 180 directions to filter linear road networks from low-resolution SAR images. Gibson [33] and Zhang and Couloigner [34] used rectangles instead of lines to detect road pixels. The advantage of directional operators is that they can transfer a 2-D local image intensity distribution into a set of 1-D intensity functions. This makes it easier to find the directions of anisotropic structures in an image by comparing the difference among the directional functions.

In the following sections, we introduce the *spoke* and the *spoke wheel* W [see Fig. 2(e)] used for finding a *footprint* of a pixel \mathbf{p} . A spoke is a line segment with a length of m pixels. A spoke wheel is a sequence of spokes $S_i(\varphi_i, m)$ ($i = 0, \dots, 4n - 1$) with common initial point \mathbf{p} and evenly spaced angles $\varphi_i = \pi i / 2n$. The set of pixels in a spoke wheel W centered at the pixel \mathbf{p} with $4n$ spokes is denoted by $W(\mathbf{p}, n, m)$.

The intersection between a spoke and a road edge provides useful information to determine the local homogeneous region around a pixel. However, we do not have a prior knowledge of where a road edge is. To search for the intersection of a spoke and the edge of a road, we start from \mathbf{p} , move in the direction of the spoke, and observe the absolute intensity differences between \mathbf{p} and the pixels along the spoke. The differences are small when the pixels are near to \mathbf{p} ; however, they may become larger when the pixels are far away from \mathbf{p} . Let S_i be the i th spoke at pixel \mathbf{p} . The *cutting point*, denoted by \mathbf{C}_i , on S_i is the first pixel such that

$$|I(\mathbf{C}_i) - I(\mathbf{p})| \geq \sigma(W(\mathbf{p}, n, m)), \quad 0 \leq i < 4n \quad (1)$$

where $\sigma(W(\mathbf{p}, n, m))$ is the intensity standard deviation on $W(\mathbf{p}, n, m)$. Notice that $\sigma(W(\mathbf{p}_1, n, m)) \neq \sigma(W(\mathbf{p}_2, n, m))$ usually holds if $\mathbf{p}_1 \neq \mathbf{p}_2$. Therefore, the thresholding in (1) is adaptive.

The road associated with each pixel is an anisotropic structure, i.e., the distance $\|\mathbf{C}_i - \mathbf{p}\|$ in some directions are much longer than that in other directions [see Fig. 3(a)]. To find the road directions, we connect the cutting points on all spokes around a pixel \mathbf{p} in a counterclockwise direction, which results in a closed polygon. This represents the *footprint* of the pixel \mathbf{p} , denoted by $F(\mathbf{p})$. Fig. 3(b) shows a footprint of the pixel \mathbf{p} in Fig. 3(a).

Fig. 4(b) shows 20 footprints of the pixels labeled in Fig. 4(a). The spoke wheel used has 64 spokes, and each spoke is 18 pixels long. The first 13 footprints have clear directional rectangular toes, which correspond to the directions of roads at those pixels in Fig. 4(a). For example, footprint 0 has three toes, which correspond to a T-shaped local homogeneous region around the pixel shown in Fig. 4(a).

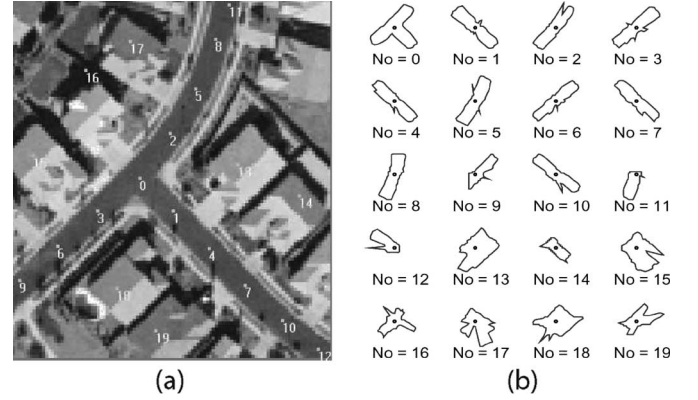


Fig. 4. Illustration of footprints. (a) Zoomed original image with 20 labeled pixels. (b) Twenty footprints corresponding to the labeled pixels in (a) are shown. The footprints are drawn at the same scale and in their original orientations. The first 13-pixel footprints are on the road, no. 16 is in a shadow, and the others are on roofs of houses.

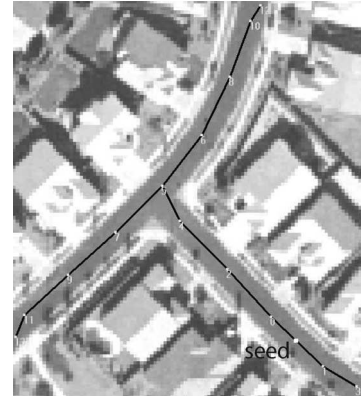


Fig. 5. Illustration of automatic road seeding starting with a seed (a pair of vertices 0, 1), which comes from the footprint of the pixel shown as a white square.

C. Automatic Road Seeding Based on the Rectangularity of Road Footprints

Most road trackers [15], [27], [32] require manual seeding. Our proposed method retains this option but can also automatically detect seeds. Road networks usually have a segment with low curvature; therefore, at least one footprint is approximately rectangular. Rosin [35] presents four methods to measure the rectangularity of polygons. We applied the minimal oriented bounding box (MOBB) method to decide if a footprint is rectangular. We threshold the ratio of its area to that of its MOBB. A footprint F is nearly rectangular if

$$\frac{\text{AREA}(F)}{\text{AREA}(\text{MOBB}(F))} > 85\% \quad (2)$$

and

$$\frac{\text{length of longer edge of MOBB}(F)}{\text{length of shorter edge of MOBB}(F)} > 2 \quad (3)$$

hold. The thresholds of 85% and 2 were selected by experiment. Equations (2) and (3) ensure that a footprint approximates a narrow rectangle. We initialize the proposed road tracker by the

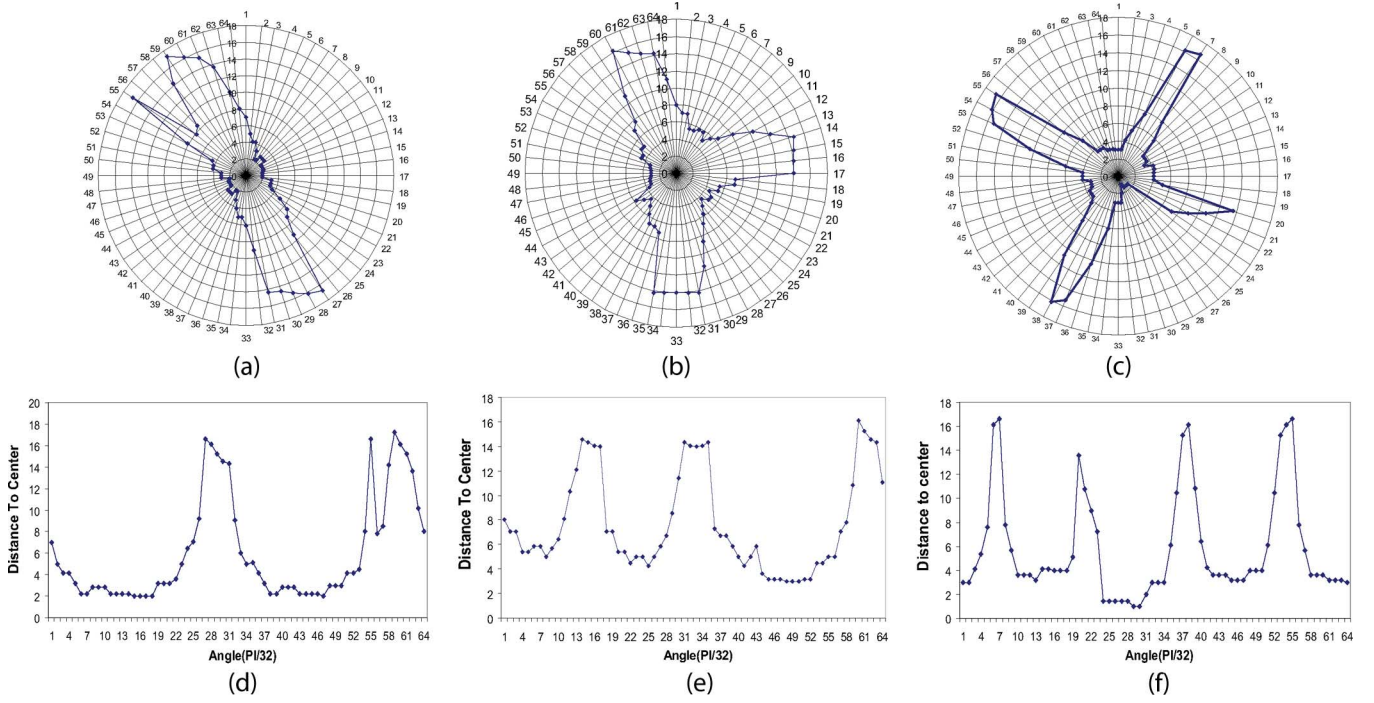


Fig. 6. Top row shows three basic footprints from a real aerial image, and bottom row shows corresponding distance functions. (a) *L* footprint with two branches. (b) *T*-shaped *F* with three branches. (c) *X*-shaped *F* with four branches. (d)–(f) Distance function $\delta_T(\theta)$.

line segment connecting the middle points of the short edges of $\text{MOBB}(F)$. Fig. 5 shows the result of road tracking.

D. Toe-Finding Algorithm for Footprint Classification

A toe-finding algorithm is designed to find the number of dominant peaks that represent road directions. For example, Fig. 6(b) and (e) shows a footprint of a pixel located at a *T*-shaped intersection and the corresponding distance function of the footprint. It is clear that the distance function has three dominating peaks. However, there are small peaks on or near dominating peaks of some footprints [see Fig. 6(d)]. This makes it difficult to precisely locate the dominating peaks. We present a toe-finding algorithm based on the distance function of a footprint, which is similar to a peak-finding algorithm for histogram analysis in [36]. Our toe-finding approach is carried out in two steps. First, we recognize the dominant peaks of the distance function. Second, we find the valleys between sequential peaks. Suppose a distance function $\delta(\theta)$ is represented by $\delta(i)$, where i is an integer ($0 \leq i < 4n$).

Toe-finding algorithm:

- Shift the distance function $\delta(i)$ so that $\delta(0)$ is less than $\text{avg}\{\delta(i)\}$. The goal of this step is to avoid splitting a dominant peak.
- Find all peaks: find the local maxima that exceed the average of the distance function

$$\begin{aligned} \delta(i) &> \delta((i-1+4n) \bmod (4n)) \\ \delta(i) &> \delta((i+1) \bmod (4n)). \end{aligned} \quad (4)$$

- Remove small peaks: if a peak is too small compared to the highest peak, then it is removed. Suppose the height of

the highest peak is δ_{\max} . For any peak j , if $\delta(j)/\delta_{\max} < \epsilon_1$, then peak j is removed. In addition, we remove peaks whose heights are less than the average height.

- Merge the peaks that are close. For two peaks $\delta(i_1)$ and $\delta(i_2)$, $i_2 > i_1$, if $|i_2 - i_1| < \epsilon_2$ or $|4n - i_2 + i_1| < \epsilon_2$, then merge the two peaks so that $\delta(j) = \max\{\delta(i_1), \delta(i_2)\}$ ($j = i_1$ or i_2). Thus, the peak with the higher value is chosen.
- Remove a peak if the valley between two peaks is not significant. This is examined by calculating the average distance along the horizontal axis value between the two peaks. Suppose that δ_{avg} is the average among the points between peaks i_1 and i_2 . Then

$$\delta_{\text{avg}} = \frac{\sum_{j=i_1}^{i_2} \delta(j)}{i_2 - i_1 + 1}. \quad (5)$$

Then, if $2\delta_{\text{avg}}/(\delta(i_1) + \delta(i_2)) > \epsilon_3$, we say that the valley is not deep enough to separate the two peaks. We remove the peak with the smaller value from the candidates.

In the algorithm, constants ϵ_1 , ϵ_2 , and ϵ_3 are selected based on experiment. We select $\epsilon_1 = 0.25$ and $\epsilon_3 = 0.8$. We set $\epsilon_2 = (\text{number of spokes})/8$, i.e., if the angle of two peaks is less than $\pi/4$, then the two peaks are merged. For example, a small peak shown in Fig. 6(d) was merged with another peak.

We now classify road footprints into five categories based on the number of their toes and the turning angle from their parent edge as follows:

- *normal*: if it has two toes and the turning angle is less than $\pi/4$;
- *L shaped*: if it has two toes and the turning angle is greater than $\pi/4$;

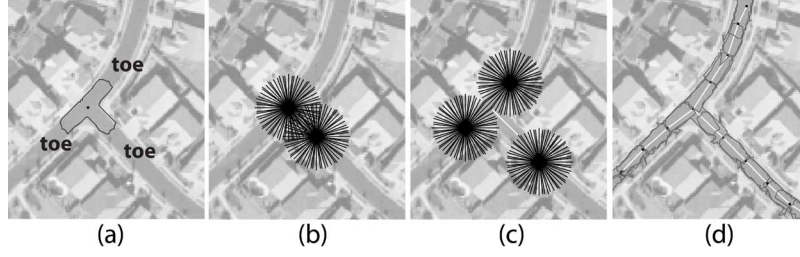


Fig. 7. Portion of road appears in an aerial image, and the main steps of our road tracker are shown. (a) Local homogeneous region (enclosed by the green polygon) around pixel \mathbf{p} with three directional rectangular toes. (b) Two seeds with an edge (e_0) as initial condition \mathbf{E}_0 of our road tracker. (c) Our road tracker with the inscribed lines \mathbf{E}_3 . (d) Extracted results of road inscribed lines \mathbf{E}_{12} (red line segments) with their footprints (green overlapped polygons).

- *T shaped*: if it has three toes;
- *X shaped*: if it has four toes;
- *other*: if it has more than four toes.

E. Road Network Extraction by Tracking Road Footprints

Our road extraction method is an iterative line segment growing process based on road footprints, called road tracking [see Fig. 7(b)–(d)]. Let \mathbf{P}_i be the set of pixels identified as road candidates after iteration i . Let $\mathbf{T}_i = (\mathbf{V}_i, \mathbf{E}_i)$ be a tree representing the road network at iteration i with edges \mathbf{E}_i and vertices \mathbf{V}_i . Every vertex² $\mathbf{v}_j \in \mathbf{V}_i$ has two types: alive or dead. If \mathbf{v}_j is alive, we determine its footprint and then search for all possible toes of the footprint by toe-finding algorithm. Because \mathbf{v}_j has exactly one edge in \mathbf{E}_i connecting it to its parent in \mathbf{V}_i , it must have at least one toe in the direction of the edge. Let t ($t \geq 1$) be the number of toes of $F(\mathbf{v}_j)$. Each toe of the footprint determines a direction and size for road tracking and generates a new vertex $\hat{\mathbf{v}}_k$ (located at the center of the toe) and a new edge $\hat{e}_k = (\mathbf{v}_j, \hat{\mathbf{v}}_k)$ ($1 \leq k \leq t - 1$). We always update $\mathbf{P}_{i+1} = \mathbf{P}_i \cup \{\text{pixels enclosed by } F(\mathbf{v}_j)\}$. When $t > 1$, we update $\mathbf{V}_{i+1} = \mathbf{V}_i \cup \{\hat{\mathbf{v}}_k\}$ and $\mathbf{E}_{i+1} = \mathbf{E}_i \cup \{\hat{e}_k\}$, and the road tree \mathbf{T}_i is expanded to $\mathbf{T}_{i+1} = (\mathbf{V}_{i+1}, \mathbf{E}_{i+1})$. For each new vertex $\hat{\mathbf{v}}_k$ ($k = 1, \dots, t - 1$), if $\hat{\mathbf{v}}_k \notin \mathbf{P}_i$, then $\hat{\mathbf{v}}_k$ is alive; otherwise, $\hat{\mathbf{v}}_k$ is dead, and its position is moved to the centroid of its footprint. Regardless of t , we set the state of the vertex \mathbf{v}_j to dead, and we will not attempt to grow from it again. The road tracking does not stop until all the vertices in a road tree are dead.

The output of our road tracker is the *inscribed lines* that approximate the structure of road networks instead of the “centerline” of road networks. This is because shadows or cars on road may deviate our road tracker from the centerlines of roads [see the inscribed line at the place labeled B in Fig. 1(b)]. Moving the inscribed lines to the centerlines is considered a preprocess that we will address in future work.

III. ROAD TREE PRUNING

Due to image noise or natural connections of roads to other structures such as parking areas, road tracking algorithm may leak into the surrounding areas. We need to prune the extracted segments that do not belong to the road. On the other hand, due

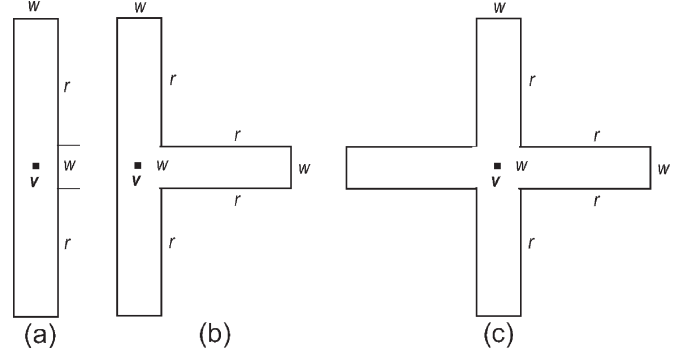


Fig. 8. Directionally local rectilinear property of a vertex \mathbf{v} on road. (a) Ideal rectangular footprint. (b) Ideal T-shaped footprint. (c) Ideal X-shaped footprint.

to shadows of trees or buildings, cars, and pavement, the detected line segments from different seeds may be disconnected. It is necessary to connect them to form the whole road network. An abundance of literature covers this kind of global connecting process with many different approaches [14], [18], [25], [26], [37]–[39].

In the following, we introduce a road tree pruning method based on the width of a road to deal with the leakage problem. We make use of a Bayes decision rule to determine whether a vertex is on the road or not based on the estimated width value of road at the vertex. The input is a road tree extracted from previous section, possibly overextracted. The output is a road network without leakage and other noise artifacts.

A. Area-to-Perimeter Ratio of a Footprint

In general, the width of a road is almost constant, or piecewise constant. However, it is difficult to estimate the width of a road for two reasons. One is that some of the extracted vertices are not on road. Another reason is that vertices at *T* intersections have multiple directions at the road cross section. Instead of directly calculating the width of road at a vertex \mathbf{v} , we consider an area-to-perimeter ratio of its footprint, called *A/P* ratio and is denoted by $d(\mathbf{v})$. Fig. 8(a) shows an ideal two-directional rectangular footprints of vertex \mathbf{v} . The lengths of the shorter edge and its longer edge are denoted by w and $w + 2r$, respectively, where w is determined by the width of road, and r is dependent on the length of the spoke. In this case, the *A/P* ratio is $d(\mathbf{v}) = (w/2)(1 - (w/(2r + 2w)))$. When the spokes are selected so that their lengths are much longer than the width of road, then $r \gg w$; therefore, $d(\mathbf{v}) \approx w/2$.

²In this paper, after a pixel on an image is extracted, it is called a *vertex*. Pixel and vertex are all denoted by bold lowercase letters.

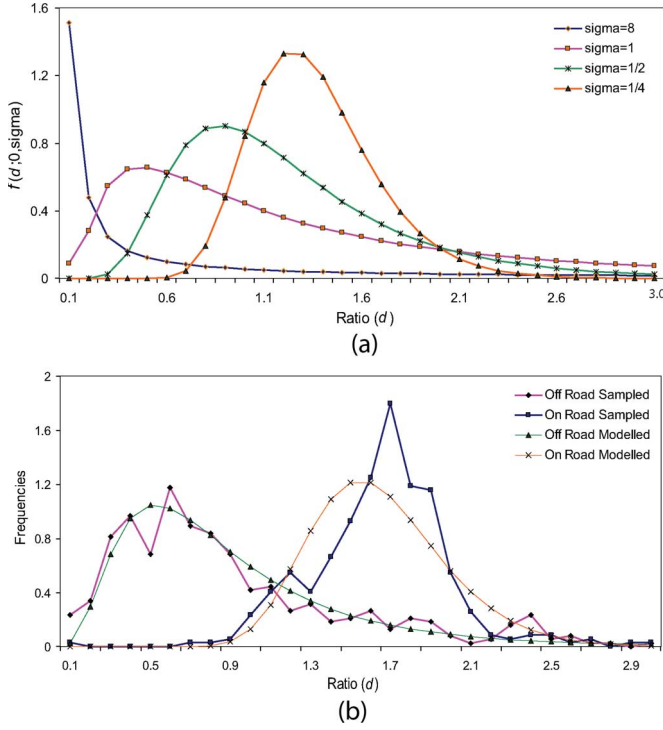


Fig. 9. (a) Probability density function of lognormal distribution with $\mu = 0$ and $\sigma = 8, 1, 1/2$, and $1/4$, respectively. (b) Conditional frequencies of A/P ratio $d(\mathbf{v})$ with $\mathbf{v} \in \mathbf{V}_r$ (squares) and $\mathbf{v} \in \mathbf{V}_n$ (diamonds), and their lognormal models are shown with triangles and \times .

Similarly, for the T -shaped footprint [see Fig. 8(b)], the A/P ratio $d(\mathbf{v}) = (w/2)(1 - (w/(3r + 2w))) \approx w/2$; for the X -shaped footprint [see Fig. 8(c)], the A/P ratio $d(\mathbf{v}) = (w/2)(1 - (w/(4r + 2w))) \approx w/2$ as well. We conclude that the A/P ratio of a footprint is close to half the road width and is independent of the number of toes in a footprint. In the next section, we prune the superfluous paths using A/P ratio $d(\mathbf{v})$ over the set of extracted vertices.

B. Road Tree Pruning Based on the Bayesian Decision Rule

Let \mathbf{V} denote the final set of detected vertices and \mathbf{E} the final set of detected edges. For each $\mathbf{v} \in \mathbf{V}$, we can define the A/P ratio. The vertices and edges are organized as a *road tree* $\mathbf{T} = (\mathbf{V}, \mathbf{E}; d(\mathbf{v}))$ [see Section II-E and Fig. 11(c)]. Among these vertices, some belong to the roads, denoted by \mathbf{V}_r , and others are falsely detected, denoted by \mathbf{V}_n . \mathbf{V}_r and \mathbf{V}_n partition \mathbf{V} . Let $d_{\max} = \max_{\mathbf{v} \in \mathbf{V}} \{d(\mathbf{v})\}$. We normalize the A/P ratio by $3.0 * d(\mathbf{v})/d_{\max}$ ³; thus, the random variable $d(\mathbf{v}) \in [0, 3]$. Road detection consists of identifying vertices that belong to a road, i.e., labeling the road tree. A binary variable ω is, therefore, associated with vertex \mathbf{v} ; $\omega = 1$ if \mathbf{v} belongs to a road ($\mathbf{v} \in \mathbf{V}_r$), and $\omega = 0$ if \mathbf{v} is not on the road ($\mathbf{v} \in \mathbf{V}_n$). The random label ω takes its value from the set $\{0, 1\}$.

For a given vertex in \mathbf{V} , we suppose that the A/P ratio of its footprint is computed, and its value is d . How does this measurement influence our decision? The decision is based on

³We suppose that $d(v)$ follows a lognormal distribution later, i.e., $\log d(v)$ follows a Gaussian distribution. Because $\log e = 1$ and 3 is the nearest integral to e , we normalize d into the interval $[0, 3]$.

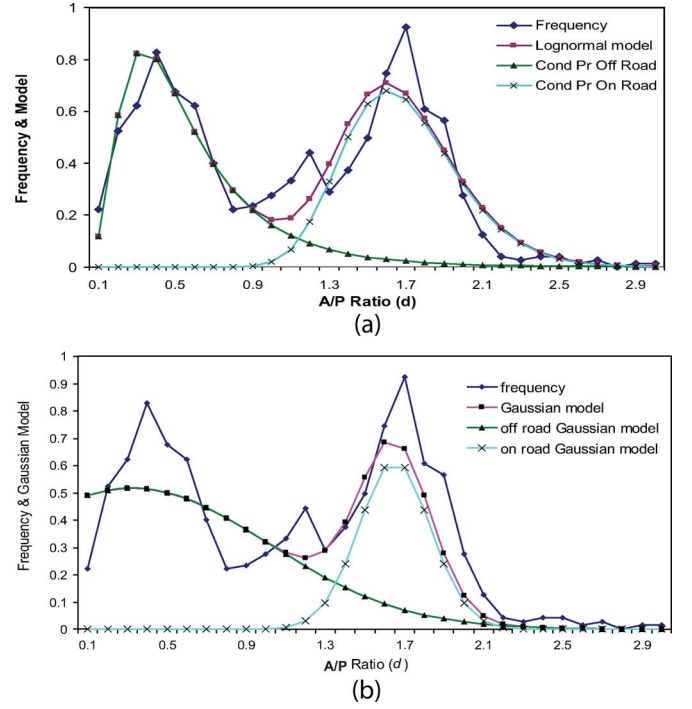


Fig. 10. Frequencies of the A/P ratio $d(\mathbf{v})$ on an aerial image [see Fig. 11(a)] and their lognormal model. (a) Evidence and its lognormal model with $\lambda = 0.508$, $\mu_0 = -0.731$, $\sigma_0 = 0.606$, $\mu_1 = 0.503$, and $\sigma_1 = 0.177$. (b) Evidence and its Gaussian model with parameters $\mu_0 = 0.327$, $\sigma_0 = 0.971$, $\mu_1 = 1.650$, and $\sigma_1 = 0.257$.

the comparison of the conditional probability distribution of ω given the observation d : $p(\omega|d)$ (also called the posterior probability distribution). Through a *Bayes decision rule* for minimizing the probability of error [40], we decide that $\omega = 1$ if $p(1|d) > p(0|d)$; otherwise, $\omega = 0$.

Using Bayes formula, we have the following posterior probability distribution:

$$p(\omega|d) = \frac{p(d|\omega)P(\omega)}{p(d)}. \quad (6)$$

For two categories

$$p(d) = \sum_{\omega \in \{0,1\}} p(d|\omega)P(\omega). \quad (7)$$

Instead of directly computing the posterior probability distribution, $P(\omega)$ and $p(d|\omega)$ have to be estimated. The conditional probability distribution of the observation field $p(d|\omega)$ stems from a supervised learning step on known areas, and the prior probability distribution $P(\omega)$ relies on the frequency function of d on the road tree \mathbf{T} , which approximates the probability distribution $p(d)$ in (7).

C. Conditional, Prior, and Posterior Probability Distributions

There are different ways to model the conditional probability for road network grouping based on observations of road candidates [18], [25], [41]. Based on observation of the conditional frequencies of the A/P ratio from different images and different regions, we assume the following: for each ω , the conditional probability density distribution $p(d|\omega)$ follows

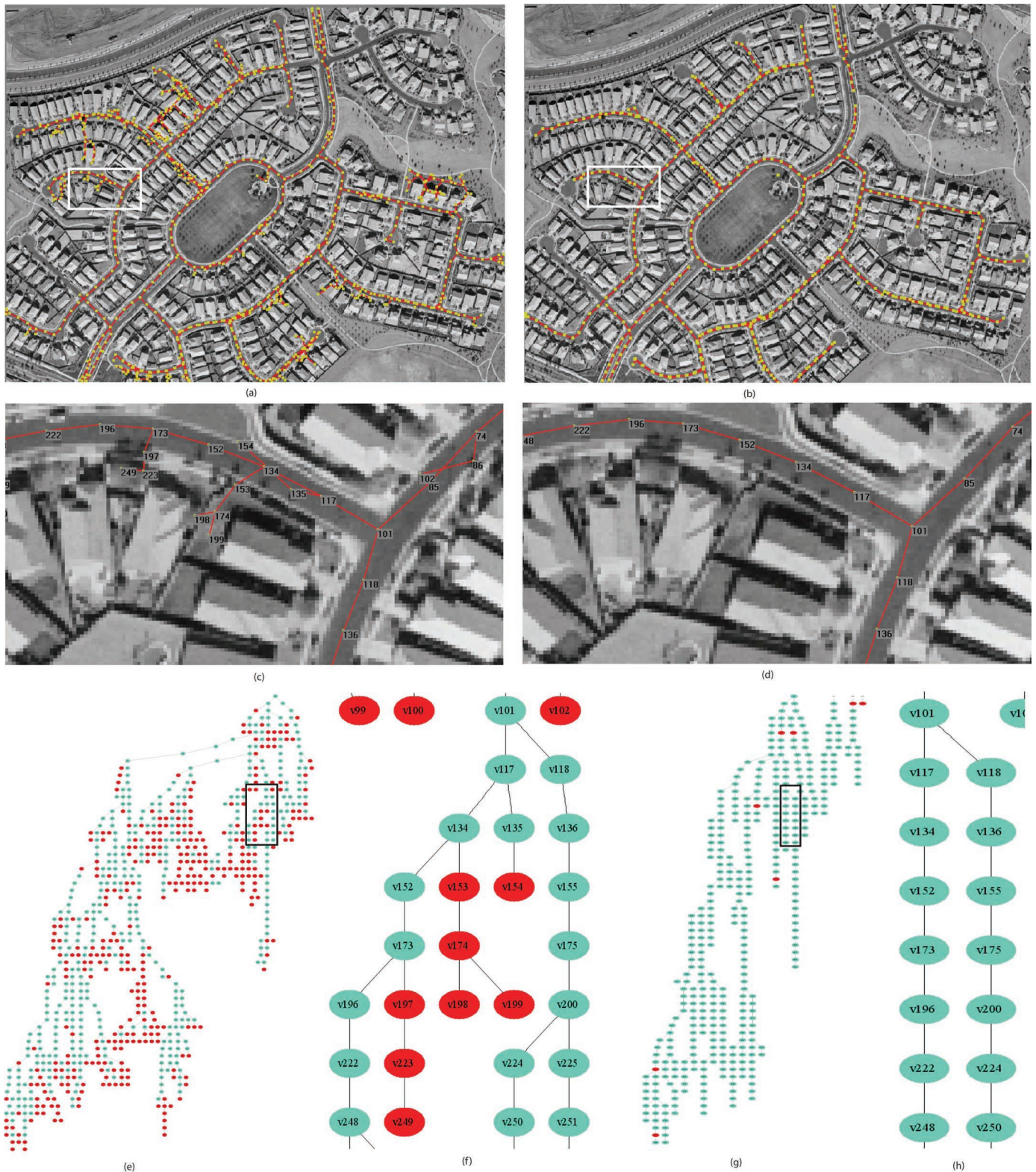


Fig. 11. Road tree extracted from an aerial image and its pruning. (a) Result of road tracking started from a given seed, which includes some branches located in the surroundings. (b) Road inscribed line after road tree pruning based on the lognormal model. (c) Region zoomed in the white window shown in (a). Due to shadows of trees, the road tracker heads into the surrounding from v_{134} to v_{153} and keeps moving around the region nearby a house where the image intensity is close to that on road. (d) Region zoomed in the white window shown in (b). All vertices off road were pruned. (e) Road tree of the extracted roads shown in (a). The color of vertex is determined by the posterior probability $p(\omega|d)$: green if $p(1|d) > p(0|d)$, red otherwise. (f) Zoomed part of road tree shown in the darker window in (f). It is clear that vertices v_{153} , 154, 174, 198, and 199 are colored red and are not on the road [see (c)]. (g) Road tree after pruning corresponds to road inscribed line shown in (b). The color code of vertices is the same as that in (f). (h) Zoomed part of the graph shown in the darker window in (h).



Fig. 12. Road inscribed line after road tree pruning based on the Gaussian model. It is obvious that some vertices on the road were also pruned.

the lognormal distribution with parameters μ and σ . That is, the conditional probability $p(d|\omega)$ can be modeled as a lognormal distribution [42]

$$p(d|\omega) \equiv f(d; \mu, \sigma) = \frac{a}{d} e^{-\frac{(\ln d - \mu)^2}{2\sigma^2}}, \quad \text{for } d > 0 \quad (8)$$

where $a = 1/(\sigma\sqrt{2\pi})$, and μ and σ are the mean and standard deviation, respectively, of $\ln d$ and are calculated by

$$\mu = \ln(E(d)) - \frac{1}{2} \ln\left(1 + \frac{\tau(d)}{E(d)^2}\right) \quad \sigma^2 = \ln\left(1 + \frac{\tau(d)}{E(d)^2}\right) \quad (9)$$

for given expected value $E(d)$ and variance $\tau(d)$ of d over $v \in \mathbf{V}$. Fig. 9(a) plots the lognormal probability density functions (pdfs) with $\mu = 0$ and a different σ . Using (9), we can calculate the parameters of the lognormal model, which is followed by the A/P ratios $d(\mathbf{v})$ over $\mathbf{v} \in \mathbf{V}_r$ and over $\mathbf{v} \in \mathbf{V}_n$, respectively.

We also make use of an example to illustrate our observation on the conditional probability $p(d|\omega)$. Fig. 11(a) shows extracted roads from an aerial image, where \mathbf{V} is the vertex set (yellow dots) of a road tree \mathbf{T} that started from a manual seed, and \mathbf{V} is manually classified into two classes \mathbf{V}_r and \mathbf{V}_n . We use histograms to estimate the conditional density $p(d|\omega)$ and then calculate the mean and the standard deviation of the lognormal distributions using (9). In Fig. 9(b), the curves labeled with diamonds and triangles show the conditional frequency and the corresponding lognormal distribution ($\mu = -0.282$, $\sigma = 0.604$) of the A/P ratio $d(\mathbf{v})$ over $\mathbf{v} \in \mathbf{V}_n$, respectively. Kolmogorov–Smirnov test with p value = 0.76 shows that there is no difference between the triangle curve and the diamond curve. The curves with squares and with x's show the conditional frequency and the corresponding lognormal distribution ($\mu = 0.479$ and $\sigma = 0.205$) of the A/P ratio $d(\mathbf{v})$ over $\mathbf{v} \in \mathbf{V}_r$, respectively. Notice that the frequencies are normalized by using $\sum_{i=1}^N f(d_i) \Delta d_i = 1$ (N is the number of bins of the histogram). Fig. 9(b) shows that for a vertex $\mathbf{v} \in \mathbf{V}$, if $1.2 < d(\mathbf{v}) < 2.2$, the probability of $\mathbf{v} \in \mathbf{V}_r$ is larger than that

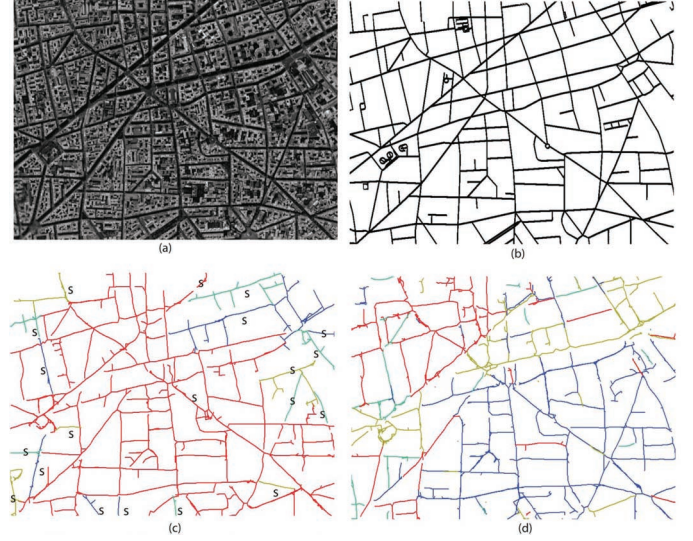


Fig. 13. Road network extraction results from an urban region. Note that the width of the road changes a lot. (a) Original image. (b) Reference roads. (c) Inscribed lines of roads extraction with manual seeding and road tree pruning (the position of each seed is labeled by S). (d) Inscribed lines of roads extraction with automatic seeding and road tree pruning. Notice that the different colored road trees are generated from different seeds.

of $\mathbf{v} \in \mathbf{V}_n$. For other values of d , \mathbf{v} has a higher probability to be off the road. Results show that the conditional frequencies follow the lognormal distribution very well.

Unfortunately, we do not have information on the two categories \mathbf{V}_r and \mathbf{V}_n for a given road tree; therefore, we are not able to use (9) to estimate the parameters of conditional probability $p(d|\omega)$. We also do not have information on the prior probabilities $P(\omega)$. In [14] and [18], a set of empirical parameters are used to estimate the conditional probability and the prior probability to detect roads. In this paper, we make use of the evidence $p(d)$ to estimate the prior probabilities $P(\omega)$ and the parameters of the lognormal pdf's followed by the conditional probabilities $p(d|\omega)$. By (7), we have

$$p(d) = p(d|\omega = 0)P(\omega = 0) + p(d|\omega = 1)P(\omega = 1) \\ \equiv \lambda f(d; \mu_0, \sigma_0) + (1 - \lambda)f(d; \mu_1, \sigma_1) \quad (10)$$

where $p(d|\omega = l) = f(d|\mu_l, \sigma_l)$ ($l = 0, 1$), $\lambda = P(\omega = 0)$, and $P(\omega = 1) = 1 - \lambda$ (since $\sum_{\omega \in \{0,1\}} p(\omega) = 1$). On the other hand, we know the frequency of $d(\mathbf{v})$ over a whole road tree, denoted by $\tilde{p}(d)$. We estimate the parameters in (10) by minimizing

$$\min_{\lambda, \mu_0, \sigma_0, \mu_1, \sigma_1} \sum_{i=1}^N [p(d_i; \lambda, \mu_0, \sigma_0, \mu_1, \sigma_1) - \tilde{p}(d_i)]^2 \quad (11)$$

where N is the number of bins in the frequency $\tilde{p}(d)$. This cost function can be minimized using the Levenberg–Marquardt algorithm over five parameters (λ , μ_1 , σ_1 , μ_2 , and σ_2).

Fig. 10(a) shows the evidence and its lognormal model with parameters $\lambda = 0.508$, $\mu_0 = -0.731$, $\sigma_0 = 0.606$, $\mu_1 = 0.507$, and $\sigma_1 = 0.177$, which are calculated by minimizing (11) by means of the Levenberg–Marquardt algorithm with an initial guess (0.4, 0.01, 0.53, 0.40, 0.20) in (10). The frequency of A/P ratio $d(\mathbf{v})$ over $\mathbf{v} \in \mathbf{V}$ is shown by the curve with diamonds (blue), and its lognormal model is given in squares.

Considering that posterior probability densities $p(d|\omega)$ in (6) have the same denominator $p(d)$, labeling can be directly based on comparing the value of $\lambda f(d; \mu_0, \sigma_0)$ with $(1 - \lambda)f(d; \mu_1, \sigma_1)$ in (10). Fig. 11(e) shows a road tree, which was extracted from an aerial image in Fig. 11(a), with different colored vertices: green if $\lambda f(d; \mu_0, \sigma_0) < (1 - \lambda)f(d; \mu_1, \sigma_1)$ and red otherwise.

To deal with the effect of shadows of building or cars on the road, we also consider three additional rules during pruning.

- 1) For branched vertex, $p(w|d(\mathbf{v})) = \max\{p(w|d(\bar{\mathbf{v}}_i) : \bar{\mathbf{v}}_i \text{ are children of } \mathbf{v})\}$.
- 2) If both the parent and one of the children of vertex \mathbf{v} are on the road, then \mathbf{v} is assigned on the road.
- 3) Short branches are pruned. In this paper, if the length of a branch is shorter than two, and it has at least one sibling whose length is longer than five, we then call it a short branch.

Fig. 11(b) and (g) shows the road detection and the corresponding new road tree after pruning. It is easy to see that some green vertices in Fig. 11(f) were pruned; for example, vertex 135 was pruned by rule 3, and, on the another hand, some red vertices remained due to rules 1 and 2. This is why there are still red vertices in Fig. 11(g).

To compare the lognormal model with the Gaussian model, we replace the lognormal model with the Gaussian model and keep the other parameters the same with the analysis above. Fig. 10(b) shows an evidence and its Gaussian model with parameters $\mu_0 = 0.327$, $\sigma_0 = 0.971$, $\mu_1 = 1.650$, and $\sigma_1 = 0.257$, which are also calculated by the Levenberg–Marquardt method with initial guess (0.4, 0.01, 0.53, 0.40, 0.20). The frequency of the A/P ratio $d(\mathbf{v})$ over $\mathbf{v} \in \mathbf{V}$ is shown by the curve with diamonds, and its Gaussian model is given in squares. Fig. 12 shows the results of pruning the extracted roads shown in Fig. 11(a). Compared with the pruning results obtained by the lognormal model shown in Fig. 11(b), the Gaussian model deletes more vertices that should be on the road. The Gaussian model allows a negative value of the A/P ratio; hence, the former part in (10), which corresponds to the distribution of the A/P ratio over \mathbf{V}_n , becomes flat and dominates [see Fig. 10(b)]. Therefore, the Gaussian model can result in the overpruning of a road tree.

IV. EXPERIMENTAL RESULTS

The first example (see Fig. 11) shows in detail the proposed approach. This section demonstrates the performance of our method in four aerial images. Two of them demonstrate the results of the proposed road extraction method using manual seeding, and the others show the performance of our method based on automatic seeding. In this section, we will also present some more general results and quantitative evaluations over a set of assorted images. In these four examples, we use the following parameters: the length of each spoke $m = 16$ pixels and the number of spokes $n = 16$. The hardware we used is a Dell Precision workstation PWS670, with Intel Xeon 2.80-GHz CPU and 8-GB RAM.

A. Evaluation Indexes

The extracted roads were compared with reference roads (as ground truth), which are generated from the road vector data from the GIS. We use three indexes in [43] to evaluate the quality of road extraction as follows:

$$\text{completeness} = \frac{L_m}{L_r} \quad (12)$$

$$\text{correctness} = \frac{L_m}{L_e} \quad (13)$$

$$\text{quality} = \frac{L_m}{L_{ur} + L_e} \quad (14)$$

where L_r is the total length of reference roads \mathbf{R} , L_e is the total length of extracted roads \mathbf{V} , L_{me} is the total length of the extracted roads that match with the reference roads, L_{mr} is the total length of the reference roads that match with the extracted roads, $L_m = \min(L_{me}, L_{mr})$, and L_{ur} is the total length of the reference roads that are unmatched with the extracted roads. Here, if the distance between a vertex \mathbf{v} on the inscribed lines of extracted roads and the inscribed lines of the reference road is less than a given tolerance, then the vertex \mathbf{v} matches the reference road, and vice versa. If two ends of an edge on the inscribed lines match the reference road, then the edge matches the reference road, and so do the edges on the reference road. In this paper, the tolerance is the width of the road.

B. Road Extraction Results With Manual Seeds

We first test our approach based on manual seeding by using two additional images. In both examples, a user manually seeds on roads.

Fig. 13(a) shows an aerial image including a high-density urban road network, which significantly changes in width and in radiometry and has different kinds of road intersections. The approach of road network extraction with manual seeding was performed. The extracted results were compared with the reference road generated from road vector data in a GIS at the same area, which was registered by hand. Our road tracker started from 20 seeds and detected 1577 vertices in total and pruned 198 vertices of them.

Fig. 14 shows the results of the road tracker on the extraction of highway overpass with complex road junctions. There are nine initial road seeds manually placed. Because the radiometry of the parking lots is almost the same as that of the highway, the tracker heads into these areas [see Fig. 14(b)]. After road tree pruning process, the branches at the parking lots were deleted [see Fig. 14(c)].

Table II shows that the completeness of the road tracker based on manual seeding is around 90%, correctness is above 85%, and quality is from 82% to 85%.

C. Road Extraction Results With Automatic Seeding

Figs. 15 and 16 show two examples to demonstrate the proposed automatic seeding method. Users just need to determine whether the road is darker or brighter than its surrounding. Our method scans the image, executes road tracking if a seed is

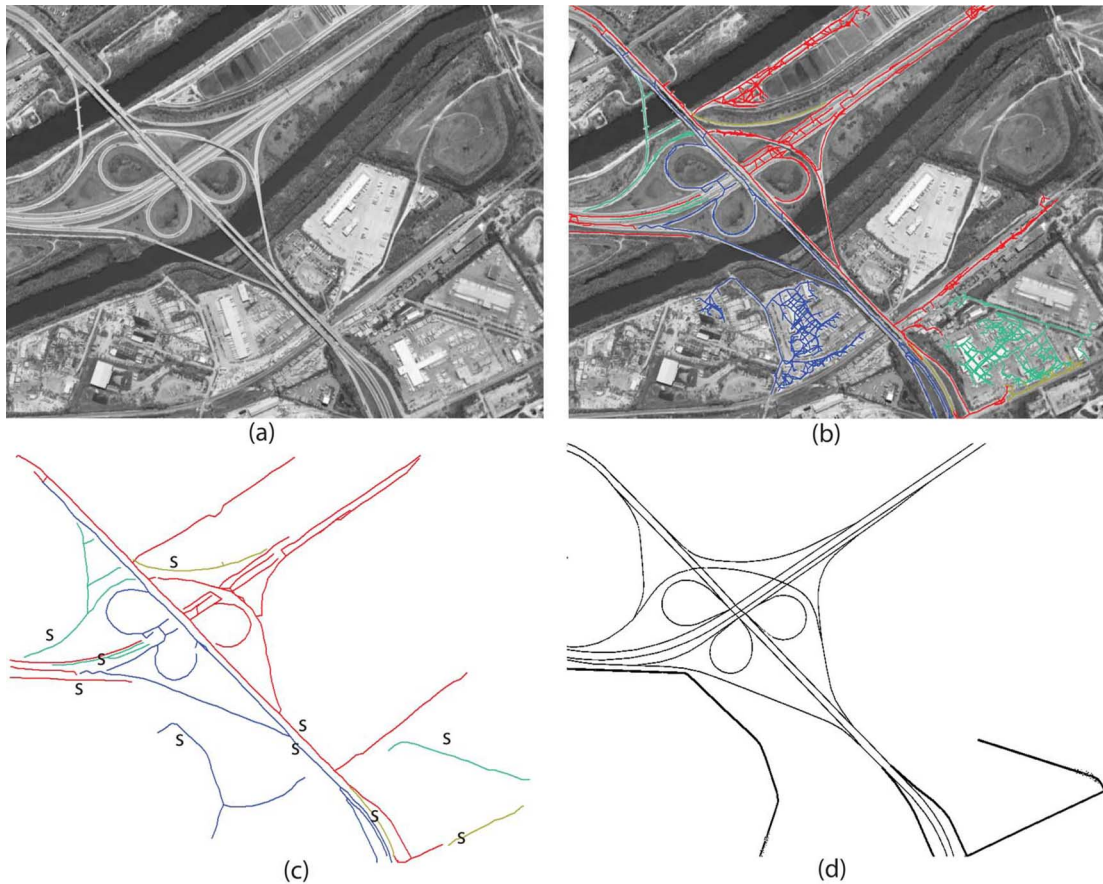


Fig. 14. Road network extraction results from a highway region. (a) Original image. (b) Inscribed line of road extraction before the road tree was pruned. (c) Inscribed line of road extraction after the road tree was pruned. Notice that the different colored road trees are generated from different seeds (the position of each seed is labeled by S). (d) Reference roads.

found, and labels the extracted region. This process does not stop until all pixels are scanned. After road tracking is done, road tree pruning trims the branches that are not on the road.

The roads in Fig. 15(a) have a lot of splinters into the surroundings, and the road network is broken into small pieces. In addition, our tracker extracts several paths in the region labeled by A (see Fig. 15), which do not appear in the reference road networks. The overextraction reduces the correctness and quality measurements in Table II. As can be seen in Fig. 16(a), the width of the road is constant; however, the intensity of the road is not homogeneous, and roads are broken by light short regions, which connect to the surrounding regions. There are a number of superfluous linear structures that are generated by the shadows of fences such as the regions labeled A in Fig. 16(a) or false rectilinear line segments between houses such as the regions labeled by B in Fig. 16(a). Before pruning [see Fig. 16(c)], there are a lot of superfluous line segments detected. Fig. 16(c) shows 30 *L*-shaped corners, 36 *T*-shaped intersections, and 2 *X*-shaped intersections that were detected. Compared with the original image, five *T*-shaped intersections are not detected, and one *X*-shaped intersection is not detected.

Tables I and II show the performances and the quality of our road tracker. Whereas Table I shows that automatic seeding costs much more computation time and creates more seeds and more false detection than manual seeding, Table II shows that

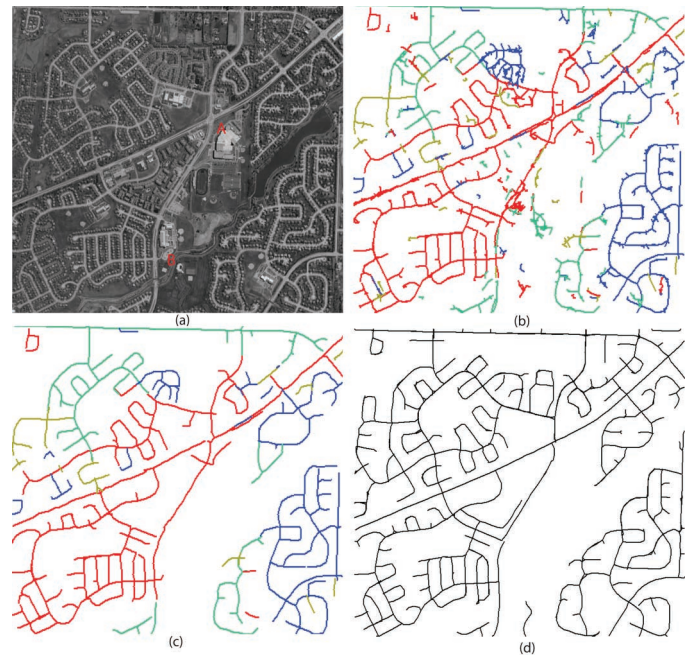


Fig. 15. Road network extraction results from a highway region. (a) Original image. (b) Inscribed line of road extraction before the road tree was pruned. (c) Inscribed line of road extraction after the road tree was pruned. (d) Reference roads.



Fig. 16. Road network detection based on automatic seeding. (a) Original image. (b) Inscribed line of road extraction before the road tree was pruned. (c) Inscribed line of road extraction after the road tree was pruned. There are 30 *L*-shaped corners, 36 *T*-shaped intersections, and 2 *X*-shaped intersections. (d) Reference roads.

TABLE I
COMPARISON OF THE PERFORMANCE OF ROAD DETECTION APPROACH PRESENTED IN THIS PAPER

Image ID	Image Size	Seeding Type	Num. Seeds	No. Extracted Vertices	Num. Pruned Vertices	Time (minutes)
Fig. 11	860x660	manual	1	382	345	1.25
Fig. 13(c)	895x670	manual	20	1577	198	2.2
Fig. 13(d)	895x670	auto	31	1748	370	6.5
Fig. 14	820x630	manual	10	1512	601	1.7
Fig. 15	884x800	auto	65	1935	675	12.1
Fig. 16	824x754	auto	76	2232	452	8.4

the quality of the results by automatic seeding is higher than that by manual seeding, and the completeness and correctness by automatic seeding can match those by manual seeding. This is because automatic seeding scans the whole image to search all possible seeds. As a side note, it is difficult to compare the performance of the different road detectors and trackers against each other since they all use different resolution imagery from different sources. The most related work is that of Jin and Davis [31], where 70–86% completeness and 70–92% correctness were achieved for a set of three-channel satellite image subsets. Here, Table II shows that the completeness of our road

TABLE II
EVALUATION RESULTS OF THE PROPOSED METHOD

Image ID	Completeness %	Correctness %	Quality %
Fig. 13(manual)	89	91	82
Fig. 13(auto)	91	90	86
Fig. 14	95	81	82
Fig. 15	85	91	85
Fig. 16	93	96	92

tracker for gray images based on automatic seeding ranges from 82% to 94%, correctness is above 90%, and quality is from 85% to 92%.

V. CONCLUSION

In this paper, an adaptive unsupervised approach to extract the inscribed lines of road networks and identify the road intersections has been presented. Our method includes both detecting and pruning level analysis.

The spoke wheel operator properly detects footprints of pixels in an aerial image to represent their local homogeneous regions. The toe-finding algorithm finds the dominant directions of the road footprint and makes it possible to initialize and track a road segment automatically. Our road tracker based on shape classification of the road footprints effectively detects road intersections, extracts almost all inscribed lines of road networks, and forms a road tree. However, the multidirectional road tracker suffers from overextraction. It is necessary to remove portions in the road tree that do not appear to be roads.

An original road tree pruning approach has been presented, which is based on a Bayesian decision model defined on the road tree, and takes into account the basic property of road network: width of a road is constant or piecewise constant. A lognormal distribution is used to characterize the A/P ratios (only dependent on the road width) of the footprints in the road tree and provides a suit prior knowledge for the Bayes decision model. Our various experimental results have shown that our road tree pruning approach efficiently trims the paths leaking into the surroundings of the roads, significantly improves the performance of our road tracker, and increases correctness and quality value of the extracted results.

Future work includes improving the footprint classification method based on Fourier shape description to decrease the amount of leakage and finding a global connecting method that deals with the connections among road trees.

ACKNOWLEDGMENT

The authors would like to thank the National Geospatial-Intelligence Agency for providing the data.

REFERENCES

- [1] M. F. Auclair Fortier, D. Ziou, C. Armenakis, and S. Wang, "Automated correction and updating of roads from aerial ortho-images," in *Proc. ISPRS*, 2000, vol. 33, pp. 34–41.
- [2] J. B. Mena, "State of the art on automatic road extraction for gis update: A novel classification," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 3037–3058, Dec. 2003.
- [3] M. Xia and B. Liu, "Image registration by super-curves," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 720–732, May 2004.
- [4] M. Cui, P. Wonka, A. Razdan, and J. Hu, "A new image registration scheme based on curvature scale space curve matching," *Vis. Comput.*, vol. 23, no. 8, pp. 607–618, Jul. 2007.
- [5] N. C. Rowe and L. L. Grewe, "Change detection for linear features in aerial photographs using edge-finding," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1608–1612, Jul. 2001.
- [6] C. C. Chen, S. Thakkar, C. A. Knoblock, and C. Shahabi, "Automatically annotating and integrating spatial datasets," in *Proc. Spatio-Temporal Database Manage.*, 2003, pp. 469–488.
- [7] C. C. Chen, C. Shahabi, and C. A. Knoblock, "Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery," in *Proc. Spatio-Temporal Database Manage.*, 2004, pp. 17–24.
- [8] L. Bentabet, S. Jodouin, D. Ziou, and J. Vaillancourt, "Road vectors update using sar imagery: A snake-based method," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 8, pp. 1785–1803, Aug. 2003.
- [9] R. Bajcsy and M. Tavakoli, "Computer recognition of roads from satellite pictures," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 9, pp. 623–637, Sep. 1976.
- [10] Y. T. Zhou, V. Venkateswar, and R. Chellapa, "Edge detection and linear feature extraction using a 2-D random field model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 1, pp. 84–95, Jan. 1989.
- [11] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf, "Detection of roads and linear structure in low-resolution aerial imagery using a multisource knowledge integration technique," *Comput. Graph. Image Process.*, vol. 15, no. 3, pp. 201–223, Mar. 1981.
- [12] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [13] J. Chanussot, G. Mauris, and P. Lambert, "Fuzzy fusion techniques for linear features detection in multitemporal SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1292–1305, May 1999.
- [14] A. Katartzis, H. Sahli, V. Pizurica, and J. Cornelis, "A model-based approach to the automatic extraction of linear features from airborne images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 9, pp. 2073–2079, Sep. 2001.
- [15] D. M. McKeown and J. L. Denlinger, "Cooperative methods for road tracking in aerial imagery," in *Proc. CVPR*, 1988, pp. 662–672.
- [16] G. Vosselman and J. Knecht, "Road tracing by profile matching and Kalman filtering," in *Automatic Extraction of Man-Made Objects From Aerial and Space Images*. Basel, Switzerland: Birkhuser Verlag, 1995, pp. 265–274.
- [17] J. Zhou, W. F. Bischof, and T. Caelli, "Robust and efficient road tracking in aerial images," in *CMRT IAPRS*, 2005, vol. 36, pp. 35–40.
- [18] F. Tupin, H. Maitre, J. F. Mangin, J. M. Nicolas, and E. Pechersky, "Detection of linear features in sar images: Application to road network extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 2, pp. 434–453, Mar. 1998.
- [19] A. C. Bovik, "On detecting edges in speckle imagery," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 10, pp. 1618–1627, Oct. 1988.
- [20] R. Huber and K. Lang, "Road extraction from high-resolution airborne SAR using operator fusion," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2001, vol. 6, pp. 2813–2815.
- [21] P. Gamba, F. Dell'Acqua, and G. Lisini, "Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping, and simple topological concepts," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 3, pp. 387–391, Jul. 2006.
- [22] W. Shi and C. Zhu, "The line segment match method for extracting road network from high-resolution satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 2, pp. 511–514, Feb. 2002.
- [23] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 113–125, Feb. 1998.
- [24] M. Barzohar and D. B. Cooper, "Automatic finding of main roads in aerial images by using geometric stochastic models and estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 707–721, Jul. 1996.
- [25] F. Tupin, B. Houshmand, and M. Datcu, "Road detection in dense urban areas using SAR imagery and the usefulness of multiple views," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2405–2414, Nov. 2002.
- [26] M. Negri, P. Gamba, G. Lisini, and F. Tupin, "Junction-aware extraction and regularization of urban road networks in high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2962–2971, Oct. 2006.
- [27] A. Baumgartner, S. Hinz, and C. Wiedemann, "Efficient methods and interfaces for road tracking," *Int. Arch. Photogramm. Remote Sens.*, vol. 34, pp. 28–31, 2002.
- [28] V. Amberg, M. Coulon, P. Marthon, and M. Spigai, "Structure extraction from high resolution SAR data on urban areas," in *Proc. IEEE Geosci. Remote Sens. Symp.*, 2004, vol. 3, pp. 1784–1787.
- [29] F. Deschenes and D. Ziou, "Detection of line junctions and line terminations using curvilinear features," Dept. Math. and Inf., Univ. Sherbrooke, Sherbrooke, QC, Canada, 1999. Tech. Rep.
- [30] C. Wiedemann and H. Ebner, "Automatic completion and evaluation of road networks," *Int. Arch. Photogramm. Remote Sens.*, vol. 33, pp. 979–986, 2000.
- [31] X. Jin and C. H. Davis, "An integrated system for automatic road mapping from high-resolution multi-spectral satellite imagery by information fusion," *Inf. Fusion*, vol. 6, no. 4, pp. 257–273, Dec. 2005.
- [32] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 1–14, Jan. 1996.
- [33] L. Gibson, "Finding road networks in Ikonos satellite imagery," in *Proc. ASPRS*, 2003, pp. 1200–1209.

- [34] Q. Zhang and I. Couloigner, "Benefit of the angular texture signature for the separation of parking lots and roads on high resolution multispectral imagery," *Pattern Recognit. Lett.*, vol. 27, no. 9, pp. 937–946, Sep. 2006.
- [35] P. L. Rosin, "Measuring rectangularity," *Mach. Vis. Appl.*, vol. 11, no. 4, pp. 191–196, Dec. 1999.
- [36] H. Cheng and Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2071–2082, Dec. 2000.
- [37] M. Amo, F. Martinez, and M. Torre, "Road extraction from aerial images using a region competition algorithm," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1192–1201, May 2006.
- [38] S. Krishnamachari and R. Chellappa, "Delineating buildings by grouping lines with MRF's," *IEEE Trans. Image Process.*, vol. 5, no. 1, pp. 164–168, Jan. 1996.
- [39] S. Hinz, "A fusion strategy for extraction of urban road nets from multiple images," in *Proc. IGARSS*, 2003, vol. 2, pp. 1059–1061.
- [40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.
- [41] K. Hedman, S. Hinz, and U. Stilla, "A probabilistic fusion strategy applied to road extraction from multi-aspect SAR data," *Int. Arch. Photogramm. Remote Sens.*, vol. 36, no. 3, pp. 55–60, 2006.
- [42] E. Limpert, W. Stahel, and M. Abbt, "Log-normal distributions across the sciences: Keys and clues," *Bioscience*, vol. 51, no. 5, pp. 341–352, May 2001.
- [43] C. Wiedemann, C. Heipke, and H. Mayer, "Empirical evaluation of automatically extracted road axes," in *Proc. CVPR Workshop Empirical Eval. Methods Comput. Vis.*, Los Alamitos, CA, 1998, pp. 172–187.



John C. Femiani is currently working toward the Ph.D. degree at the Department of Computer Science and Engineering, Arizona State University, Tempe.

His research interests include pattern recognition, computer graphics, and image processing.



Ming Cui received the B.E. degree in civil engineering and the M.S. degree in computer science from Zhejiang University, Hangzhou, China, in 2002 and 2005, respectively. He is currently working toward the Ph.D. degree at Arizona State University, Tempe.

His research interests include computer graphics and image processing.



Peter Wonka received the M.S. degree in urban planning and the Ph.D. degree in computer science from the Technical University of Vienna, Vienna, Austria.

Prior to coming to Arizona State University, Tempe, he was a Postdoctorate Researcher at the Georgia Institute of Technology, Atlanta, for two years. His research interests include various topics in computer graphics, particularly real-time rendering and procedural modeling.



Jiuxiang Hu received the Ph.D. degree in mathematics from Lanzhou University, Lanzhou, China, in 1994.

Since receiving the Ph.D. degree, he was a Postdoc from 1995 to 1996. He then joined the Department of Computer Science and Software Engineering School, Huazhong University of Science and Technology, Wuhan, China, where he was promoted to an Associate Professor in 1997. He is currently a Research Scientist with the I3DEA Lab, Division of Computing Studies, Arizona State University's

Polytechnic Campus, Mesa. His research interests include computer graphics, visualization, statistical and geometric techniques for image processing, and computer vision.



Anshuman Razdan received the B.S. degree in mechanical engineering from the Regional Engineering College, Kurukshetra, India, and the M.S. degree in mechanical engineering and the Ph.D. degree in computer science from Arizona State University (ASU), Tempe.

He is currently an Associate Professor with the Division of Computing Studies and the Director of the Advanced Technology Innovation Collaboratory and the I3DEA Lab, ASU's Polytechnic Campus, Mesa. He has been a pioneer in computing-based interdisciplinary collaboration and research at ASU. His research interests include geometric design, computer graphics, document exploitation, and geospatial visualization and analysis. He is the Principal Investigator and a Collaborator on several federal grants from agencies including the National Science Foundation, the National Geospatial-Intelligence Agency, and the National Institutes of Health.

Dr. Razdan is a member of the IEEE Computer Society.