A Vision-Based System For Automatic Detection and Extraction Of Road Networks

Charalambos Poullis, Suya You, Ulrich Neumann
CGIT/IMSC, USC
Los Angeles, CA 90089
charalambos@poullis.org, {suyay|uneumann}@graphics.usc.edu

Abstract

In this paper 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 presicion 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 then 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 development of a vision-based road detection and extraction system for the accurate and reliable delineation 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.

Firstly, local orientation information is extracted using a bank of Gabor filters, which is encoded into a tensorial representation. This representation can simultaneously capture the geometrical 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.

Secondly, 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, 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.

In summary, our system combines the strengths of the proposed techniques to resolve the challenging problem of extracting complex road networks. We leverage 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.

We have extensively tested the performance of the proposed system with a wide range of remote sensing data including aerial photographs, satellite images, and LiDAR and present our results.

2. Related Work

Different methodologies have been proposed and developed so far and can be categorized as follows:

2.1. Pixel-based

In [2] lines are extracted in an image with reduced resolution as well as roadside edges in the original high resolution image. Similarly, [10] uses a line detector to extract lines from multiple scales of the original data. [9] applies the edge detector on multi-resolution images and uses the result as input to the higher-level processing phase. [13] applies Steger's differential geometry approach for the line extraction. In [1] they use a Deriche operator for the edge detection with an added hysteresis threshold, followed by an edge smoothing using the Ramer algorithm.

In [9] 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). [5] presents a technique where a directional adaptive filter is used for the detection of pixels with particular orientation. Similarly, [12] achieves excellent results by using a gaussian model based approach. In order to extract the road magnitude and orientation for each point, they use a quadruple orthogonal line filter set.

2.2. Region-based

In [15] they use predefined membership functions for road surfaces as a measure for the image segmentation and clustering. Likewise, in [4] they use the reflectance properties, from the ALS data and perform a region growing algorithm to detect the roads. [8] uses a hierarchical network to classify and segment the objects. A slightly different approach is proposed in [10] where a line detector and a classification algorithm are applied on multiple scales of the original data and the results are then merged.

2.3. Knowledge-based

In [13], human input is used to guide a system in the extraction of context objects and regions with associated confidence measures. The system in [14] 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. [4] 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, [15] uses reflectance as a measure for the image segmentation and clustering. Explicit knowledge about geometric and radiometric properties of roads is used in [13] to construct road segments from the hypotheses of roadsides. In [1] 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. [11] takes high resolution images as input along with prior knowledge about the roads e.g. road models and road properties.

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. Figure 1 visually summarizes our approach.

Firstly, we exploit the characteristic that roads are locally linear with smoothly varying curvature, and leverage the multi-scale, multi-orientation nature of gabor filters to detect geospatial features of different orientations and widths. The geospatial features are encoded into a tensorial representation which has the significant advantage that it can capture multiple types of geometric information therefore eliminating the need for thresholding. The refinement and classification is then performed using tensor voting which takes into account the global context of the extracted geospatial features. In addition, tensor voting can effectively deal with noisy and incomplete data therefore resolving commonly occurring problems due to occlusions and shadows from cars, vegetation and buildings.

Secondly, a novel orientation-based segmentation tech-

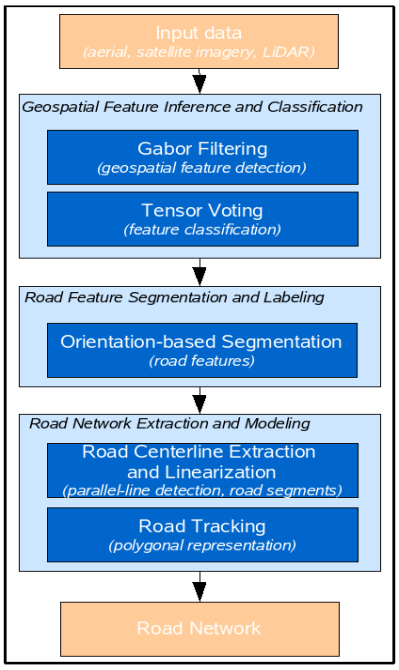


Figure 1. System overview.

nique is proposed for the fast and efficient segmentation of road features. A key advantage of this segmentation is that it incorporates the globally refined geometric information of the classified curve features which results in segmentations with better defined boundaries.

Finally, road centerline information extracted with a pair of single and bi-modal gaussian-based filters is linearized using an iterative Hough transform. This eliminates the need for specifying the number of peaks and other thresholds required by the Hough transform and iteratively extracts all dominant linear segments. These linear segments are then converted into their equivalent polygonal representations using the width information extracted earlier by the filters. Polygonal boolean operations are lastly performed for the correct handling of overlaps at junctions/intersections.

4. Geospatial Feature Inference and Classification

4.1. Gabor Filtering

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. A two dimensional gabor function $g(x, y)$ in space domain is given by

$$g(x,y) = e^{j(2\pi(u_0 x + v_0 y) + \phi)} \times \kappa e^{(-\pi(s_x^2(x-x_0)_\theta^2 + s_y^2(y-y_0)_\theta^2))} \tag{1}$$

where (u_0, v_0) is the spatial frequency, ϕ is the phase of the sinusoidal, κ 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. The remaining parameters in equation 1 are computed as functions of the orientation and frequency parameters as in [6].

The application of the bank of gabor filters results in a total of 40 response images(8 orientations x5 frequencies). 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 the next section 4.2.

4.2. Tensor Voting

Tensor voting is a perceptual grouping and segmentation framework introduced by [7]. 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,

$$T = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \vec{e}_1^T \\ \vec{e}_2^T \\ \vec{e}_3^T \end{bmatrix} \tag{2}$$

$$T = \lambda_1 \vec{e}_1 \vec{e}_1^T + \lambda_2 \vec{e}_2 \vec{e}_2^T + \lambda_3 \vec{e}_3 \vec{e}_3^T \tag{3}$$

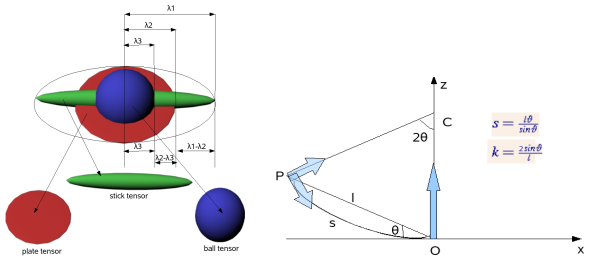
where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are eigenvalues, and $\vec{e}_1$, $\vec{e}_2$, $\vec{e}_3$ are the eigenvectors corresponding to $\lambda_1, \lambda_2, \lambda_3$ respectively. By applying the spectrum theorem, the tensor T in equation 3 can be expressed as a linear combination of three basis tensors(ball, plate and stick) as in equation 4.

$$T = (\lambda_1 - \lambda_2)\vec{e}_1\vec{e}_1^T + (\lambda_2 - \lambda_3)(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T) + \lambda_3(\vec{e}_1\vec{e}_1^T + \vec{e}_2\vec{e}_2^T + \vec{e}_3\vec{e}_3^T) \tag{4}$$

In equation 4, $(\vec{e}_1\vec{e}_1^T)$ 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 geometrical interpretation of tensor decomposition is shown in Figure 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 in the gabor filter response images computed previously is encoded using equation 2 into a unit plate tensor(representing a curve) with the orientation $\vec{e}_3$ aligned to the filter orientation and is scaled by the magnitude of the response of that point. The resulting eight tensors for each point are then added together which produces a single tensor per point capturing the local geometrical information. To summarize, 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.



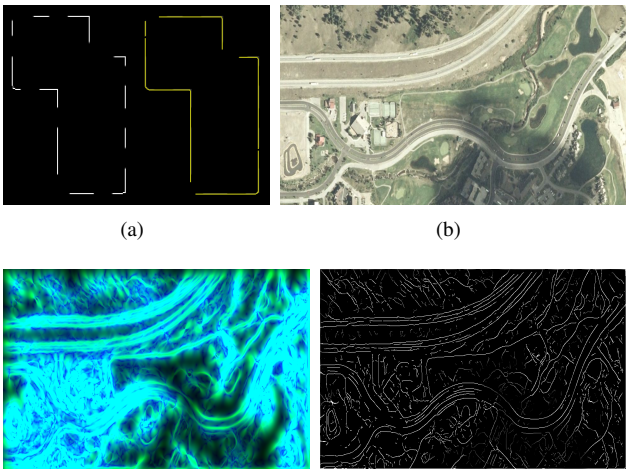
(a) (b)

Figure 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.

The encoded points then cast a vote to their neighbouring 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 [7] given by

$$DF(s, \kappa, \sigma) = e^{-(\frac{s^2 + c\kappa^2}{\sigma^2})} \quad (5)$$

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 neighbourhood size as shown in Figure 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.



(a) (b)

(c) (d)

Figure 3. (a) Successfull 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). and classification using tensor voting. (d) Classified curve features derived from 3(c). Note that no thresholds were used.

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 Figure 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 Figure 3(c). The saliency map is used for the classification of the curve points which are shown in Figure 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 Figure 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. Graph-cut Overview

In [3] the authors interpret image segmentation as a graph partition problem. Given an input image I, an undirected graph $G =< V, E >$ 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 neighbouring 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 $e \in C$. The minimum cut problem can then be defined as finding the cut with the minimum cost which can be achieved in near polynomial-time.

5.2. Labels

The binary case 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 $N_\theta = [2, 16]$ generates acceptable results. Thus the set of labels L is defined to be $L = \{O_{\theta_1}, B_{\theta_1}, O_{\theta_2}, B_{\theta_2} ..., O_{\theta_{N_\theta}}, B_{\theta_{N_\theta}}\}$ with size $|L| = 2 * N_\theta$.

5.3. 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

$$E(f) = E_{data}(f) + \lambda * E_{smooth}(f) \tag{6}$$

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,

$$E_{data}(f) = \sum_{p \in I} D_p(f(p)) \tag{7}$$

As in [3], 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,

$$D_p(f(p)) = \frac{1 - ln(P(I_p|f(p)))}{2 - ||\theta_p - \theta_{f(p)}||^2} \tag{8}$$

The energy data term then becomes,

$$E_{data}(f) = \sum_{p \in I} (\frac{1 - ln(P(I_p|f(p)))}{2 - ||\theta_p - \theta_{f(p)}||^2}) \tag{9}$$

Energy smoothness term. The smoothness term provides a measure of the difference between two neighbouring 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

$$\Delta_{p,q} = \frac{1 + (I_p - I_q)^2}{2 - ||\theta_p - \theta_q||)^2} \tag{10}$$

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 neighbouring pixels p, q under a labeling f as

$$\widehat{\Delta_{p,q}} = \frac{1 + (I_{f(p)} - I_{f(q)})^2}{2 - ||\theta_{f(p)} - \theta_{f(q)}||^2} \tag{11}$$

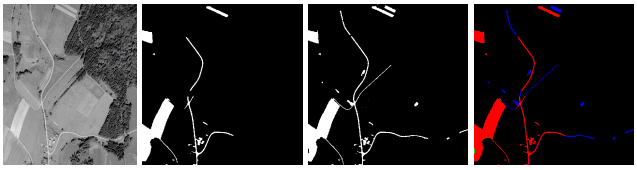
Using the smoothness measure defined for the observed data and the smoothness measure defined for any given labeling we can finally define the energy smoothness term as follows,

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f(p), f(q)) \tag{12}$$

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} K_{p,q} * \widehat{\Delta_{p,q}} \quad (13)$$

where N is the set of neighbouring pixels, $K_{p,q} = [e^{-\frac{\Delta^2_{p,q}}{2*\sigma^2}}]$, and σ controls the smoothness uncertainty. Intuitively, if two neighbouring 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 propability of $\widehat{\Delta_{p,q}}$ being small. To summarize, the function $E(f)$ penalizes heavily for severed edges between neighbouring 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 neighbouring 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 Figure 4.



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

Figure 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)

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,

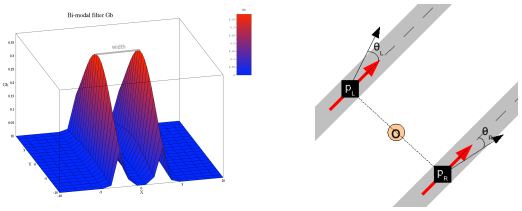
$$G_b = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}}[e^{-[\frac{(x-\frac{w}{2})^2_r}{\sigma^2_x}+\frac{y^2_r}{\sigma^2_y}]} + e^{-[\frac{(x+\frac{w}{2})^2_r}{\sigma^2_x}+\frac{y^2_r}{\sigma^2_y}]}] \quad (14)$$

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

$$(x-\frac{w}{2})_r = (x-\frac{w}{2})cos(\phi) + ysin(\phi) \quad (15)$$

$$y_r = -(x-\frac{w}{2})sin(\phi) + ycos(\phi) \quad (16)$$

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 Figure 5(a).



(a) (b)

Figure 5. (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.

Bi-modal 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 coindicidental presence of two curve pixels along the filters' 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. Figure 5(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, equation 14 becomes,

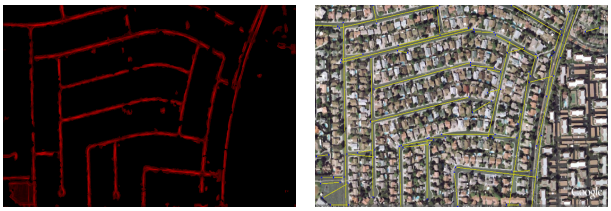
$$G_b = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}}[cos(\theta_L)e^{-[\frac{(x-\frac{w}{2})^2_r}{\sigma^2_x}+\frac{y^2_r}{\sigma^2_y}]} + cos(\theta_R)e^{-[\frac{(x+\frac{w}{2})^2_r}{\sigma^2_x}+\frac{y^2_r}{\sigma^2_y}]}] \quad (17)$$

In addition to the bi-modal filters, single mode gaussian 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.

Single mode and bi-modal 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. This 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 Figure 6. The majority of the centerlines are correctly extracted automatically. However, some false positives still exist.



(a) (b)

Figure 6. (a) The response magnitude map computed by the filters is used for the voting of Hough transform. (b) The majority of centerlines are successfully and automatically extracted.

6.2. Road Tracking

Using the automatically extracted width and orientation information computed by the filters, a tracking algorithm converts the linear segments into their equivalent polygonal representations i.e. road segments. In some cases where the road network is particularly complex, the automatically extracted linear segments may contain false positives and false negatives. For such cases, we employ an interactive approach for the further refinement which can have the form of several actions outlined below,

1. Adding a seed point. Once a seed point is added the filters are applied to derive the width and orientation information. The system then recursively performs a local neighbourhood search to find a candidate pixel that minimizes the function,

$$f(x,y) = argmin(w_d * D_{(x,y)} + w_\theta * O_{(x,y)} + w_w * (W_{(x,y)})) \quad (18)$$

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 neighbouring points are found. The weights used for the examples were defined 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 spline is fit between the most appropriate endpoints resulting in a single merged centerline.

4. Deleting a centerline.

5. Smoothing. The centerline vector is converted to dense points. A snake is then used to refine the spatial position of those points using the centerline magnitude map(Figure 6(a)) as an external force.

6. Approximation/Point reduction. A centerline consisting of dense points is approximated using Iterative End-Point Fit thus reducing the number of points.

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/intersections, round-abouts, etc.

7. Experimental Results

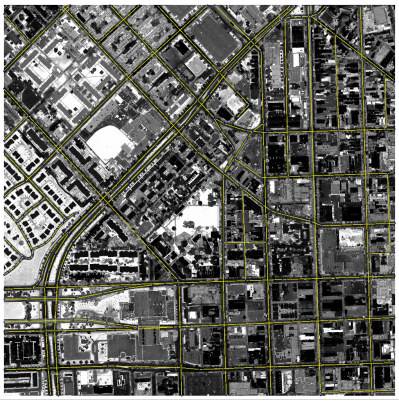
Figure 7 shows the final extracted road network using an airborne LiDAR image of an urban area in Baltimore. The automatically extracted and interactively refined centerlines are shown as vectors(yellow lines) overlaid on the original image in Figure 7(a). The road segments which are tracked using the width and orientation information computed by the filters are shown in Figure 7(b). Figure 7(c) shows the result of the boolean operations on the 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. Conclusion

We have presented a vision-based road detection and extraction system for the accurate and reliable delineation of complex transportation networks from remote sensor data. To our best knowledge, there is no work done in combining the perceptual grouping theories and optimized segmentation techniques for the extraction of road features and road map information. Our system is 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.

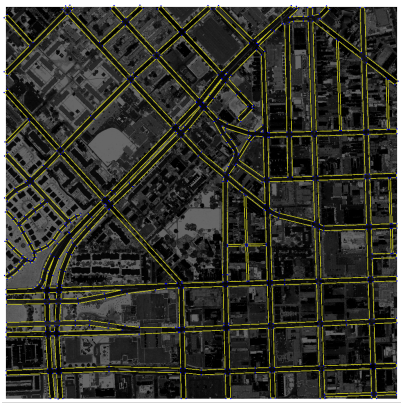
Firstly, we leveraged the local precision and the multi-scale, multi-orientation capability of gabor filters, combined with the global context of the tensor voting for the extraction and accurate classification of geospatial features. In addition, a tensorial representation was employed for the encoding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

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



(a)

(b)



(c)

Figure 7. The result of an 2Kx2K 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.

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.

References

[1] A. Barsi and C. Heipke. Artificial neural networks for the detection of road junctions in aerial images. In *ISPRS Archives, Vol. XXXIV, Part 3/W8, Munich, 17.-19. Sept.*, 2003.

[2] A. Baumgartner, C. T. Steger, H. Mayer, W. Eckstein, and H. Ebner. Automatic road extraction in rural areas. In *ISPRS Congress*, pages 107–112, 1999.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001.

[4] S. Clode, F. Rottensteiner, and P. Kootsookos. Improving city model determination by using road detection from lidar data. In *Inter. Archives of the PRSASI Sciences Vol. XXXVI - 3/W24, pp. 159-164, Vienna, Austria*, 2005.

[5] F. Dell'Acqua, P. Gamba, and G. Lisini. Road extraction aided by adaptive directional filtering and template matching. In *ISPRS Archives, Vol. XXXVI, Part 8/W27, Tempe,AZ, 14.-16. March.*, 2005.

[6] I. R. Fasel, M. S. Bartlett, and J. R. Movellan. A comparison of gabor filter methods for automatic detection of facial landmarks. In *International Conference on Automatic Face and Gesture Recognition*, pages 231–235, 2002.

[7] G. Guy and G. G. Medioni. Inference of surfaces, 3D curves, and junctions from sparse, noisy, 3D data. *IEEE Trans. Pattern Anal. Mach. Intell*, 19(11):1265–1277, 1997.

[8] P. Hofmann. Detecting buildings and roads from ikonos data using additional elevation information. In *Dipl.-Geogr.*

[9] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Mach. Vis. Appl*, 12(1):23–31, 2000.

[10] G. Lisini, C. Tison, D. Cherifi, F. Tupin, and P. Gamba. Improving road network extraction in high-resolution sar images by data fusion. In *CEOS SAR Workshop 2004*, 2004.

[11] R. Peteri, J. Celle, and T. Ranchin. Detection and extraction of road networks from high resolution satellite mages. In *ICIP03*, pages I: 301–304, 2003.

[12] Porikli and F. M. Road extraction by point-wise gaussian models. Technical report, MERL, July 2003. SPIE Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery IX, Vol. 5093, pp. 758-764.

[13] B. Wessel. Road network extraction from sar imagery supported by context information. In *ISPRS Proceedings*, 2004.

[14] C. Zhang, E. Baltsavias, and A. Gruen. Knowledge-based image analysis for 3d road construction. In *Asian Journal of Geoinformatic 1(4)*, 2001.

[15] Q. Zhang and I. Couloigner. Automated road network extraction from high resolution multi-spectral imagery. In *ASPRS Proceedings*, 2006.