ISPRS Journal of Photogrammetry and Remote Sensing 65 (2010) 165–181

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Delineation and geometric modeling of road networks

Charalambos Poullis *, Suya You

Computer Graphics and Immersive Technologies Lab, Integrated Media Systems Center, University of Southern California, United States

ARTICLE INFO

Article history:
Received 7 April 2008
Received in revised form
2 October 2009
Accepted 6 October 2009
Available online 14 November 2009

Keywords:
Road extraction
Network delineation
Road detection
Road modeling

ABSTRACT

In this work we present a novel vision-based system for automatic detection and extraction of complex road networks from various sensor resources such as aerial photographs, satellite images, and LiDAR. Uniquely, the proposed system is an integrated solution that merges the power of perceptual grouping theory (Gabor filtering, tensor voting) and optimized segmentation techniques (global optimization using graph-cuts) into a unified framework to address the challenging problems of geospatial feature detection and classification.

Firstly, the local precision of the Gabor filters is combined with the global context of the tensor voting to produce accurate classification of the geospatial features. In addition, the tensorial representation used for the encoding of the data eliminates the need for any thresholds, therefore removing any data dependencies.

Secondly, a novel orientation-based segmentation is presented which incorporates the classification of the perceptual grouping, and results in segmentations with better defined boundaries and continuous linear segments.

Finally, a set of gaussian-based filters are applied to automatically extract centerline information (magnitude, width and orientation). This information is then used for creating road segments and transforming them to their polygonal representations.

1. Introduction

Recent technological advancements have caused a significant increase in the amount of remote sensor data and of their uses in various applications. Efficient and inexpensive techniques in the area of data acquisition have popularized the use of remote sensor data and led to their widespread availability. However, the interpretation and analysis of such data still remains a difficult and manual task. Specifically in the area of road mapping, traditional methods require time-consuming and tedious manual work which does not meet the increasing demands and requirements of current applications. Although considerable attention has been given on the development of automatic road extraction techniques it still remains a challenging problem due to the wide variations of roads (urban, rural, etc) and the complexities of their environments (occlusions due to cars, trees, buildings, etc).

In this work we focus on the automatic and reliable detection and extraction of transportation networks from remote sensor data including aerial photographs, satellite images, and LiDAR. We present an integrated solution that merges the strengths of perceptual grouping theory (Gabor filters, tensor voting) and segmentation (global optimization by graph-cuts), under a unified framework to address the challenging problem of automated feature detection, classification and extraction. The proposed approach leverages the multi-scale, multi-orientation capabilities of Gabor filters for the inference of geospatial features, the effective and robust handling of noisy, incomplete data of tensor voting for the feature classification and the fast and efficient optimization of graph-cuts for the segmentation and labeling of road features.

2. Related work

A plethora of work has been proposed for solving the complex problem of extracting road networks from remote sensor data. Almost all of the existing work shares similar processing pipeline and relies on the combination of pixel-based, region-based and knowledge-based techniques. However, several distinctions exist between the different processing components. Below is an overview of the state-of-the-art in this area. Mayer et al. (2006) offers a comprehensive survey on the state-of-the-art road extraction techniques from a variety of different datasets.

In Baumgartner et al. (1999) lines are extracted in an image with reduced resolution as well as road-side edges in the original high resolution image. Using both resolution levels and explicit knowledge about roads, hypotheses for road segments are generated and

* Corresponding author. Tel.: ++1 3102109787.
E-mail addresses: charalambos@poullis.org (C. Poullis), suyay@graphics.usc.edu (S. You).

0924-2716/$ – see front matter
doi:10.1016/j.isprsjprs.2009.10.004

are grouped iteratively into larger segments. Although the results seem promising, the proposed method is focused on extracting road networks for rural areas.

Lisini et al. (2004) presents a system which relies on adaptive filtering to determine predominant orientations of the roads. The response of the filtering is then used to extract linear segments which are then connected based on tolerances determined by the spatial resolutions. This approach relies on various hard thresholds and data-dependent parameters thus requires considerable user interaction to tune the parameters prior to processing the data.

A different approach which tries to reduce the number of tunable parameters is presented in Laptev et al. (2000). The authors propose the integration of the well-established techniques of multi-scale image processing and active contour models to resolve the complex problem of road extraction. They use a multi-scale ridge detector for the detection of lines at a coarser scale, and then use a local edge detector at a finer scale for the extraction of parallel edges which are optimized using a variation of the active contour models technique (snakes). The results indicate that the approach performs very well especially for rural areas.

Similarly, Wessel (2004) employs Steger's differential geometry approach (Mayer and Steger, 1998) for the extraction of linear segments. Context information about road networks is then used to connect the linear segments into roads. Steger's differential geometry approach is also employed in Bacher and Mayer (2005) for the extraction of linear segments from multi-spectral images. The extracted lines are then used for training through an automatic supervised classification to produce a road class image which can be used to verify road hypotheses. The approach has been shown to perform well on rural areas only.

The authors in Barsi and Heipke (2003) present an approach for extracting road junctions. To achieve this they train a feed-forward artificial neural network to learn a junction model which supports junctions of up to four arms. The training is performed interactively and the junctions are extracted using a Deriche operator for the edge detection with an added hysteresis threshold, followed by an edge smoothing using the Ramer algorithm. Although the result is not a complete road network the approach seems to perform very well for rural areas.

The system in Zhang et al. (2001) integrates knowledge processing of color image data and information from digital geographic databases, extracts and fuses multiple object cues, thus takes into account context information, employs existing knowledge, rules and models, and treats each road subclass accordingly. Clode et al. (2005) uses a rule-based algorithm for the detection of buildings at a first stage and then at a second stage the reflectance properties of the road. Similarly, Zhang and Couloigner (2006) uses reflectance as a measure for the image segmentation and clustering. Explicit knowledge about geometric and radiometric properties of roads is used in Wessel (2004) to construct road segments from the hypotheses of road-sides. In Barsi and Heipke (2003) the developed system can detect a variety of road junctions using a feed-forward neural network, which requires collected data for the training of the network. Peteri et al. (2003) take high resolution images as input along with prior knowledge about the roads e.g. road models and road properties.

In Porikli (2003) the authors present an approach based on point-wise Gaussian models. A set of quadruple line filters is applied on the image to extract linear segments. Additionally, road points which are not perceptible by the line filters are enhanced using the likelihood of each image point as being part of a road. The results are impressive however, this approach only deals with images where the roads appear as thin linear features and have no width.

A method which relies on elevation data is presented in Clode et al. (2005). LiDAR data provides accurate elevation information which can be used to resolve problems occurring using optical imagery such as road overlaps due to bridges. A region growing algorithm is used to segment the road segments from other points in the data such as buildings, trees, etc. The road candidates are then vectorized using a phase-coded disk which allows the extraction of roads of different widths and different orientations.

The importance of scale-space processing is described in the work of Mayer and Steger (1998). Building on similar concepts, the authors in Heller and Pakzad (2005) present a concept to automatically adapt road models for high resolution images to models appropriate for images of lower resolution with similar spectral characteristics. Additionally, in Heuwold (2006) the author presents a framework for the verification of the automatic adaptation of object models consisting of parallel line-type objects parts to a lower image resolution. Similarly, in Hinz and Baumgartner (2003) the authors present an automatic road extraction technique by integrating detailed knowledge about roads and their context using explicitly formulated scale-dependent models. A slightly different approach which combines a scale-space processing framework with the introduction of Markov random fields is presented in Tupin et al. (2002).

On a different note, the authors in Mena and Malpica (2005) present an automatic method for road extraction which uses a new technique, named Texture Progressive Analysis and consists of a fusion of information streaming from three different sources for the image. The approach was successfully applied on rural as well as semi-urban areas with successful results.

Zhou et al. (2007) present a user-guided image interpretation system which integrates inputs from human experts with computational algorithms in order to learn road tracking. Although the results seem promising, the goal of completely eliminating the need for human intervention and interactions is still not achieved.

An approach which combines a line-based road extraction and area-based color segmentation techniques is presented in Ziems et al. (2007). They show that the incorporation of prior information into the line-based road extraction algorithm allows the robust estimation and automatic tune-up of parameters that control the contrast between road and background, the homogeneity within the road objects and the global threshold for masking out non-road areas.

The aforementioned work clearly indicates that the predominant approach for addressing the complex problem of road extraction involves the multi-scale processing of the input data. In addition to the scale-space processing, an imperative part of road extraction systems is the elimination of data-dependent parameters since this directly affects the applicability of the system. Although very impressive and promising results have already been reported as mentioned above, the majority of the existing work in the area focuses on particular types of datasets (i.e. LiDAR or satellite images) and/or particular types of scenes (i.e. rural, urban, forest, etc). The result is road extraction systems which perform well for one type but fail for another unless numerous parameters are fine-tuned.

Hence, the goal of our work is to design and develop a system which relies on well-established computer vision techniques, incorporates scale-space processing, requires no (or minimal and stable) parameter tuning and can simultaneously process various remote sensor data such as LiDAR, intensity response and satellite imagery. The solution to these problems is sought in the development of a novel system which combines the strengths of perceptual grouping (Gabor filters, Tensor Voting) and global optimization (Graph-Cuts) for the geospatial feature inference and classification. As a result, the proposed system has no data dependencies and requires minimal parameters which were found to be stable and remain fixed for all the examples presented (scale factor for the Tensor voting, number of labels and smoothness factor for the optimization). The results shown in Section 7 indicate the high success rate of our system on all types of datasets and scenes, and verify the validity of the approach.

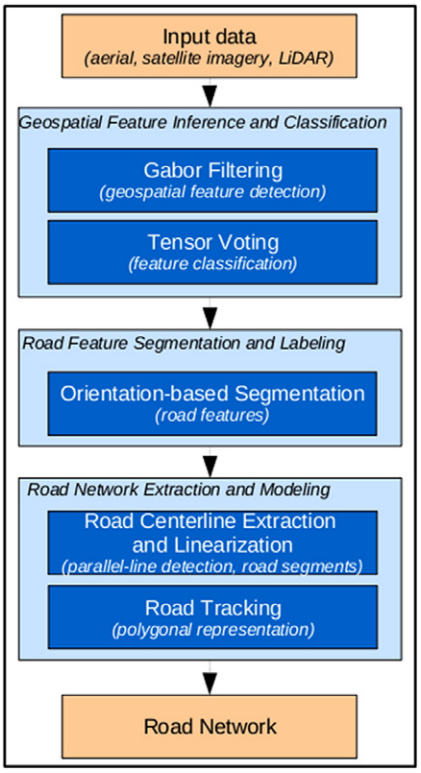


Fig. 1. System overview.

3. System overview

Although many different approaches have been proposed and developed for the automatic extraction of road networks, it still remains a challenging problem due to the wide variations of roads e.g. urban, rural, mountainous etc and the complexities of their environments e.g. occlusions due to cars, trees, buildings, shadows etc. For this reason, traditional techniques such as pixel- and region-based have several problems and often fail when dealing with complex road networks. Our proposed approach addresses these problems and provides solutions to the difficult problem of automatic road extraction. Fig. 1 summarizes our approach.

Firstly, geospatial feature inference and classification. Local orientation information is extracted using a bank of Gabor filters, which is encoded into a tensorial representation. This representation can simultaneously capture the geometric information of multiple feature types passing through a point (surface, curve, junction) and an associated measure of the likelihood of that point being part of each type. A tensor voting is then performed which globally communicates and refines the information carried at each point. An important advantage of combining Gabor filters and tensor voting for the classification is that it eliminates the need for hard thresholds. Instead, the refined likelihoods of each point give an accurate estimate of the dominant feature passing through that point, and are therefore used for the classification into curve and junction features. Furthermore, it removes the limitation of tensor voting to work only with binary images and extends its application to grayscale images.

Secondly, road feature segmentation and labeling. A novel orientation-based segmentation using graph-cuts is performed. An important aspect of this segmentation is that it incorporates the orientation information of the classified curve features and favors towards keeping those curves connected. The result is a binary segmentation into road and non-road candidates.

Finally, road network extraction and modeling. A pair of gaussian-based bi-modal and single mode kernels are developed for the automatic detection of road centerlines and the extraction of width and orientation information from the segmented road candidates. Linear segments resulting from the application of an iterative Hough transform on the road centerlines, are validated and refined (merge, split, approximate, smooth). Using the automatically extracted width and orientation information, a tracking algorithm converts the refined linear segments into their equivalent polygonal representations.

4. Geospatial feature inference and classification

4.1. Gabor filtering

A 2D Gabor function $g(x, y)$ in spatial frequency domain is given by,

$$g(x, y) = c(x, y) \times e(x, y) \tag{1}$$

where $c(x, y)$ is a complex sinusoidal, known as the carrier, and $e(x, y)$ is a 2D Gaussian function, known as the envelope.

The complex sinusoidal carrier is defined as,

$$c(x, y) = e^{j(2\pi(u_0 x + v_0 y) + \phi)} \tag{2}$$

where (u_0, v_0) is the spatial frequency and ϕ is the phase of the sinusoidal. The spatial frequency can also be expressed in polar coordinates as magnitude F_0 and direction ω_0. The 2D Gaussian envelope is defined as,

$$e(x, y) = Ae^{(-\pi(s_x^2(x-x_0)_\vartheta^2 + s_y^2(y-y_0)_\vartheta^2))} \tag{3}$$

where A is a scale of the magnitude, (s_x, s_y) are scale factors for the axes, (x_0, y_0) is the peak coordinates and ϑ is the rotation angle.

An attractive characteristic of the Gabor filters is their ability to tune at different orientations and frequencies. Thus by fine-tuning the filters we can extract high-frequency oriented information such as discontinuities and ignore the low-frequency clutter.

We employ a bank of Gabor filters tuned at 8 different orientations θ linearly varying from $0 \leq \theta < \pi$, and at 5 different high frequencies (per orientation) to account for multi-scale analysis. The remaining parameters of the filters in Eq. (3) are computed as functions of the orientation and frequency parameters as in Manjunath and Ma (1996).

The application of the bank of Gabor filters results in a total of 40 response images (8 orientations $\times$5 frequencies) as shown in the Table 1. The response images corresponding to filters of the same orientation and different frequency are added together. The result is a single response image per orientation (total of 8) which is then encoded using a tensorial representation as explained in Section 4.2.

4.2. Tensor voting

Tensor voting is a perceptual grouping and segmentation framework introduced by Medioni et al. (2000). A key data representation based on tensor calculus is used to encode the data. A point $x \in \mathbb{R}^3$ is encoded as a second-order symmetric tensor T and is defined as,

$$T = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \vec{e}_1^T \\ \vec{e}_2^T \\ \vec{e}_3^T \end{bmatrix} \tag{4}$$

$$T = \lambda_1 \vec{e}_1 \vec{e}_1^T + \lambda_2 \vec{e}_2 \vec{e}_2^T + \lambda_3 \vec{e}_3 \vec{e}_3^T \tag{5}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are eigenvalues, and $\vec{e}_1, \quad \vec{e}_2, \quad \vec{e}_3$ are the eigenvectors corresponding to $\lambda_1, \lambda_2, \lambda_3$ respectively. By applying the spectrum theorem, the tensor T in Eq. (5) can be expressed as a linear combination of three basis tensors (ball, plate and stick) as in Eq. (6).

Table 1
Gabor filters are applied at 8 different orientations and 5 different high frequencies. Output images of the same orientation (and varying frequency) are grouped together resulting in a total of 8 images (one for each orientation) as shown in the last column. Similarly, the 8 images can then be grouped together resulting in a single image depicting the detected edges.

Freq.($\phi \rightarrow$)-Orient.($\theta \downarrow$)	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4	$\sum_{i=0}^{4}(\phi_i)$
θ_0						
θ_1						
θ_2						
θ_3						
θ_4						
θ_5						
θ_6						
θ_7						
$\sum_{j=0}^{7}\theta_j$	×	×	×	×	×	

Fig. 2. (a) Tensor decomposition into the stick, plate and ball basis tensors in 3D. (b) Votes cast by a stick tensor located at the origin O. C is the center of the osculating circle passing through points P and O.

$$T = (\lambda_1 - \lambda_2)\vec{e}_1\vec{e}_1^T + (\lambda_2 - \lambda_3)(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T) + \lambda_3(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T + \vec{e}_3\vec{e}_3^T). \quad (6)$$

In Eq. (6), $(\vec{e}_1\vec{e}_1^T)$ describes a stick (surface) with associated saliency $(\lambda_1 - \lambda_2)$ and normal orientation $\vec{e}_1$, $(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T)$ describes a plate (curve) with associated saliency $(\lambda_2 - \lambda_3)$ and tangent orientation $\vec{e}_3$, and $(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T + \vec{e}_3\vec{e}_3^T)$ describes a ball (junction) with associated saliency λ_3 and no orientation preference. The geometric interpretation of tensor decomposition is shown in Fig. 2(a).

An important advantage of using such a tensorial representation is its ability to capture the geometric information for multiple feature types (junction, curve, surface) and a saliency, or likelihood, associated with each feature type passing through a point.

Every point (x, y) in the Gabor filter response images computed previously is encoded using Eq. (4) into a unit plate tensor (representing a curve) with the orientation $\vec{e}_3$ aligned to each filter's G_i orientation and is scaled by the magnitude of the response of that point $(G_i \otimes I)_{x,y}$. The resulting eight tensors for each point are then added together which produces a single tensor $T_{(\mathbf{x},\mathbf{y})}$ per point

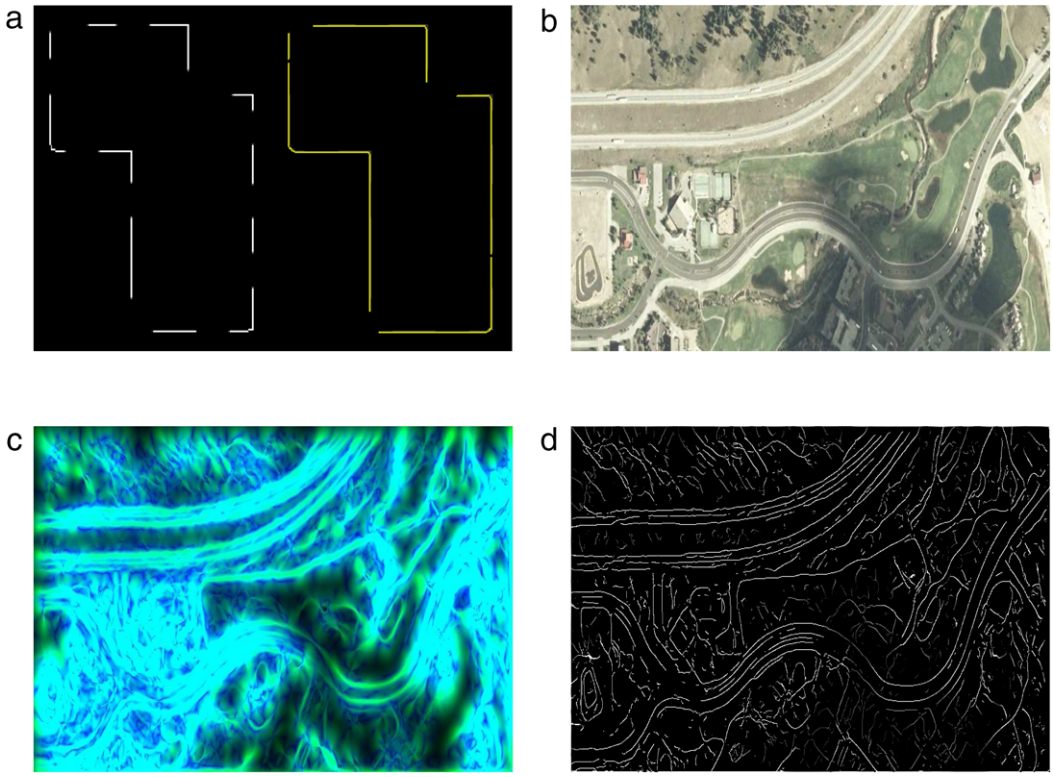


Fig. 3. (a) Successful handling of discontinuities. Before (left) and after (right) the tensor voting process. (b) Original image of Copper Mountain area in Colorado. (c) Saliency map indicating the refined likelihoods produced by the tensor voting. Green indicates curve-ness ($\lambda_2 - \lambda_3$), blue indicates junction-ness (λ_3). (d) Classified curve features derived from 3(c). Note that no thresholds were used. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

capturing the local geometric information and is given by,

$$T_{gabor} = \sum_{i=0}^{8}((G_i \otimes I)_{x,y} * T_{x,y,i}). \tag{7}$$

Using the tensor decomposition Eq. (6), all pixels for which $(\lambda_2 - \lambda_3) > \lambda_3$ are classified as part of curves with tangent orientation $\vec{e}_3$. Similarly all pixels for which $\lambda_3 > (\lambda_2 - \lambda_3)$ are classified as junction points with no orientation preference.

For example, if a point p_c lies along a curve in the original image its highest response will be at the Gabor filter with a similar orientation as the direction of the curve. Encoding the eight responses of pixel p_c as unit plate tensors, scaling them with the point's response magnitudes and adding them together results in a tensor where $(\lambda_2 - \lambda_3) > (\lambda_1 - \lambda_2)$, $(\lambda_2 - \lambda_3) > \lambda_3$ and the orientation $\vec{e}_3$ is aligned to the direction of the curve i.e. a plate tensor. Similarly a tensor representing a point p_j which is part of a junction will have $\lambda_3 > (\lambda_2 - \lambda_3)$, $\lambda_3 > (\lambda_2 - \lambda_3)$ i.e. a ball tensor.

The encoded points then cast a vote to their neighboring points which lie inside their voting fields, thus propagating and refining the information they carry. The strength of each vote decays with increasing distance and curvature as specified by each point's stick, plate and ball voting fields. The three voting fields can be derived directly from the saliency decay function (Guy and Medioni, 1997) given by

$$DF(s, \kappa, \sigma) = e^{-\left(\frac{s^2 + c\kappa^2}{\sigma^2}\right)} \tag{8}$$

where s is the arc length of OP, κ is the curvature, c is a constant which controls the decay with high curvature (and is a function of σ), and σ is a scale factor which defines the neighborhood size as shown in Fig. 2(b). The blue arrows at point P indicate the two types of votes it receives from point O: (1) a second-order vote which is a second-order tensor that indicates the preferred orientation at the receiver according to the voter and (2) a first-order vote which is a first-order tensor (i.e. a vector) that points toward the voter along the smooth path connecting the voter and receiver. The scale factor σ is the only free variable in the framework.

After the tensor voting the refined information is analyzed and used to classify the points as curve or junction features. An example of a mountainous area with curvy roads is shown in Fig. 3(b). A saliency map indicating the likelihood of each point as being part of a curve (green) and a junction (blue) is shown in Fig. 3(c). The saliency map is used for the classification of the curve points which are shown in Fig. 3(d). A point with $(\lambda_2 - \lambda_3) > \lambda_3$ is classified as a curve point and a point with $\lambda_3 > (\lambda_2 - \lambda_3)$ is classified as a junction point. Intuitively, a greener point is a curve and a bluer point is a junction.

A key advantage of combining the Gabor filtering and tensor voting is that it eliminates the need for any thresholds therefore removing any data dependencies. The local precision of the Gabor filters is used to derive information which is *directly* encoded into tensors. The tensors are then used as an initial estimate for global context refinement using tensor voting and the points are classified based on the their *likelihoods* of being part of a feature type. This unique characteristic makes the process invariant to the type of images being processed. In addition, the global nature of tensor voting makes it an ideal choice when dealing with noisy, incomplete and complicated images and results in highly accurate estimates about the image features. This is demonstrated in Fig. 3(a) where the original image shows a polygon with many gaps of different sizes in white and the recovered, classified curve points are shown in yellow. As it can be seen most of the discontinuities were successfully and accurately recovered.

5. Road feature segmentation and labeling

The classification of tensor voting provides an accurate measure of the type of each feature i.e. junctions and curves. However, these features result from the presence of roads as well as buildings, cars, trees, etc. A segmentation process is performed to segment only the road features from the classified curve features. The geometric structure of the curve features combined with color information extracted from the image, is used to guide an orientation-based segmentation using optimization by graph-cuts which produces a labeling of road and non-road candidates.

5.1. Labels

The binary label case of graph-cuts described in Appendix, can easily be extended to a case of multiple terminal vertices. We create two terminal vertices for foreground O and background B pixels for each orientation θ for which $0 \leq \theta \leq \pi$. In our experiments, we have found that choosing the number of orientation labels in the range $2 \leq N_\theta \leq 16$ generates acceptable results. Thus the set of labels L is defined to be $L = \{O_{\theta_1}, B_{\theta_1}, O_{\theta_2}, B_{\theta_2} \ldots, O_{\theta_{N_\theta}}, B_{\theta_{N_\theta}}\}$ with size $|L| = 2 * N_\theta$.

5.2. Energy minimization function

Finding the minimum cut of a graph is equivalent to finding an optimal labeling $f : I_p \longrightarrow L$ which assigns a label $l \in L$ to each pixel $p \in I$ where f is piecewise smooth and consistent with the original data. Thus, our energy function for the graph-cut minimization is given by

$$E(f) = E_{data}(f) + \lambda * E_{smooth}(f) \tag{9}$$

where λ is the weight of the smoothness term.

Energy data term. The data term provides a per-pixel measure of how appropriate a label $l \in L$ is, for a pixel $p \in I$ in the *observed* data and is given by,

$$E_{data}(f) = \sum_{p \in I} D_p(f(p)). \tag{10}$$

As in Boykov et al. (2001), the initial seed points are used twice:

(1) To compute an intensity distribution (in our case color distribution using gaussian mixture models) for the background and foreground pixels. A measure of how appropriate a labeling is, is then given by computing the negative log-likelihood i.e. $-\ln(P(I_p|f(p)))$.
(2) To encode the hard constraints for the segmentation. Foreground and background pixels are assigned the lowest and highest value of the function $D_p(f(p))$, respectively. For all other pixels, D_p is computed as,

$$D_p(f(p)) = \frac{1 - \ln(P(I_p|f(p)))}{2 - \|\theta_p - \theta_{f(p)}\|^2}. \tag{11}$$

The energy data term then becomes,

$$E_{data}(f) = \sum_{p \in I} \left(\frac{1 - \ln(P(I_p|f(p)))}{2 - \|\theta_p - \theta_{f(p)}\|^2} \right). \tag{12}$$

Energy smoothness term. The smoothness term provides a measure of the difference between two neighboring pixels $p, q \in I$ with labels $l_p, l_q \in L$ respectively. Let I_p and I_q be the intensity values in the *observed* data of the pixels $p, q \in I$ respectively. Similarly, let θ_p and θ_q be the initial orientations for the two pixels recovered as explained in Section 4.2. We define a measure of the *observed* smoothness between pixels p and q as

$$\Delta_{p,q} = \frac{1 + (I_p - I_q)^2}{2 - \|\theta_p - \theta_q\|^2}. \tag{13}$$

In addition, we define a measure of smoothness for the global minimization. Let $I_{f(p)}$ and $I_{f(q)}$ be the intensity values under a labeling f. Similarly, let $\theta_{f(p)}$ and $\theta_{f(q)}$ be the orientations under the same labeling. We define a measure of the smoothness between neighboring pixels p, q under a labeling f as

$$\widehat{\Delta_{p,q}} = \frac{1 + (I_{f(p)} - I_{f(q)})^2}{2 - \|\theta_{f(p)} - \theta_{f(q)}\|^2}. \tag{14}$$

Using the smoothness measure defined for the observed data and the smoothness measure defined for any given labeling we can finally define the energy smoothness term as follows,

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f(p), f(q)) \tag{15}$$

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} K_{p,q} * \widehat{\Delta_{p,q}} \tag{16}$$

where N is the set of neighboring pixels, $K_{p,q} = [e^{-\frac{\Delta^2_{p,q}}{2*\sigma^2}}]$, and σ controls the smoothness uncertainty. Intuitively, if two neighboring pixels p and q have similar intensity and similar orientation in the observed data, then $\Delta_{p,q}$ will be small and thus there is a high probability of $\widehat{\Delta_{p,q}}$ being small. To summarize, the function $E(f)$ penalizes heavily for severed edges between neighboring pixels with similar intensity and orientation, and vice versa.

An advantage of the proposed orientation-based segmentation is that by incorporating orientation information in the optimization process it ensures that linear segments are not severed, even in the case where the color difference between neighboring pixels is relatively big. By using the classified curve feature information to guide the segmentation process we combine the fast computational times of graph-cuts and the high accuracy of the information derived using the perceptual grouping to produce results with better defined boundaries compared to traditional segmentation techniques as demonstrated in Fig. 4.

5.3. Segmentation results

As explained in Section 5.1 our method generates two types of information:

- a labeling which segments foreground and background pixels and
- a labeling which assigns an orientation to each pixel. For clarity, we show only the orientation information associated with the foreground pixels.

The examples demonstrate the effectiveness of this approach not only on images of road networks but also on general purpose. An 8-neighborhood system is used for computing the smoothness term, and is fixed for all examples shown. The number of orientation labels N_θ may vary and is specified for each example. User interaction is limited to specifying a set of seed points at the very beginning of the segmentation process and the results shown are the outputs after a single segmentation execution.

A simplistic case is shown in Fig. 5 that demonstrates the types of output of our segmentation. The input is a synthetic image shown in Fig. 5(a). The sparse orientation information computed as explained in Section 4 is shown in Fig. 5(b). The output is a foreground–background labeling shown in Fig. 5(c) and an orientation labeling shown in Fig. 5(d). The dense orientation information in Fig. 5(d) can be visually verified: the orientation of the horizontal segment is aligned with the X axis i.e. 0° and the orientation of the vertical segment is aligned with the Y axis i.e. 90°. At the junction

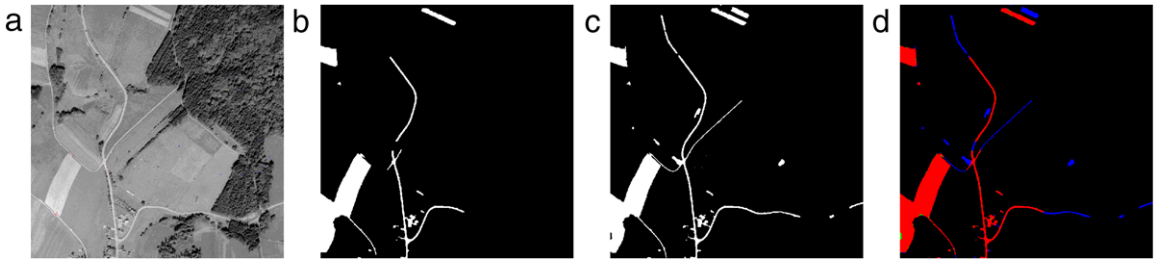
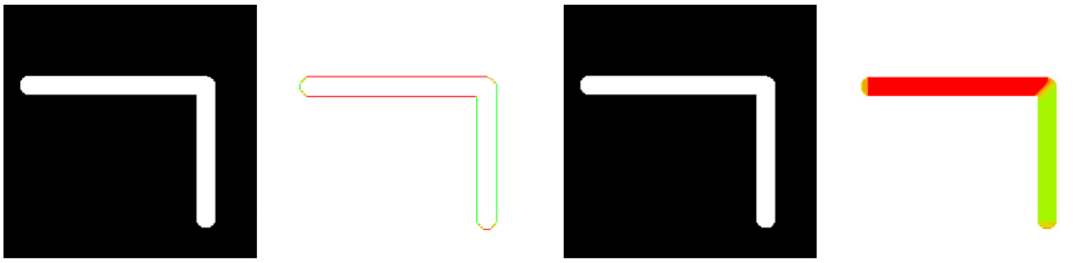
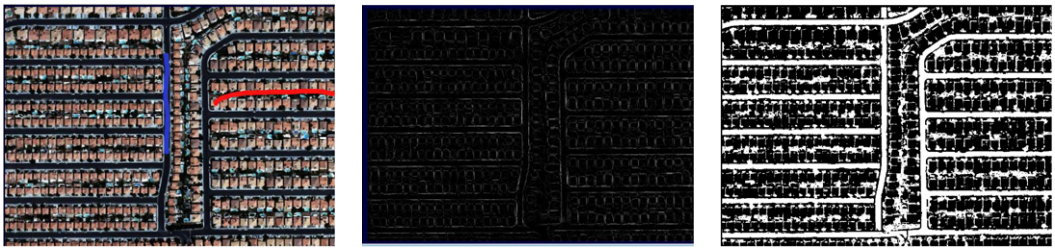


Fig. 4. Comparison between traditional intensity- and orientation-based segmentation. (a) Original image. (b) Intensity-based segmentation. (c) Orientation-based segmentation. (d) Color-coded segmentation difference (red: common points, green: only in intensity segmentation, blue: only in orientation segmentation). Image from Laptev et al. (2000). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Original. (b) Sparse orientation. (c) Color labels. (d) Dense orientation labels.

Fig. 5. Simple synthetic example which demonstrates the dense orientation output of the segmentation ($N_\theta = 8$).



(a) Input image. (b) Extracted curve points. (c) Segmentation.

Fig. 6. Segmentation example for urban area. ($\lambda = 0.1$, $N_\theta = 8$). The foreground and background pixels indicated by the user are shown in red and blue, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

point the orientation is aligned to the diagonal between the X and Y axes i.e. 45°.

Fig. 6 shows an example of an urban area. The foreground and background pixels are indicated by the user on the original image. A gaussian mixture model (GMM) is then used to compute the color distribution of these foreground/background areas. In contrast to Rother et al. (2004) we do not fix the order of the model to a particular number, but instead use a Minimum Description Length (MDL) estimator (Rissanen, 1983) to compute the order. Subsequently, the color distributions are incorporated during the computation of the energy functions in the graph-cut optimization.

A more complicated example is shown in Fig. 7 of a rural area. This example shows a comparison between intensity-based segmentation, our method and mean-shift segmentation and demonstrates a dramatic improvement of our segmentation method. The image is a complex road network shown in Fig. 7(q) which consists of many topologically-free linear features occupying the entire image space. The image is then segmented using different smoothness weights λ for the graph-cut methods and different color tolerances τ for the mean-shift. As it can be seen, our segmentation outperforms the other techniques and preserves the boundaries even for the high values of λ. The mean-shift segmentation also successfully detects most of the parts of the salient dark colored roads however it fails to detect the lesser salient roads as in Fig. 7(m)–(n). For the comparison the *same* seed points and smoothness λ were used.

6. Road network extraction and modeling

6.1. Road centerline extraction and linearization

The extraction of the road centerlines is performed using a set of gaussian-based filters. A bi-modal filter is employed to detect parallel lines and is defined as a mixture of gaussian kernels given by,

$$G_b = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}}\left[e^{-\left[\frac{(x-\frac{w}{2})_r^2}{\sigma_x^2}+\frac{y_r^2}{\sigma_y^2}\right]}+e^{-\left[\frac{(x+\frac{w}{2})_r^2}{\sigma_x^2}+\frac{y_r^2}{\sigma_y^2}\right]}\right] \tag{17}$$

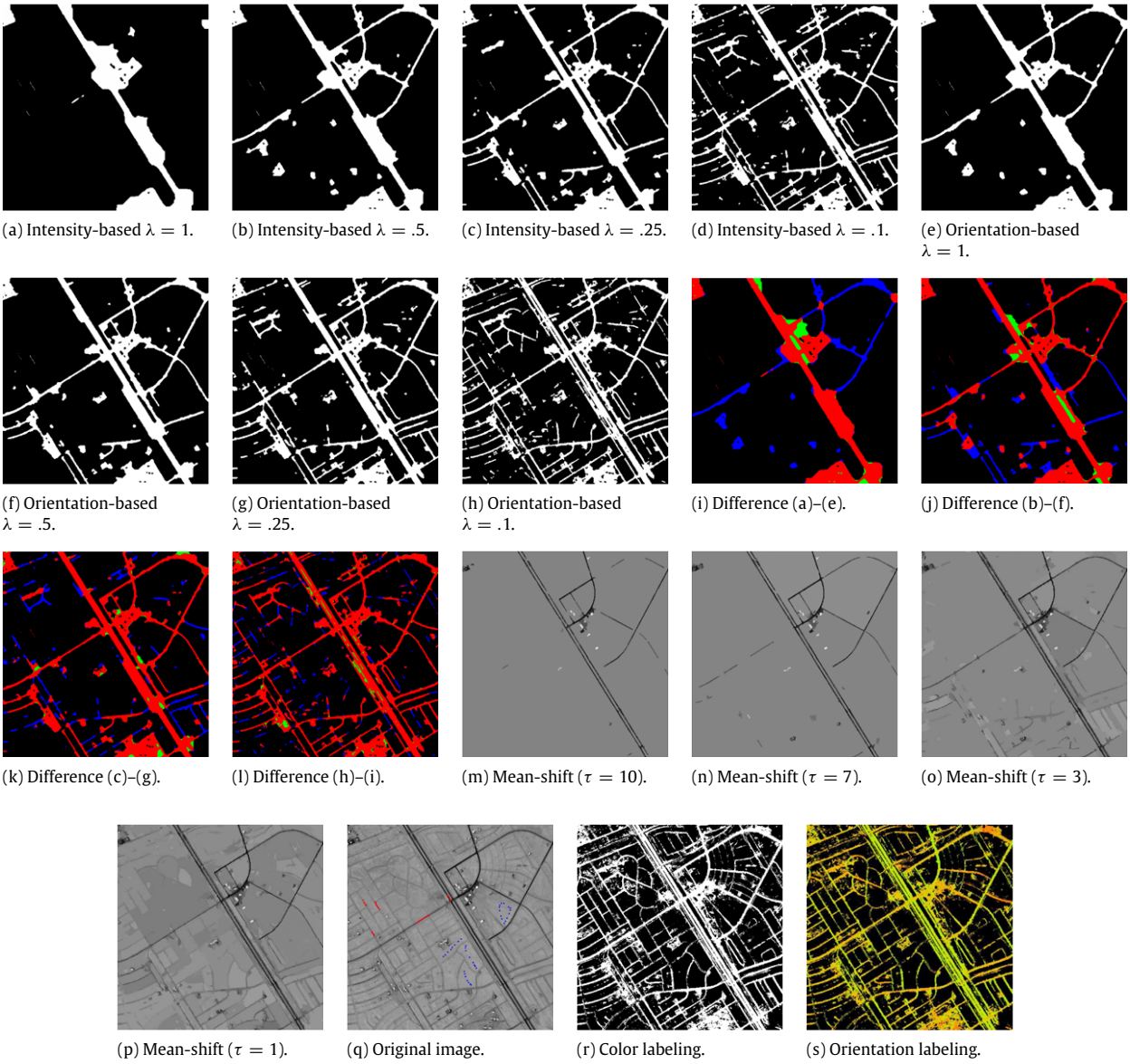
where the $(\ldots)_r$ subscript stands for a rotation operation such that

$$\left(x-\frac{w}{2}\right)_r = \left(x-\frac{w}{2}\right)\cos(\phi)+y\sin(\phi) \tag{18}$$

$$y_r = -\left(x-\frac{w}{2}\right)\sin(\phi)+y\cos(\phi) \tag{19}$$

where ϕ is the orientation of the filter and $0 \leq \phi \leq \pi$ and w is the distance between the peaks. The bi-modal filter is shown in Fig. 8(a).

Bi-modal *road-side* filters of different orientations ϕ and widths w are applied to the classified curve features computed previously as explained in Section 4. In order to overcome problems arising from the coincidental presence of two curve pixels along the filters'



(a) Intensity-based $\lambda = 1$. (b) Intensity-based $\lambda = .5$. (c) Intensity-based $\lambda = .25$. (d) Intensity-based $\lambda = .1$. (e) Orientation-based $\lambda = 1$.

(f) Orientation-based $\lambda = .5$. (g) Orientation-based $\lambda = .25$. (h) Orientation-based $\lambda = .1$. (i) Difference (a)–(e). (j) Difference (b)–(f).

(k) Difference (c)–(g). (l) Difference (h)–(i). (m) Mean-shift ($\tau = 10$). (n) Mean-shift ($\tau = 7$). (o) Mean-shift ($\tau = 3$).

(p) Mean-shift ($\tau = 1$). (q) Original image. (r) Color labeling. (s) Orientation labeling.

Fig. 7. Segmentation: Intensity-based (a)–(d), orientation-based (e)–(h), difference between intensity- and orientation-based (i)–(l) and mean-shift (m)–(p). (for (e)–(h) $N_\theta = 8$). (q) shows the original image with the foreground seed pixels in red and the background in blue (shown magnified by a factor of 3). In (r) and (s) we show the color and orientation labeling produced with our approach for $\lambda = .075$, $N_\theta = 32$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

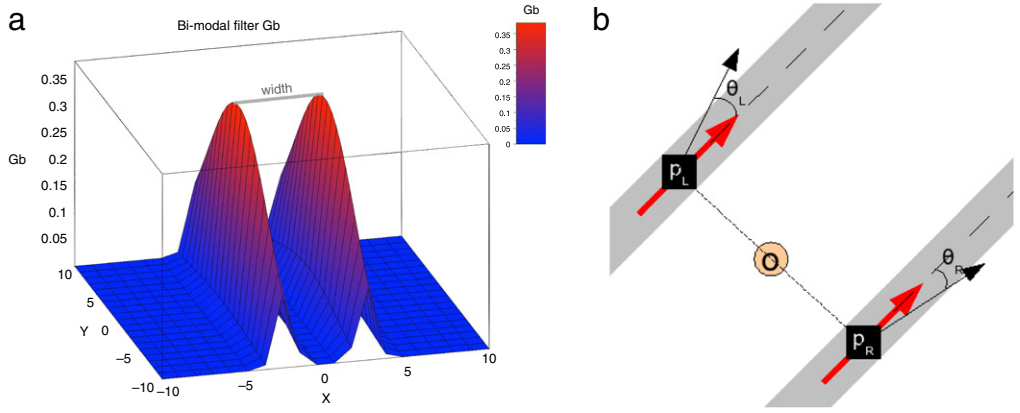
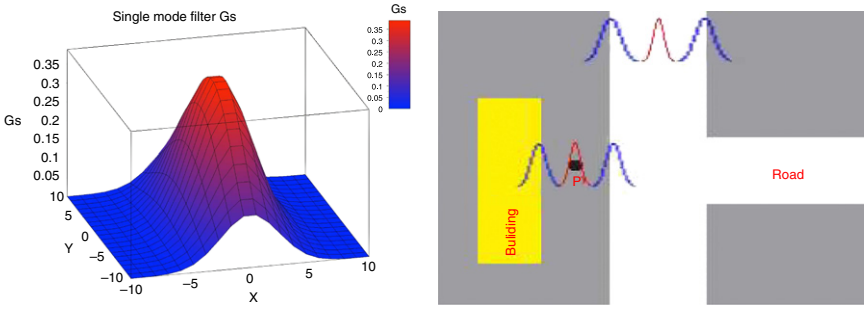


Fig. 8. (a) The bi-modal filter G_b is applied to the classified curve features. (b) Red arrows: filter orientation (at peaks). Black arrows: actual pixel orientation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Single mode filter.

(b) An example of the use of the single mode (red) and bi-modal filter (blue).

Fig. 9. The single mode filter is applied on the binary segmented image. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

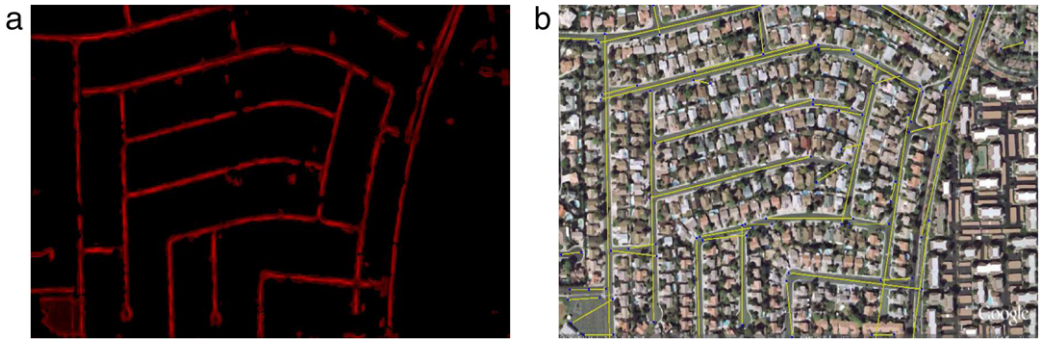


Fig. 10. (a) The response magnitude map computed by the filters is used for the voting of Hough transform. (b) The majority of the centerlines are successfully extracted automatically.

peaks, orientation information is used to weigh the response. This ensures that the maximum response only occurs when both pixels have the same orientation and are aligned to the filter's orientation. Fig. 8(b) demonstrates the application of a bi-modal filter to a point O. The orientations θ_L and θ_R of the left and right road-side points p_L and p_R respectively are used to scale the response. Thus, Eq. (17) becomes,

$$G_b = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}}\left[\cos(\theta_L)\mathrm{e}^{-\left[\frac{(x-\frac{w}{2})_r^2}{\sigma_x^2}+\frac{y_r^2}{\sigma_y^2}\right]} + \cos(\theta_R)\mathrm{e}^{-\left[\frac{(x+\frac{w}{2})_r^2}{\sigma_x^2}+\frac{y_r^2}{\sigma_y^2}\right]}\right]. \quad (20)$$

In addition to the bi-modal filters, single mode gaussian *road-area* filters are applied to the segmented binary image containing the road candidates. This ensures that the area between any parallel lines is indeed a part of the road and therefore should appear in the result of the segmentation. An example that demonstrates the commonly occurring problem handled by the single mode filter is shown in Fig. 9(b) where the edge of the building and the road-side will cause a high response to the bi-modal filter at location p. However, the single mode filter will result in zero response because the binary segmented image will not include the area between the edges.

Road-area and road-side filters of different widths and orientations are combined as $G_t = G_b * G_s$ and are used for the extraction of centerline information. A point along the centerline of a road of orientation θ_R and width w_R, will have a maximum response to a filter with the same or similar orientation and width. Thus, for each pixel we record the filter parameters (orientation, width) for which it returns a maximum response.

Finally, the centerline response magnitudes are used as votes in an iterative Hough transform. The iterative implementation of the Hough transform has the significant advantage that no input parameters are required for the Hough transform, such as number of peaks, minimum vote thresholds, etc. therefore making the linearization process entirely automatic. The result is a set of lines representing the segments of the road network as shown in the example of Fig. 10. The majority of the centerlines are correctly extracted automatically (>80%). However, some false positives still exist due to the global nature of Hough transform.

6.2. Road tracking

Using the automatically extracted width and orientation information computed by the road-area and road-side filters, a tracking algorithm converts the linear segments into their equivalent polygonal representations i.e. road segments. Two road-side points are introduced for each point on a centerline. The spatial location of the road-side points is determined by the road width and the road orientation given by,

$$P_r = P_c + \begin{bmatrix} \frac{w\sin(\theta)}{2} \\ \frac{-w\cos(\theta)}{2} \end{bmatrix} \qquad P_l = P_c + \begin{bmatrix} \frac{-w\sin(\theta)}{2} \\ \frac{w\cos(\theta)}{2} \end{bmatrix} \quad (21)$$

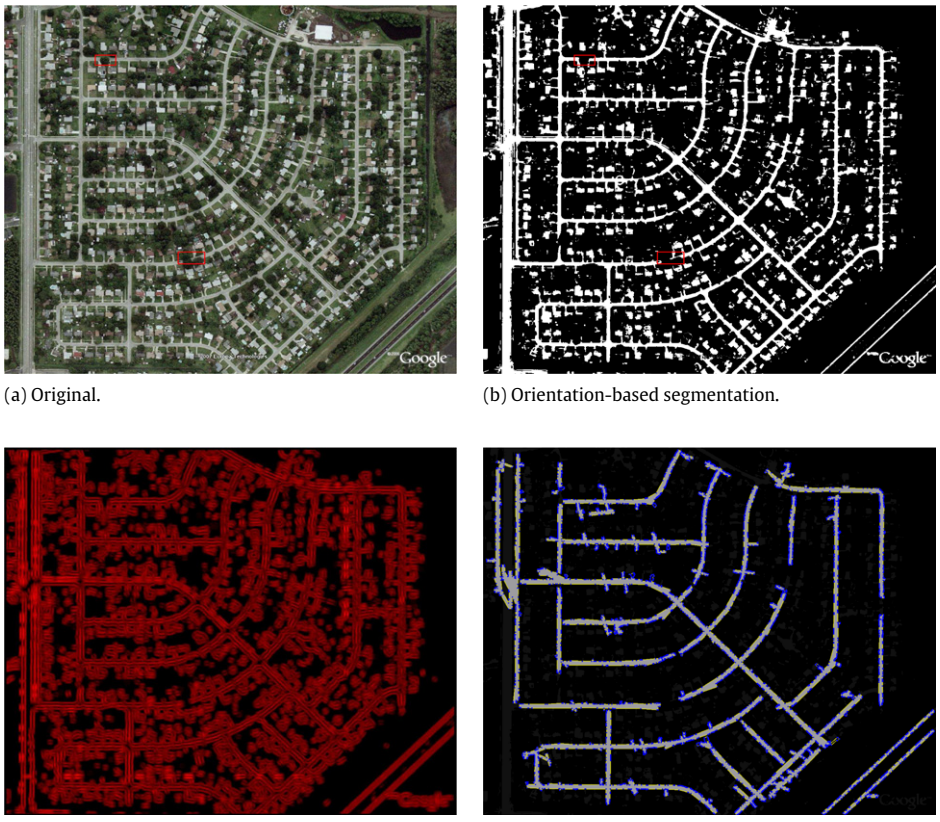
where $P_c = \begin{bmatrix} x \\ y \end{bmatrix}$ the 2D coordinates of the centerline point, w is the width of the road and θ is the orientation of the road. Fig. 11 shows an example of the road tracking. The road-sides are created using Eqs. (21) and are shown as green lines.

In some cases where the road network is particularly complex, the automatically extracted linear segments may contain false positives and false negatives. This is due to the global nature of the Hough transform and the difficulty to correctly handle areas of high curvature such as curvy roads, and areas with no particular

(a) The road centerlines are shown as yellow lines. The points of the centerline vector are shown in blue.

(b) The road segments. The created road-side lines are shown in green and their points are shown in blue.

Fig. 11. Road tracking. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Original.

(b) Orientation-based segmentation.

(c) Centerline extraction.

(d) Extracted road network.

Fig. 12. High resolution satellite image of an urban site with no additional information. In (a), (b) the marked areas show low contrast due to occlusions and shadows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

orientation such as parking lots. For such cases, we employ an interactive approach for further refinement which can have the form of several actions outlined below:

(1) *Adding a seed point.* Once a seed point is added the road-area and road-side filters are applied to derive the width and orientation information. A local neighborhood search is then performed which finds a candidate pixel that minimizes the function,

$$f(x,y) = \text{argmin}(w_d * D_{(x,y)} + w_\theta * O_{(x,y)} + w_w * (W_{(x,y)})) \quad (22)$$

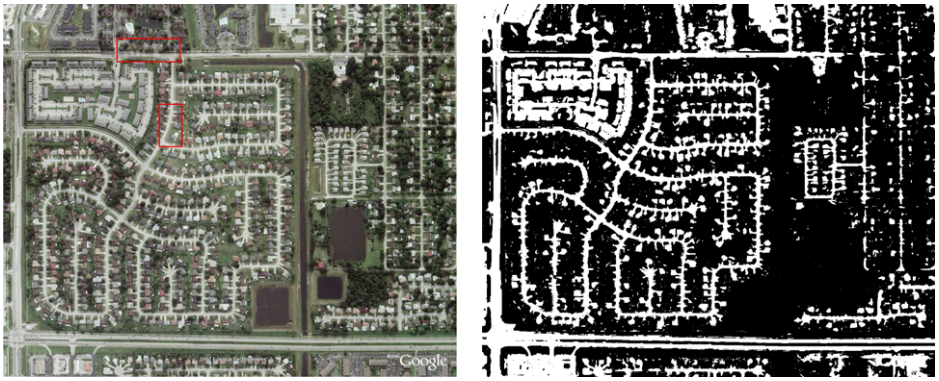
where $D_{(x,y)}$ is the euclidian distance between the candidate and the seed point, $O_{(x,y)}$ is the orientation difference, $W_{(x,y)}$ is the width difference and w_d, w_θ, w_s are weights corresponding to each term, respectively. This process is recursively repeated and each candidate point which minimizes $f(x,y)$ is added to the current line until no more neighboring points are found. The weights used for the examples were experimentally derived as follows: $w_\theta = 0.4$, $w_d = 0.3$, $w_s = w_m = 0.3$.

(2) *Adding or editing a centerline.* Once a centerline is added the filters are applied at a fixed orientation aligned to the specified centerline's slope.

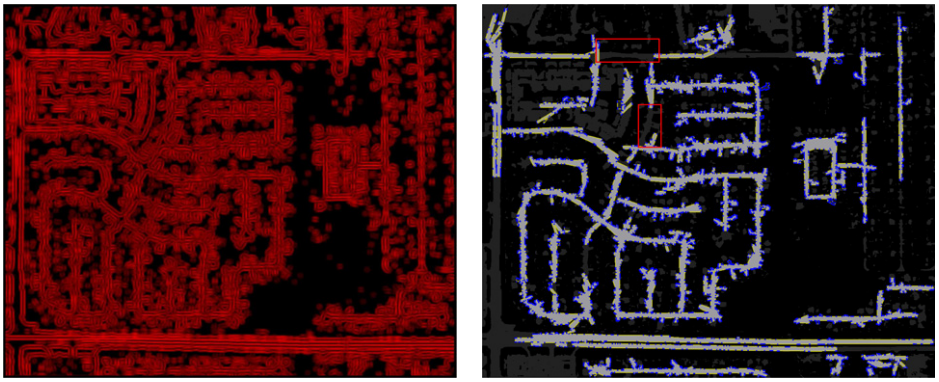
(3) *Merging of two centerlines.* Given two centerlines a Hermite interpolation is performed which generates a cubic polynomial between the most appropriate endpoints of the centerlines. We initialize the Hermite formulation using the spatial location of the endpoints and the precomputed orientations (by tensor voting). This results in a smooth cubic curve $P(u)$ containing a fixed set of points parameterized by $0 \leq u \leq 1$ and is given by,

$$P(u) = U^T M B \quad (23)$$

$$U^T = [u^3 u^2 u^1 1] \quad (24)$$



(a) Original. (b) Orientation-based segmentation.



(c) Centerline extraction. (d) Extracted road network.

Fig. 13. High resolution satellite image of an urban site with no additional information. In (a), (b) the marked areas show low contrast due to occlusions and shadows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

$$B = \begin{bmatrix} P_i \\ P_{i+1} \\ O_i \\ O_{i+1} \end{bmatrix} \tag{26}$$

where P_i, P_{i+1} are the two endpoints of the centerlines, O_i, O_{i+1} are their orientations computed by the tensor voting and $0 \leq u \leq 1$ is a parameterization which controls the interpolated points' locations. The result of the merging is a single centerline consisting of the initial two centerlines and the smooth interpolated curve connecting them.

(4) *Deleting a centerline.* A deleted centerline is removed from the set of linear segments however, the underlying width and orientation information is not altered.

(5) *Smoothing.* The centerline vector is converted to a set of dense points. A "snake" is then used to refine the spatial position of those points using the centerline magnitude map (Fig. 10(a)) as an external force. The energy function being minimized is a weighted combination of internal and external forces defining the contour and is given by,

$$E_{total} = \int_0^1 [a(s) * E_{elasticity}(v(s))\mathrm{d}s + b(s) * E_{stiffness}(v(s)) + c(s) * E_{image}(v(s))]\mathrm{d}s \tag{27}$$

where a, b, c are weights, $v(s) = (x(s), y(s))$ is the parametric form of the snake and $x(s)$, $y(s)$ are x, y coordinates along the snake with $0 \leq s \leq 1$. The elasticity term controls whether the snake can be second-order discontinuous at point $v(s)$ and develop a corner. The stiffness term controls the spacing between the points on the snake. The last term, attracts the snake to image features such as edges. In our implementation the elasticity weight a is set to the magnitude in the curve map of each point. Thus, if a point was classified as a curve with a high saliency, it becomes very difficult for the snake to develop a corner/junction at that point. The stiffness weight remains the same for all points of the snake and is $b = 0.5$. The weight for the last term is set as $c = 0.7$.

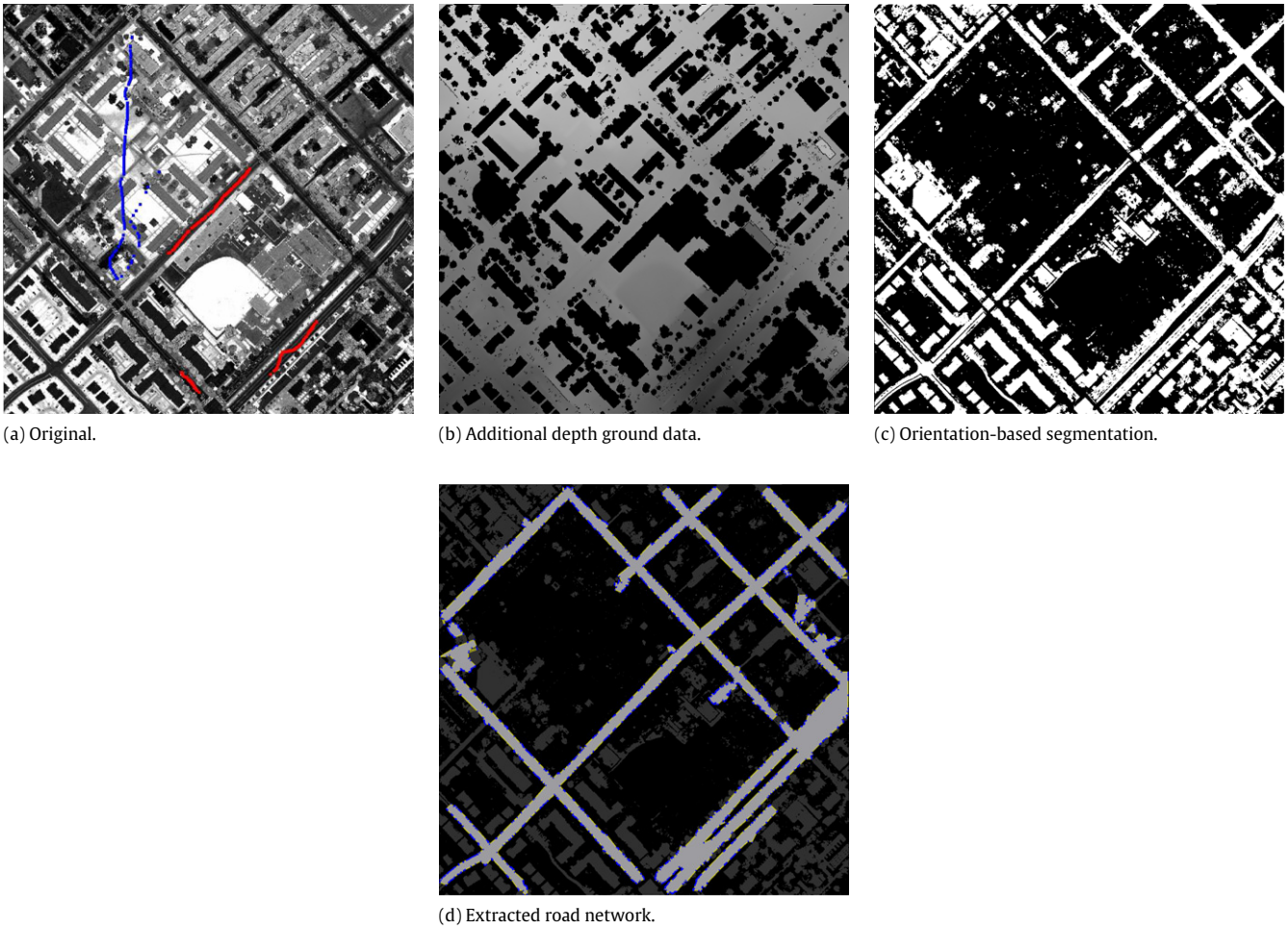
(7) *Approximation/Point reduction.* A centerline consisting of dense set of points is approximated using Iterative End-Point Fit thus reducing the number of points. A user-defined parameter controls the maximum allowed approximation error.

Using the precomputed width and orientation information the centerlines are converted into road segments which consist of a centerline vector, a left road-side and a right road-side separated by the width of the road. Finally, a set of polygonal boolean operations is applied to the road segments. This results in a polygonal representation of the entire road network which allows for the efficient and correct handling of overlaps due to junctions, roundabouts, etc.

7. Experimental results

Experimental results are presented which confirm the validity of the proposed system. In our experiments, the smoothness term is set to $\lambda = 0.25$ and the number of orientation labels to $N_\theta = 8$ unless otherwise specified.

Fig. 12(a) shows a high resolution satellite image of an urban site at a resolution of 1.32 K × 1.12 K. The orientation-based



(a) Original.

(b) Additional depth ground data.

(c) Orientation-based segmentation.

(d) Extracted road network.

Fig. 14. Airborne LiDAR scanner data. (a) shows the initial points selected by the user for the segmentation. Blue points are for background and red points are for foreground objects. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

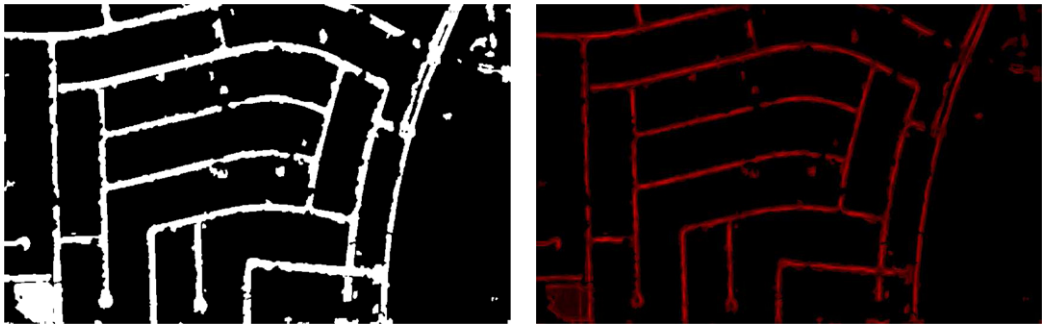
segmentation result is shown Fig. 12(b) where the foreground objects appear in white and the background objects appear in black. The red rectangles marked in Fig. 12(a) and (b) show areas of the network with low intensity variations due to shadows and occlusions, which were not included in the segmentation result. Fig. 12(c) shows the centerlines extracted after applying the bimodal and single mode filters. The automatically generated result is the road network vector data shown in Fig. 12(d). When in automatic mode, we assume the topological characteristic of roads that they are interconnected segments and belong to a larger network, and thus keep only the longest detected road network as the final result. This becomes obvious by the missing segment of the vertical highway on the bottom left of the image. Although it appears in the segmentation result as a road candidate, it was not included in the final network since no connection was found.

Fig. 13(a) shows another example of a satellite image from an urban site. Although the segmentation in Fig. 13(b) successfully classified the road and non-road pixels, the tracking algorithm failed to connect some parts due to big variations to the road width. The red rectangles marked in Fig. 13(a) and (d) indicate such areas.

An intensity-return (IR) image is shown in Fig. 14(a) as captured by an airborne LiDAR scanner at a resolution of 1 K $\times$ 1 K. In this case, additional depth ground data was used as shown in Fig. 14(b). The result of the orientation-based segmentation is shown in Fig. 14(c). Fig. 14(d) shows the resulting vector data for the extracted road network. User interaction was limited to a selection of a few points for the segmentation at the beginning of the execution shown in Fig. 14(a) in blue (background) and red (foreground). Using additional depth data improves the overall performance as in this case since it helps eliminate false positives areas such as parking lots and buildings. In conclusion, the entire road network was successfully recovered.

Fig. 15 shows an interactive result. Using a few user marked road and non-road pixels the process starts by segmenting the original image as in Fig. 15(a). The centerline information is then automatically extracted as shown in Fig. 15(b), and is used to automatically form linear centerline segments as in Fig. 15(c). The user can add, adjust, connect or remove centerline segments, resulting in a refined set of centerline components as in Fig. 15(d). Each time a change occurs the centerline information in Fig. 15(b), (i.e. magnitude, width and orientation) is recomputed automatically to reflect the current set of centerline components. Using the width and orientation information the road segments can then be tracked automatically as shown in Fig. 15(e). Finally, Fig. 15(f) shows the result of combining the road segments together and converting them to their equivalent polygonal representations. Note the correct handling of the junctions.

Fig. 16 shows the final extracted road network using an airborne LiDAR image of an urban area in Baltimore. The complete road network extracted using this approach including the 3D polygonal representations of the roads are shown in Fig. 17. The automatically extracted and interactively refined centerlines are shown as vectors (yellow lines) overlaid on the original image in Fig. 16(a). The road segments which are tracked using the width and orientation information computed by the filters are shown in Fig. 16(b). Fig. 16(c) shows the result of the boolean operations on the



(a) Orientation-based segmentation.

(b) Automatic extraction of centerline information (Only the magnitude is shown).

(c) Connected centerline components (automatic).

(d) Refined connected centerline components (interactive).

(e) Tracked road segments.

(f) Vector data.

Fig. 15. Interactive mode. (a) The result of the segmentation. (b) The centerline information extracted automatically by the parallel-line detector. (c) The centerline components recovered automatically. (d) The user refined centerline components. (e) The tracked road segments automatically computed using (d). (f) The final road map vector data.

polygonal representation of the road segments. As it can be seen overlapping areas e.g. at junctions are handled efficiently and correctly and produce nicely looking intersections.

8. Evaluation

The evaluation of the extracted road networks is performed using the evaluation framework introduced in Wiedemann and Hinz (1999), in terms of the completeness, correctness and quality.

- Completeness. The completeness is defined as the ratio of the true positives from the sum of the true positives and false negatives given by,

$$E_{Completeness} = \frac{TruePositives}{TruePositives + FalseNegatives}. \quad (28)$$

The optimal value of the $E_{Completeness}$ is 1, in which case 100% of the roads are recovered.

- Correctness. The correctness is defined as the ratio of the true positives from the sum of the true and false positives given by,

$$E_{Correctness} = \frac{TruePositives}{TruePositives + FalsePositives}. \quad (29)$$

The optimal value of the $E_{Correctness}$ is 1, in which case 100% of the extracted roads are actual roads.

- Quality. The quality is a measure of the "goodness" of the final result and is given by,

$$E_{Quality} = \frac{TruePositives}{TruePositives + FalsePositives + FalseNegatives}. \quad (30)$$

The optimal value for the $E_{Quality}$ is 1, in which case 100% of the extracted roads are correct and complete.

The above parameters i.e. true positives, false positives and false negatives are determined based on existing geospatial databases. In cases where no additional geospatial information is available, the operator indicates the required parameters.

As it is evident from the results and the metrics of Table 2 the proposed approach performs well for rural as well as urban areas. The success of our approach depends primarily on the performance of the low-level grouping and mid-level segmentation processes. The grouping and segmentation are the two essential components that can drastically affect the outcome. The use of tensor voting framework significantly improves the grouping results since it

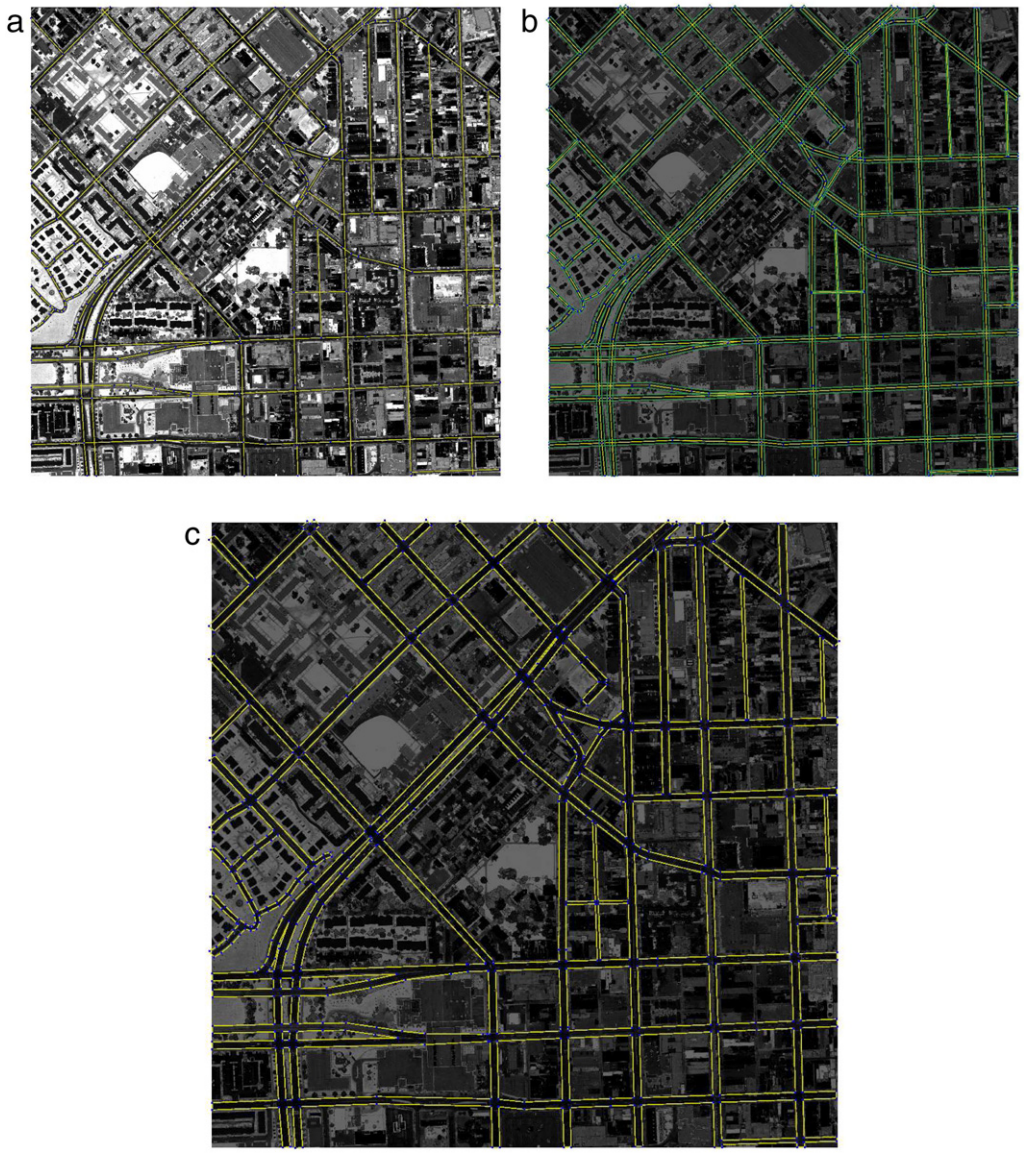


Fig. 16. The result of a 2 K × 2 K urban area. (a) Centerline vectors overlaid on original image. (b) Tracked road segments using the automatically extracted width and orientation. Note the overlap at junctions. (c) Road network using polygonal representation. The overlaps are correctly handled by the boolean operations to form properly looking intersections/junctions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Evaluation of extracted road networks. The images for Oregon, Kentucky and Las Vegas were taken from Google (2009).

Examples	Evaluation measures		
	Completeness (%)	Correctness (%)	Quality (%)
Oregon (12(d))	82	80	68.6
Kentucky (13(d))	86	75.5	67.3
Baltimore (14(d))	83	66	58.8
Las Vegas (15(c))	71.4	80	60.6

eliminates any "guessing" (by using thresholds) when making decisions about neighboring features. However, problems may still arise in cases where occlusion is severe, for example in densely built areas with high buildings like Manhattan, NY or in areas where trees occlude large parts of the roads. In this cases, the use of additional information such as LiDAR data is almost necessary to be able to generate useful results. Examples of these cases were shown in Figs. 12 and 13.

Similarly, the use of a specialized segmentation process considerably improves the segmented road pixels. However, the segmentation process is based on the assumption that the Gaussian distributions of the foreground and background pixels (i.e. road and non-road pixels) are easily separable. Although this is true in most cases with color imagery, there are instances where non-road pixels exhibit the same reflectance properties (i.e. have the same color) as the road pixels. The example shown in Fig. 14 demonstrates this problem. In this example, an intensity-return (IR) image and a depth image of the same area were used for the road extraction. The segmentation of the IR image produces many false positives due to the fact that many rooftops are paved using the same material as the roads, hence their response will be similar to that of the true road pixels. The use of the additional LiDAR elevation information however, resolves this problem.

9. Conclusion

We have presented a framework for the automatic and reliable detection and extraction of complex transportation networks from remote sensor data. Uniquely, our framework is an integrated solution that merges the strengths of perceptual grouping theory (Gabor filters, tensor voting) and segmentation (global optimization

(a) A 10 K × 4 K area covering Baltimore and surrounding areas.

(b) The 3D polygonal representations. The elevation information from the LiDAR was used to model the roads correctly and handle special cases such as bridges.

Fig. 17. The complete road network for downtown Baltimore and surrounding areas.

by graph-cuts), under a unified framework to address the challenging problem of automated feature detection, classification and extraction.

Firstly, we leveraged the local precision and the multi-scale, multi-orientation capability of Gabor filters, combined with the global context of the tensor voting for the extraction and accurate classification of geospatial features. In addition, a tensorial representation was employed for the encoding which removed any data dependencies by eliminating the need for hard thresholds. Moreover, the integration of the Gabor filtering and tensor voting

removes the limitation of tensor voting to only work with binary images, and extends its application to grayscale images.

Secondly, we have presented a novel orientation-based segmentation using graph-cuts for segmenting road features. A major advantage of this segmentation is that it incorporates the orientation information of the classified curve features to produce segmentations with better defined boundaries.

Finally, a set of gaussian-based filters were developed for the automatic detection of road centerlines and the extraction of width and orientation information. The linearized centerlines were finally tracked into road segments and then converted to their polygonal representations.

Extensive tests have shown that the proposed system performs well for all datatypes and scenes, and has consistently achieved a minimum success rate of an average of 69.3%. However, our experiments have shown that the use of LiDAR data significantly improves the road feature segmentation and labeling due to the known elevation of each point and therefore results in better segmented roads. Satellite imagery on the other hand, seems to perform poorly in cases where the color distributions of the background and foreground objects (roads) were very similar thus hard to discriminate. A particularly difficult example which demonstrates this is the test case shown in Fig. 7, where the road network and the background (dirt) in the grayscale image have similar color distributions. A possible method to overcome this problem would be to incorporate prior knowledge about the road reflectance properties by training the system and to additionally compute the color distributions based on the spatial location of the points (i.e. locally and not globally as it is currently the case), which is a direction we are currently exploring.

Appendix. Graph-cut

In Boykov et al. (1999, 2001) the authors interpret image segmentation as a graph partition problem. Given an input image I, an undirected graph $G = \langle V, E \rangle$ is created where each vertex $v_i \in V$ corresponds to a pixel $p_i \in I$ and each undirected edge $e_{i,j} \in E$ represents a link between neighboring pixels $p_i, p_j \in I$. In addition, two distinguished vertices called *terminals* V_s, V_t, are added to the graph G. An additional edge is also created connecting every pixel $p_i \in I$ and the two *terminal* vertices, e_{i,V_s} and e_{i,V_t}. For weighted graphs, every edge $e \in E$ has an associated weight w_e.

A *cut* $C \subset E$ is a partition of the vertices V of the graph G into two disjoint sets S, T where $V_s \in S$ and $V_t \in T$. The cost of each cut C is the sum of the weighted edges $e \in C$ and is given by

$$|C| = \sum_{\forall e \in C} w_e. \tag{A.1}$$

The minimum cut problem can then be defined as finding the cut with the minimum cost. An algorithm for solving this problem has been proven to require polynomial time (Boykov et al., 1999).

Energy minimization function

Finding the minimum cut of a graph is equivalent to finding an optimal labeling $f : I \longrightarrow L$ which assigns a label $l \in L$ to each pixel $p \in I$, and f is piecewise smooth and consistent with the original data. The energy function is then given by,

$$E(f) = E_{data}(f) + \lambda * E_{smooth}(f) \tag{A.2}$$

where λ is the weight of the smoothness term.

Energy data term

The data term in Eq. (A.2) measures the cost of re-labeling the original data with a new labeling f. It is defined us the sum of the per-pixel measure (D_p) of how appropriate each label $f_p \longrightarrow l \in L$ is, for each pixel $p \in I$ in the original data and is given by,

$$E_{data}(f) = \sum_{p \in I} D_p(f_p). \tag{A.3}$$

Energy smoothness term

The smoothness term in Eq. (A.2) measures the cost of re-labeling neighboring pixels with a new labeling f. It is defined as the sum of the differences between two neighboring pixels $p, q \in I$ under a labeling $f_p \longrightarrow l_p \in L$ and $f_q \longrightarrow l_q \in L$ respectively and is given by,

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) \tag{A.4}$$

where N is the set of neighboring pixels and $V_{\{p,q\}}$ measures the difference between the neighboring pixels, also known as the interaction potential function.

References

Bacher, U., Mayer, H., 2005. Automatic road extraction from multispectral high resolution satellite images. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W24), 29–34.

Barsi, A., Heipke, C., 2003. Artificial neural networks for the detection of road junctions in aerial images. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (Part 3/W8), 17–19.

Baumgartner, A., Steger, C.T., Mayer, H., Eckstein, W., Ebner, H., 1999. Automatic road extraction in rural areas. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (Part 3), 107–112.

Boykov, Y., Jolly, M.-P., (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: International Conference on Computer Vision. pp. 105–112.

Boykov, Y., Veksler, O., Zabih, R., (1999). Fast approximate energy minimization via graph cuts. In: International Conference on Computer Vision. pp. 377–384.

Clode, S., Rottensteiner, F., Kootsookos, P., 2005. Improving city model determination by using road detection from lidar data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W24), 159–164.

Google, (2009). Google earth. http://earth.google.com (Accessed October 1, 2009).

Guy, G., Medioni, G.G., 1997. Inference of surfaces, 3D curves, and junctions from sparse, noisy, 3D data. IEEE Transactions on Pattern Analysis Machine Intelligence 19, 1265–1277.

Heller, J., Pakzad, K., 2005. Scale-dependent adaptation of object models for road extraction. In: Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation. pp. 23–28.

Heuwold, J., (2006). Verification of a methodology for the automatic scale-dependent adaptation of object models. In: Photogrammetric Computer Vision. p. (on CD-ROM).

Hinz, S., Baumgartner, A., 2003. Automatic extraction of urban road networks from multi-view aerial imagery. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 58 (Part 1), 83–98.

Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., Baumgartner, A., 2000. Automatic extraction of roads from aerial images based on scale space and snakes. Machine Vision Applications 12 (1), 23–31.

Lisini, G., Tison, C., Cherifi, D., Tupin, F., Gamba, P., (2004). Improving road network extraction in high-resolution sar images by data fusion. In: Committee on Earth Observation Satellites Synthetic Aperture Radar Workshop. p. (on CD-ROM).

Manjunath, B.S., Ma, W.-Y., 1996. Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8), 837–842.

Mayer, H., Hinz, S., Bacher, U., Baltsavias, E., 2006. A test of automatic road extraction approaches. In: Photogrammetric Computer Vision, p. (on CD-ROM).

Mayer, H., Steger, C.T., 1998. Scale-space events and their link to abstraction for road extraction. ISPRS Journal of Photogrammetry and Remote Sensing 53 (Part 2), 62–75.

Medioni, G., Lee, M.S., Tang, C.K., 2000. A Computational Framework for Segmentation and Grouping. Elsevier.

Mena, J.B., Malpica, J.A., 2005. An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. Pattern Recognition Letters 26 (9), 1201–1220.

Peteri, R., Celle, J., Ranchin, T., 2003. Detection and extraction of road networks from high resolution satellite mages. In: International Conference on Image Processing. pp. 301–304.

Porikli, F., 2003. Road extraction by point-wise gaussian models. SPIE Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery IX 5093, 758–764.

Rissanen, J., 1983. A universal prior for integers and estimation by minimum description length. Annals of Statistics 11 (2), 416–431.

Rother, C., Kolmogorov, V., Blake, A., 2004. "Grabcut": Interactive foreground

extraction using iterated graph cuts. ACM Transactions on Graphics 23 (3), 309–314.
Tupin, F., Houshmand, B., Datcu, M., 2002. Road detection in dense urban areas using SAR imagery and the usefulness of multiple views. IEEE Transactions on Geoscience and Remote Sensing 40, 2405–2414.
Wessel, B., 2004. Road network extraction from sar imagery supported by context information. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34 (Part 3B), 360–365.
Wiedemann, C., Hinz, S., 1999. Automatic extraction and evaluation of road networks from satellite imagery. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (Part 3/2W5), 95–100.
Zhang, C., Baltsavias, E., Gruen, A., 2001. Knowledge-based image analysis for 3d road construction. Asian Journal of Geoinformatic 1 (4), 3–14.
Zhang, Q., Couloigner, I., (2006). Automated road network extraction from high resolution multi-spectral imagery. In: Proceedings of ASPRS Annual Conference. p. (on CD-ROM).
Zhou, J., Cheng, L., Bischof, W.F., 2007. Online learning with novelty detection in human-guided road tracking. IEEE Transactions on Geoscience and Remote Sensing 45, 3967–3977.
Ziems, M., Gerke, M., Heipke, C., Automatic road extraction from remote sensing imagery incorporating prior information and colour segmentation. In: Photogrammetric Image Analysis. p. 141.