UCRL-CONF-218522

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Using Gradients, Alignment and Proximity to Extract Curves and Connect Roads in Overhead Images

Barry Y. Chen, David W. Paglieroni

January 30, 2006

SPIE Defense and Security Symposium: Optics and Photonics in Global Homeland Security
Orlando, FL, United States
April 17, 2006 through April 21, 2006

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Using Gradients, Alignment and Proximity to Extract Curves and Connect Roads in Overhead Images [1]

Barry Y. Chen and David W. Paglieroni [2]

ABSTRACT

A robust approach for automatically extracting roads from overhead images is developed in this paper. The first step involves extracting a very dense set of edge pixels using a technique based on the magnitude and direction of pixel gradients. In step two, the edges are separated into successive channels of edge orientation that each contain edge pixels whose gradient directions lie within a different angular range. A de-cluttered map of edge curve segments is extracted from each channel, and the results are merged into a single composite map of broken edge curves. The final step divides broken curves into segments that are nearly linear and classifies each segment as connected at both ends or disconnected. A measure of connectability between two disconnected line segments based on proximity and relative alignment is defined mathematically. Each disconnected segment is paired with the disconnected segment that it is most connectable to. Pairs of segments are merged if their separation and misalignment are below thresholds (manually specified at present) and the connectability of the pair is two-way optimal. Extended curve and road extraction examples are provided using commercial overhead images.

Keywords: curve extraction, road detection, pixel gradients, edges

1. INTRODUCTION

Although extended curves in overhead images can be attributed to shadows or geological features (such as rivers and edges of land masses, cliffs or forests), they are very often associated with man-made roads. The ability to automatically extract extended curves (especially roads) from overhead images is fundamental and important to computer-aided image understanding. A novel approach for automatically extracting extended curves (such as roads) from overhead images is presented in this paper.

A major objective of image understanding is to be able to accurately represent the content of a scene in terms of relationships between extracted objects and entities at a higher level of abstraction than with pixels. A critical step towards image understanding is thus computer-aided extraction of wide varieties of content from images, including extended curves such as roads, extended regions such as forests, man-made structures such as buildings, moving vehicles such as trucks, etc.

The ability to automatically extract roads from images is useful in itself. In particular, although road networks are available in Geographical Information Systems (GIS's) for some sites, they are unavailable for many others. Even when available, road network data is often incomplete, inaccurate or outdated. The tedious labor intensive process of populating a GIS with road data involves extensive manual inspection of overhead imagery. Sufficiently robust tools for automatically extracting roads from images could dramatically reduce manual inspection time. They could also be used to flag areas of significant change in existing road databases.

Systems such as the Image Content Engine (ICE) cue human analysts for content, such as specific types of buildings or moving vehicles, in broad area search images [1]. Once ICE has matched the necessary patterns or extracted the appropriate features from images to be searched, it provides prioritized cues to human analysts for manual confirmation. It normally takes much less time to manually inspect lists of prioritized cues than to manually search entire images. The relevance of the cues, the prioritization of relevant cues, the false alarm rate, confidence levels and search efficiency can all be dramatically improved by requiring matches to buildings and moving vehicles from ICE to be close to roads from GIS's or roads extracted automatically from the same images.

Over the past several decades many systems have been developed for the automatic extraction of extended curves and roads from overhead images. These systems can be generally categorized into three classes based on the method

---

[1] This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

[2] Lawrence Livermore National Laboratory chen52@llnl.gov, paglieroni1@llnl.gov

they use to identify road pixels from the background: edge linking methods, Hough methods, and pattern matching methods. Edge linking methods model roads as long, slowly curving lines (or parallel lines in high resolution images). Roads are formed by linking large sets of edge pixels whose magnitude and orientations change slowly from one edge pixel to the next. This linking process is often implemented as a search problem where the next pixel to link is determined either by heuristics [2][3], as the next node in a minimal path graph search [4], or by an underlying road path model (see [5] for an example of a deterministic path model and [6] and [7] for stochastic path models).

Related to edge linking methods are Hough transform-based methods [8]. The Hough transform is used to identify edge pixels belonging to the same line, and because some roads are especially well modeled by straight parallel lines, Hough methods can find roads in urban environments (many of which are straight) [9] or initial straight road segments for other road tracking algorithms [10].

Pattern matching methods look for pixels or a group of pixels that exhibit road-like characteristics. For example, [11] identify road pixels by finding pixels whose intensity in MSS band 2 of the LANDSAT-1 images falls within that expected for concrete or asphalt regions. A similar pixel spectra-based classification method is found in [12]. Other pattern methods find groups of pixels that match textures or profiles representative of roads. [5] extends [13]'s approach which identifies road pixels by correlating the intensity cross-section of typical roads to potential road locations in the image. Texture classification techniques, such as those based on Gabor wavelets [14-16], can be tuned to detect image tiles with linear textures of varying orientations and as such can be used to find regions whose textures resemble that of roads.

A novel and robust approach for automatically extracting extended curves (such as roads) from overhead images is presented in this paper. As shown in Fig.1, images are transformed into gradients, gradients are transformed into edges (Section 2), edges are transformed into broken curves (Section 3), broken curves are modeled as broken polylines (Section 4), and aligned broken polylines are connected (Section 5). Several parameters are involved, but there are simple guidelines for choosing most parameter values, and specific values are usually not critical.

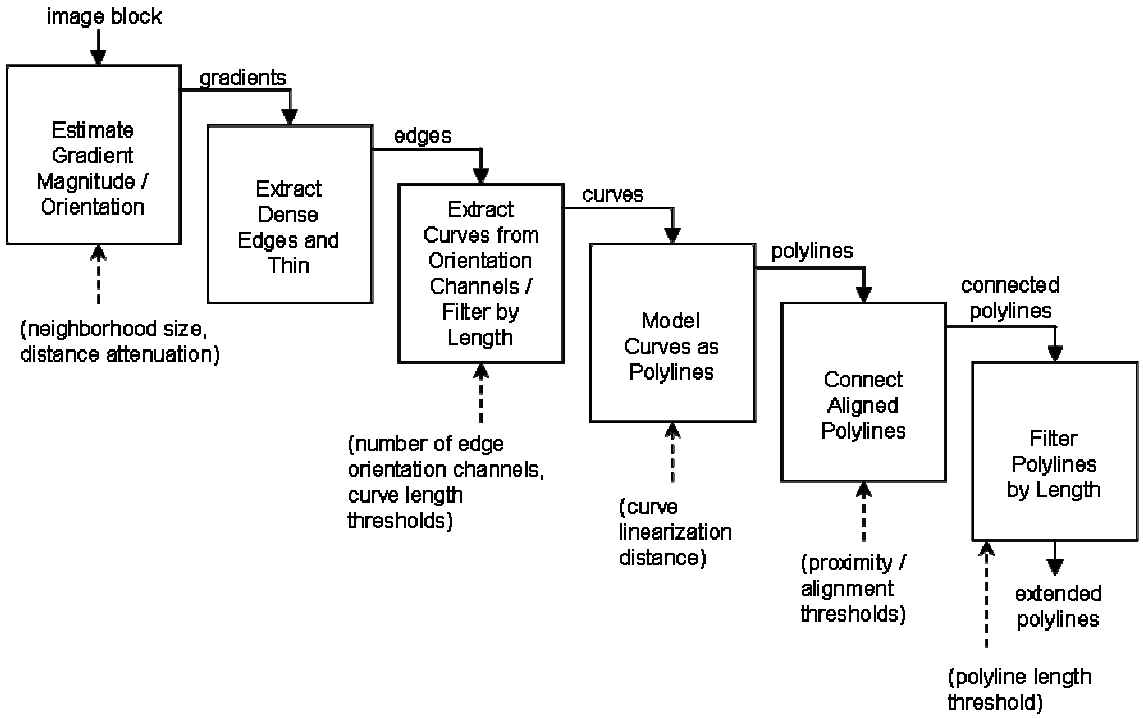


Fig.1 Block diagram for extracting extended polylines from overhead image blocks.

We have discovered that edge curve extraction is facilitated when edges of greatest possible density (derived from gradient magnitudes and directions) are separated into overlapping channels of edge orientation (derived from gradient directions). By de-cluttering each channel, recombining the de-cluttered channels, and independently de-cluttering the recombined edge image, we have observed that extended edge curves can be more effectively extracted while edge

clutter can be more easily suppressed. Like traditional edge linking methods, this approach implicitly models roads as long, slowly curving lines, but unlike many edge linking methods roads are extracted as a result of these separate de-cluttering operations. Finally, we linearly connect disconnected polyline representations of broken edge curves if and only if each of the two open line segments to be joined is the one that is most connectable to the other. Examples that demonstrate these concepts are provided throughout the paper.

2. DENSE EDGES FROM GRADIENTS

Most algorithms for detecting edges in images u seek local maxima in image gradient magnitudes. Gradients are traditionally estimated by applying 1D filters of Gaussian derivatives in the columns and rows direction to u [17][18]. Gradient local maxima occur at points where the magnitude of the gradient image is zero. Edge coarseness is regulated using the Gaussian variance parameter σ.

We have chosen a more geometrical approach to gradient estimation in which the gradient $\dot{u}(c,r)$ of u at a pixel with column and row coordinates $[c,r]$ is estimated as a weighted sum of direction vectors from the center of pixel $[c,r]$ to the centers of neighbor pixels $[c',r']$. The weight assigned to a neighbor $[c',r']$ is determined by its distance from $[c,r]$ and the difference between pixel values at $[c',r']$ and $[c,r]$. $\dot{u}(c,r)$ is complex with magnitude $A(c,r)$ and direction $\theta(c,r)$:

$$\dot{u}(c,r) \stackrel{\Delta}{=} A(c,r)\, e^{j\theta(c,r)} = \sum_{(c',r')\in R(c,r\,|\,\rho)} [\, u(c',r') - u(c,r)\,] \cdot \frac{(c'-c) + j(r'-r)}{[(c'-c)^2 + (r'-r)^2]^{k/2}} \tag{1}$$

where $R(c,r\,|\,\rho)$ is the neighborhood containing all pixels that intersect a disk of radius ρ centered on $[c,r]$. Though defined for monochrome images u, the gradient estimate can be generalized to multi-band images by writing equation (1) for each band, summing the equations into one "composite" equation and dividing both sides by the number of bands. This is tantamount to interpreting $u(c,r)$ in equation (1) as the mean of pixel $[c,r]$ values across all bands.

Equation (1) can be expressed in convolution form as

$$\dot{u}(c,r) = [\, h(c,r\,|\,\rho,k) * 1(c,r)\,] \cdot u(c,r) - [\, h(c,r\,|\,\rho,k) * u(c,r)\,] \tag{2}$$

$$h(c,r\,|\,\rho,k) = \frac{c + jr}{[c^2 + r^2]^{k/2}} \quad \text{for} \quad [c,r] \in R(0,0\,|\,\rho) \tag{3}$$

where "*" is the convolution operator, $1(c,r)$ is an image whose pixels are all ones, and h is a complex filter. Although h is not separable, the FFT can be applied in equation (2) to estimate the gradient image with efficiency (even for large filter masks) comparable to image convolution in the spatial domain with separable 2D filters decomposed into pairs of 1D Gaussian filters.

One can increase coarseness by factoring more neighbor pixels into the gradient estimate. This is achieved by using a larger σ value in the Gaussian formulation. However, because there is only one parameter, the impact of distance-to-neighbor on the gradient estimate cannot be controlled independent of coarseness. This explains why Gaussian edges derived from large σ values have rounded corners, and why one might therefore wish to combine Gaussian edges across successive coarseness resolutions. On the other hand, equation (1) uses two parameters. The neighborhood radius parameter $\rho > 0.5$ regulates coarseness and noise suppression in gradient estimates, while the distance-to-neighbor attenuation parameter $k \geq 0$ independently regulates the impact of distance-to-neighbor on gradient estimates and edge precision. There are combinations of parameters ρ and k that can produce coarse low-noise gradient estimates and more precise edges with relatively sharp corners simultaneously. Combining edges across successive coarseness resolutions is thus much less of an issue when equation (1) is used.

To understand how ρ and k interact to produce various gradient and edge estimation behaviors, consider the direction vector $\underline{d}(c,c',r,r'\,|\,k) \stackrel{\Delta}{=} [(c'-c) + j(r'-r)] / [(c'-c)^2 + (r'-r)^2]^{k/2}$ in equation (1), and assume that $u(c',r') - u(c,r)$ is constant. If $k = 0$, pixels $[c',r']$ farther from $[c,r]$ have a greater impact on $\dot{u}(c,r)$ when they should have less. If $k = 1$, $\underline{d}$ is a unit vector, so the impact of pixels $[c',r']$ on $\dot{u}(c,r)$ does not vary with distance from $[c,r]$. If k

= 2, the impact of pixels $[c',r']$ on $\dot{u}(c,r)$ varies inversely with distance from $[c,r]$, but collectively, the set of all pixels at fixed distance from $[c,r]$ have the same impact on $\dot{u}(c,r)$, independent of distance. If $k = 3$, the set of all pixels at fixed distance from $[c,r]$ collectively have an impact on $\dot{u}(c,r)$ that varies inversely with distance. If $k > 3$, the influence of pixels $[c',r']$ on $\dot{u}(c,r)$ diminishes severely with increasing distance from $[c,r]$. For $k \leq 3$, gradient estimate coarseness increases with ρ. However, as k increases beyond 3, ρ has much less impact on gradient coarseness because the distance attenuation is more severe.

Moreover, when ρ is small and $k < 3$, fine detail edges can be derived from the gradient estimates, and such settings are appropriate when using gradients to detect small objects [19]. In fact, the gradient estimate in equation (1) degenerates to the Prewitt estimate when $(\rho, k) = (1,0)$ [20] and the Sobel estimate when $(\rho, k) = (1,2)$ [22]. It can be shown that for ideal binary edges passing through centers of 3x3 neighborhoods, Sobel gradient direction estimates are exact at roughly every *20°* of edge slope, and the estimates are close in between. Among all estimates in equation (1) for which $\rho = 1$, the Sobel estimate is thus among the most accurate. When ρ is large and k varies from 0 to 1, imprecise noise-suppressed coarse edges with curved corners are produced. When ρ is large and k varies between 1 and 3, more precise noise-suppressed edges with sharper corners are produced. For example, we often use $(\rho, k) = (5,2)$ to extract noise-suppressed extended curves with decent edge precision from overhead images, although the exact values of the parameters are not that important.

Edges are locations of local maxima in the gradient estimate $\dot{u}(c,r)$ for which the gradient magnitude $A(c,r) \geq A_0 \geq 0$. A simple test for gradient local maxima was proposed in [21]. The test makes a distinction between gradient orientation $\theta(c,r) \in [-\pi, \pi]$ in equation (1) and line orientation (slope angle)

$$\theta'(c,r) \stackrel{\Delta}{=} \begin{cases} \theta(c,r) & 0 \leq \theta(c,r) < \pi \\ \theta(c,r) + \pi & \theta(c,r) < 0 \\ 0 & \theta(c,r) = \pi \end{cases} \in [0, \pi) \tag{4}$$

$[c,r]$ is said to be an edge pixel if

$$A(c,r) > max\,[A_0, A(c+\Delta_c, r+\Delta_r), A(c-\Delta_c, r-\Delta_r)] \tag{5a}$$

$$[\Delta_c, \Delta_r] = \begin{cases} [1,0] & \theta'(c,r) < \pi/8 \text{ or } \geq 7\pi/8 \\ [1,1] & \pi/8 \leq \theta'(c,r) < 3\pi/8 \\ [0,1] & 3\pi/8 \leq \theta'(c,r) < 5\pi/8 \\ [1,-1] & 5\pi/8 \leq \theta'(c,r) < 7\pi/8 \end{cases} \tag{5b}$$

The edge map will be as dense as possible when $A_0 = 0$. Edges are not allowed to occur within ρ' pixels of the first and last image row and column, where ρ' is the smallest integer at least as large as ρ.

Boundary following for curve extraction from edges is facilitated by first completely thinning all of the edges. The 8 structuring elements for complete symmetrical thinning are

$$H_0 = \begin{bmatrix} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{bmatrix}, \qquad H_1 = \begin{bmatrix} x & x & 0 \\ 1 & 1 & x \\ x & 1 & x \end{bmatrix} \tag{6}$$

and their 90° rotations, where "x" corresponds to "don't care" (either 0 or 1). On each iteration, the 8 structuring elements are applied successively to each edge pixel. Iterations continue until convergence is achieved.

Fig.2 shows a 256x256 overhead image block containing roads and a 16x16 sub-block (courtesy of DigitalGlobe). Fig.3 shows estimated gradients and fully thinned dense edges ($A_0 = 0$) at various levels of coarseness. As the coarseness of the gradient estimate increases, some detail is lost, but the gradient magnitudes and directions become less noisy, and the edge corners remain relatively sharp.

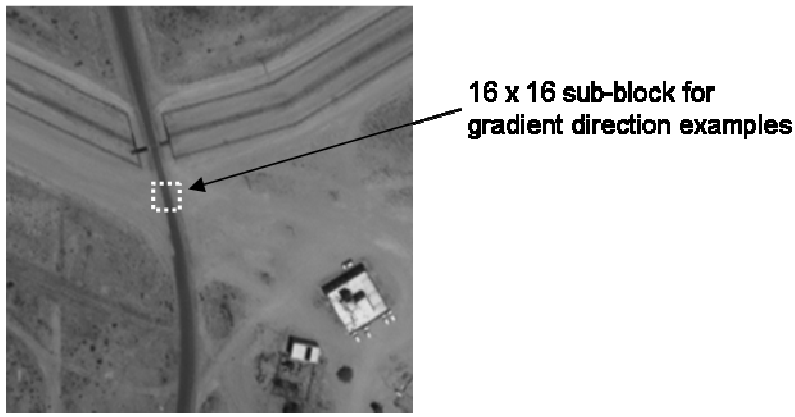


Fig.2 256 x 256 overhead image block (courtesy of DigitalGlobe).

3. BROKEN CURVES FROM EDGE ORIENTATION CHANNEL DECOMPOSITIONS

Following [21], a novel approach to curve extraction based on edge pixel connectivity and orientation is described in this Section. Edge pixel orientation is defined as the line orientation ($\theta' \in [0, \pi)$ in equation (4)) derived from the gradient direction θ at an edge pixel. We have observed and discovered that edge curve extraction is facilitated by first separating maps of edges (especially cluttered edges) into different edge orientation channels. Our approach involves separating edges into N overlapping edge orientation channels, separately decluttering each channel by independently extracting the curves from each channel that have at least $L_{channel} > 0$ edge pixels, recombining curves extracted from each channel into a single composite map of curves, and decluttering the result by extracting only curves with at least $L_{composite}$ edge pixels.

For $i = 0 \ldots N-1$, edge orientation channel i contains all edge pixels for which the edge orientation θ' lies within a range defined in terms of the center $\theta_0(i)$ of edge orientation channel i and the edge orientation channel half-width Δ_θ:

$$\theta' \in [\, \theta_0(i) - \Delta_\theta, \theta_0(i) + \Delta_\theta \,], \quad \theta_0(i) = i\Delta_\theta, \quad \Delta_\theta = \pi / N \tag{7}$$

Note that successive channels overlap by Δ_θ. Successive channels must overlap because curves with pixels whose edge orientations lie near the border between adjacent non-overlapping channels can be missed. Although the exact value is not important, N should be chosen such that Δ_θ is neither too large nor too small. We typically use $N = 20$ (channels of width 18°). If Δ_θ is too restrictive, it will be difficult to form edge curves within a single channel because edge orientations are noisy. If Δ_θ is too large, the disparity between orientations of edges within a channel might be too large.

Edge curves (often broken) are extracted from maps of fully thinned edges using standard boundary following techniques. An eligible edge pixel (i.e., an edge pixel not yet assigned to an edge curve) is first chosen as a seed pixel for boundary following. If the seed pixel has one or two neighbors (i.e., eligible edge pixels in its 8-neighborhood), then the boundary is traced from the seed pixel in one or two directions. Otherwise, the seed pixel is the only edge curve pixel. Once an edge pixel has been traced or chosen as a seed, it becomes ineligible. Boundary following from a traced pixel that is not a seed can continue only if the traced pixel has exactly one neighbor.
A_0 and $L_{composite}$ have meaning in the context of traditional edge detection: A_0 is a threshold on edge strength for edges extracted from gradient local maxima, and $L_{composite}$ is an edge de-cluttering (cleaning) parameter. Our approach to extracting broken edge curves can be viewed as a generalization of traditional edge detection when $N > 1$ edge orientation channels are considered and a third parameter $L_{channel}$ for curve length is factored in. Specifically, in

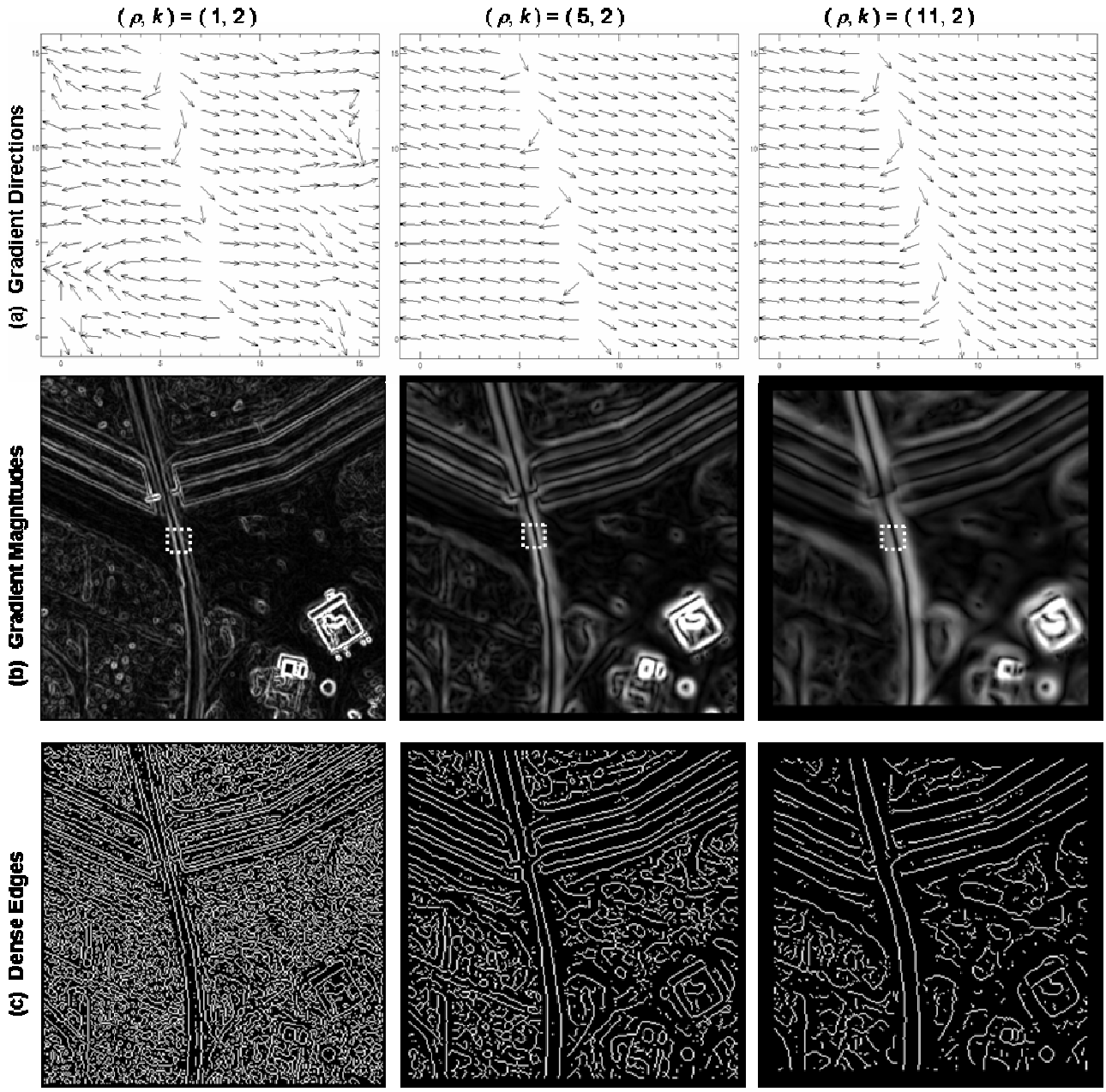


Fig.3 (a) Gradient directions (b) gradient magnitudes and (c) dense edges for the image block and sub-block in Fig.2.

traditional approaches, the edge map E to de-clutter by extracting curves with at least $L_{composite}$ edge pixels is formed by applying some gradient strength threshold $A_0 \geq 0$ to the gradient local maxima. In our approach, E is instead formed by separating the map of all gradient local maxima ($A_0 = 0$) into $N > 1$ edge orientation channels, separately de-cluttering each channel by extracting curves with at least $L_{channel} \leq L_{composite}$ edge pixels, and recombining the de-cluttered channels. Note that for $A_0 = 0$, our approach degenerates to traditional edge detection when either $N = 1$ or $L_{channel} = 0$.

We require $L_{composite} \geq L_{channel} > 0$. Although the exact values assigned to these parameters are not important, too many edges can get removed from each channel when $L_{channel}$ is too large. We often use ($L_{channel}$, $L_{composite}$) = (3,20). Fig.4 shows three edge orientation channels, first unprocessed ($L_{channel}$ = 0), and then de-cluttered ($L_{channel}$ = 3). The edge channels were extracted from the dense edge map shown in the middle of the last row of Fig.3 for which the gradient estimation parameters were (ρ, k) = (5,2). The sets of first all cluttered channels and then all de-cluttered channels were recombined into two composite curve maps. Fig.5 illustrates what happens when these composite curve maps are increasingly de-cluttered. One can see that from the standpoint of extracting extended curves, $L_{channel}$ and $L_{composite}$ play important separate roles. It was only by using $L_{channel}$ and $L_{composite}$ together, that most of the extended edge curves were extracted and most of the edge clutter was simultaneously suppressed (see the lower right hand image in Fig.5).

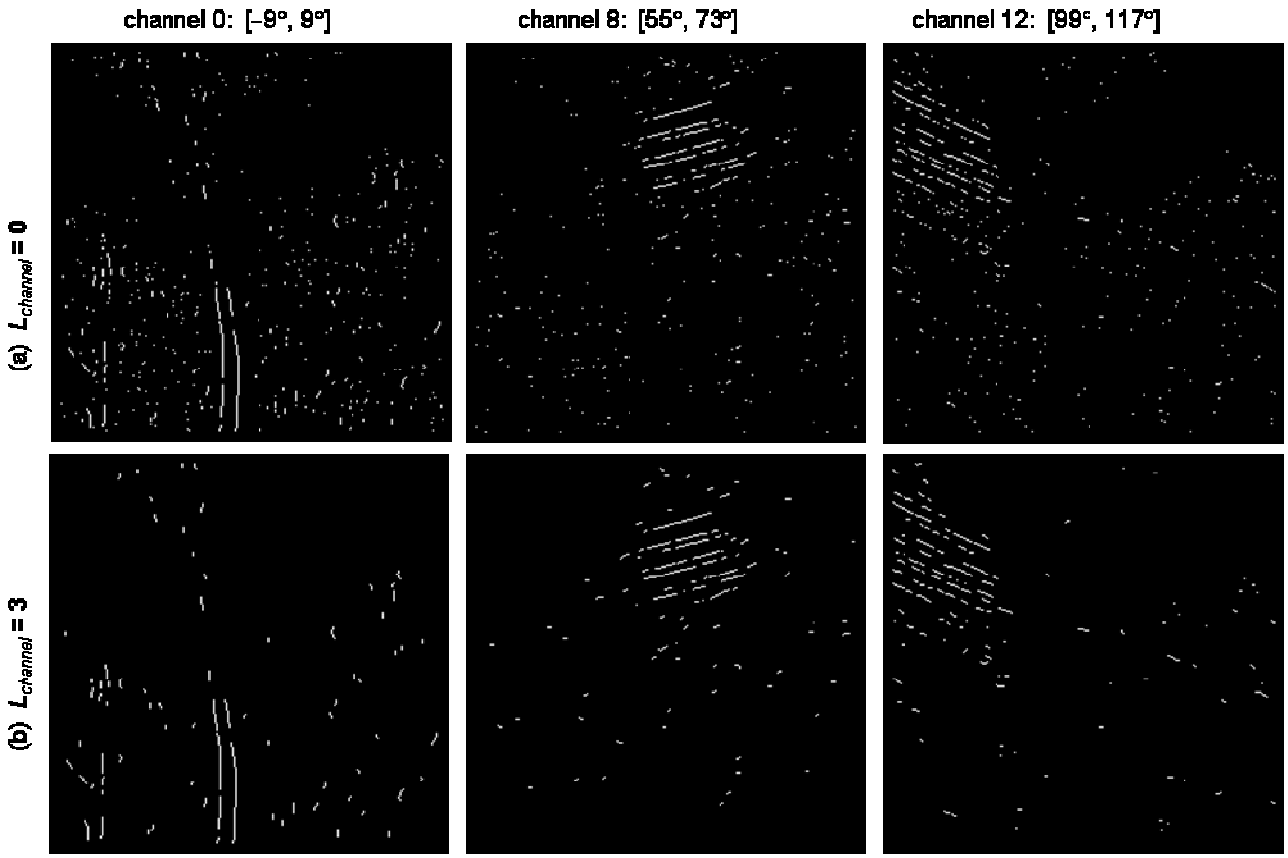


Fig.4 (a) Cluttered and (b) de-cluttered edge orientation channels extracted from the dense edge map shown in the middle of the last row of Fig.3.

4. BROKEN POLYLINES FROM BROKEN CURVES

It is easier to quantify the connectability between broken edge curves when they are modeled as polylines. A robust splitting scheme for segmenting edge curves into polyline segments was proposed in [23]. An edge curve represented as a sequence of 8-connected edge pixels is first modeled by the line segment that connects the curve endpoints. The [α, β, γ] parameterization of the line L containing distinct edge curve endpoints with column and row coordinates [c_0 ,r_0] and [c_1 ,r_1] is given by

(8) $$\alpha c + \beta r = \gamma , \qquad [\alpha, \beta, \gamma] = [\, r_1 - r_0 , c_0 - c_1 , \alpha c_0 + \beta r_0 \,]$$

Fig.5 Maps of broken curves formed by recombining the complete sets of (a) cluttered and (b) de-cluttered edge orientation channels referred to in Fig.4.

The squared distance from a point $[c',r']$ to the line L is

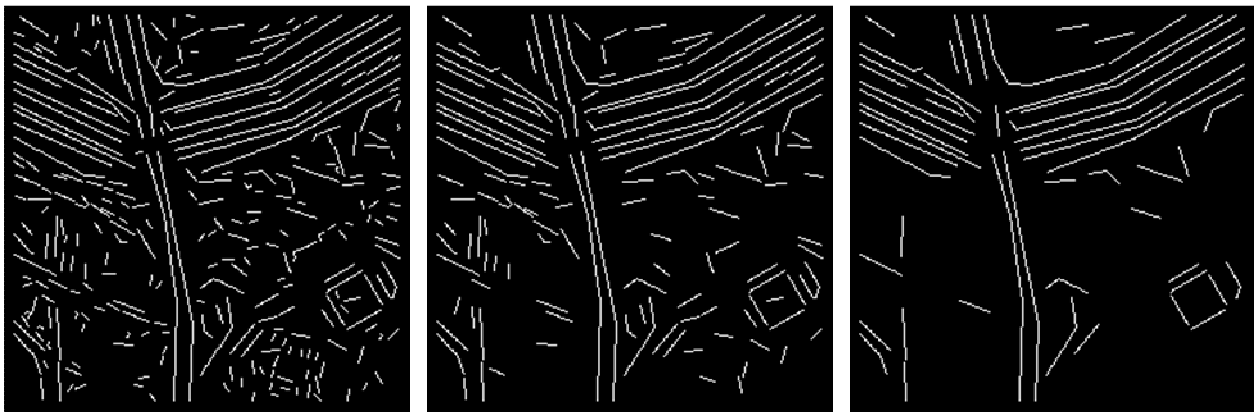
(9) $$D^2([c', r'], L) = (\alpha c' + \beta r' - \gamma)^2 / (\alpha^2 + \beta^2)$$

The edge curve pixel $[c',r']$ that maximizes the distance D from L is found. If $D([c', r'], L) \geq D_0(L)$, the edge curve is split by adding a polyline vertex at $[c',r']$. We define the distance threshold to be

(10) $$D_0(L) = \min(D_{max}, \| c_1 - c_0, r_1 - r_0 \| / 4)$$

where D_{max} is the modeling parameter that regulates the accuracy of the polyline. As D_{max} decreases, the disparity between an edge curve and its polyline model tends to decrease, while the number of polyline vertices tends to increase. In practice, high quality polylines with modest numbers of vertices can be obtained by using $D_{max} = 3$ pixels.

Each time a new polyline vertex is added, two new edge curves are created, and an attempt is made to split them. Convergence is achieved when none of the sub-curves that the original edge curve was divided into can be split further. The resulting polyline models of edge curves are represented as sequences of polyline vertices. Additional polyline attributes are derived and maintained for convenience. These include whether or not the polyline is closed, the length of each polyline segment, the overall length of the polyline, and whether or not the two polyline endpoints are connected to other polylines. Fig.6 shows polyline representations ($D_{max} = 3$) for the de-cluttered broken edge curves in Fig.5(b).



($L_{channel}$, $L_{composite}$) = (3,5) ($L_{channel}$, $L_{composite}$) = (3,10) ($L_{channel}$, $L_{composite}$) = (3,20)

Fig.6 Polyline representations based on D_{max} = 3 for the broken edge curves in Fig.5(b).

5. ALIGNED CONNECTION OF BROKEN POLYLINES

The polylines extracted in the previous step are sometimes disconnected due to occlusion and noise. These broken polylines should be connected if they are sufficiently close and well aligned. It is necessary to mathematically define and quantify the aligned connectability between pairs of line segments in order to find and connect open ends of broken polylines. Consider two line segments L_1 and L_2 as shown in Fig.7. Let $[c_1, r_1]$ be the open endpoint of L_1 closest to an open endpoint of L_2, and let $[c_2, r_2]$ be the open endpoint of L_2 closest to an open endpoint of L_1. For L_1 and L_2, the degree of connectability is based on *endpoint separation*

$$\delta(L_1, L_2) \stackrel{\Delta}{=} [\,(c_1 - c_2)^2 + (r_1 - r_2)^2\,]^{1/2} \tag{11}$$

and *misalignment angle*

$$m\,(\underline{L}_1, \underline{L}_2) \stackrel{\Delta}{=} max\,(\phi_1, \phi_2) \in [0, \pi]\,, \quad \phi_k = cos^{-1} \frac{\underline{L}_0 \cdot \underline{L}_k}{||\underline{L}_0||\;||\underline{L}_k||}\,, \quad k = 1, 2 \tag{12}$$

where $\underline{L}_1$ is the vector along L_1 terminating on the connection point $[c_1, r_1]$, $\underline{L}_2$ is the vector along L_2 emanating from the connection point $[c_2, r_2]$, and $\underline{L}_0$ is the vector emanating from $[c_1, r_1]$ and terminating on $[c_2, r_2]$ (see Fig.7). ϕ_k is thus the angle between $\underline{L}_0$ and $\underline{L}_k$. Colloquially, $m\,(\underline{L}_1, \underline{L}_2)$ is the maximum angle a car driving along L_1 or L_2 would have to turn in order to travel on the connecting L_0 segment.

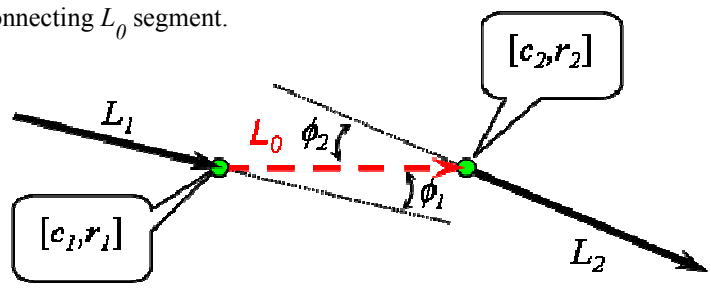


Fig.7 Geometry of aligned connectability between a pair of line segments

L_1 and L_2 are potentially connectable as aligned line segments if $\delta(L_1, L_2) < \delta_0$ and $m(\underline{L}_1, \underline{L}_2) < m_0$ for suitable decision thresholds $\delta_0 > 0$ and $m_0 \geq 0$. For $\delta(L_1, L_2) < \delta_0$, the degree of aligned connectability between L_1 and L_2 increases as $m(\underline{L}_1, \underline{L}_2)$ decreases. L_1 and L_2 are to be connected linearly if they are potentially connectable and there is *optimal two-way connectability* between them. L_1 and L_2 are said to be optimal two-way connectable if

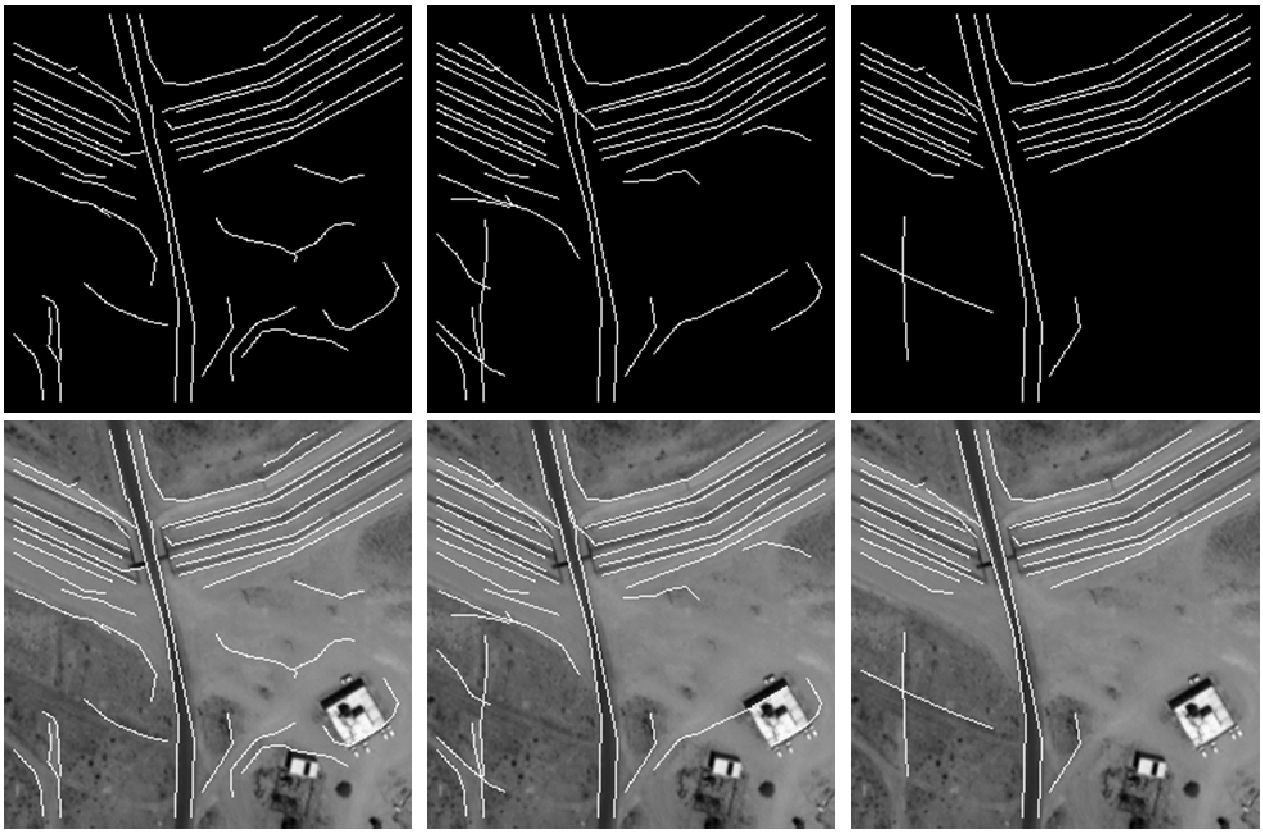
$$L_1 = \underset{L:\ \delta(L,L_2) < \delta_0}{arg\ min}\ m(\underline{L}, \underline{L}_2) \quad \text{and} \quad L_2 = \underset{L:\ \delta(L,L_1) < \delta_0}{arg\ min}\ m(\underline{L}, \underline{L}_1) \tag{13}$$

Our approach for connecting broken polylines is similar to the approach found in [24] except that our approach requires the *optimal two-way connectability* condition to be satisfied.

Once connected, polylines with overall length less than $L_{polyline} \geq L_{composite}$ can be removed. Fig.8 shows connected polylines derived from the broken polylines in Fig.6 with various decision thresholds and $L_{polyline} = 40$. We have observed that smaller $L_{composite}$ values lead to images that contain many short and closely-spaced broken polylines, while larger $L_{composite}$ values give rise to longer polylines that are separated by greater distance. In general, when starting from images with many short and closely-spaced polylines, smaller values of δ_0 are required to reduce the number of spurious connections between distant segments that happen to be well aligned. On the other hand, when starting from images generated using larger $L_{composite}$ values, larger values for δ_0 are required to bridge the larger gaps. At larger values of δ_0, the misalignment angle m_0 needs be smaller to reduce the number of spurious connections between distant segments that are not well aligned, but with smaller values of δ_0, the angle for m_0 can be larger to accommodate nearby segments that form a slowly curving road. For the examples provided in Fig. 8, the decision thresholds δ_0 and m_0 are set manually. In operational practice, δ_0 and m_0 might be set statistically by deriving Bayesian statistical thresholds from populations of existing broken polylines in images (a topic for future research).

6. SUMMARY AND FUTURE DIRECTIONS

A novel approach for automatically extracting extended curves such as roads from overhead images has been presented. It extends traditional approaches to edge and curve detection by (1) using a geometrically motivated gradient estimate which produces coarse noise-suppressed edges that are relatively precise with sharp corners (2) decomposing maps of dense edges into separate edge orientation channels, de-cluttering each channel separately and extracting edge curves from recombined channels, and (3) modeling the curves as polylines, identifying two-way optimal polylines for aligned connection and connecting only those that are sufficiently close and aligned. Although several parameters are involved, there are simple guidelines for choosing most parameter values, and specific values are usually not critical. The exceptions are the decision thresholds for aligned connection of polylines which are now set manually. It should be possible to use populations of short disconnected curves extracted from image blocks to derive Bayesian statistical decision thresholds. A curve consolidation algorithm for replacing groups of closely separated parallel curves with single connected curves that have thickness attributes is also needed. Finally, an algorithm for extending curves across successive image blocks is needed so that curve extraction can be extended to overhead images of arbitrary size.



$(L_{channel}, L_{composite}) = (3,5)$
$[\delta_0, m_0, L_{polyline}] = [12.5, \pi/4, 40]$.

$(L_{channel}, L_{composite}) = (3,10)$
$[\delta_0, m_0, L_{polyline}] = [25, \pi/5, 40]$.

$(L_{channel}, L_{composite}) = (3,20)$
$[\delta_0, m_0, L_{polyline}] = [50, \pi/12, 40]$.

Fig.8 Connected polylines derived from the broken polylines in Fig.6. Top row: connected polylines. Bottom row: connected polylines overlay on original image.

REFERENCES

[1] J. M. Brase, D. W. Paglieroni, D. N. Poland, G. F. Weinert, C. W. Grant, A. S. Lopez and S. Nikolaev, "Image Content Engine (ICE): A System for Fast Image Database Searches", SPIE Defense and Security Symposium: Optics and Photonics in Global Homeland Security, Proc. SPIE, Vol. 5781, 28-30 March 2005, Orlando, FL.

[2] C. Steger, "An Unbiased Detector of Curvilinear Structures", IEEE Trans. PAMI, Vol. 20, No. 2, February 1998, pp. 113-125.

[3] A. Baumgartner, W. Eckstein, C. Heipke, S. Hinz, H. Mayer, B. Radig, C. Steger and C. Wiedemann, "T-REX: TUM Research on Road Extraction", Festschrift für Prof. Dr. Ing. Heinrich Ebner zum 60. Geburtstag, Lehrstuhl für Photogrammetrie und Fernerkundung, Technische Universität München, 1999, pp. 43 – 64.

[4] M. A. Fischler, J. M. Tenenbaum and H. C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique", Computer Graphics and Image Processing, Vol. 15, No. 3, 1981, pp. 201-223.

[5] D. M. McKeown and J. L. Denlinger, "Cooperative Methods for Road Tracking in Aerial Imagery", Proc. Of IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 662-672, 1988.

[6] G. Vosselman and J. de Knecht, "Road Tracing by Profile Matching and Kalman Filtering", Proc. Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, 1995, pp. 265-274.

[7] P. Perez, A. Blake, and M. Gangnet, "Jetstream: Probabilistic Contour Extraction with Particles", Proc. International Conference on Computer Vision, 2001.

[8] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Communications of the ACM, Vol. 15, No. 1, January 1972, pp. 11-15.

[9] X. Hu, C. V. Tao and Y. Hu, “Automatic Road Extraction from Dense Urban Area by Integrated Processing of High Resolution Imagery and LIDAR Data”, Proc. XXth ISPRS Congress, 12-23 July 2004, Istanbul, Turkey.

[10] M. Bicego, S. Dalfini, G. Vernazza and V. Murino, "Automatic Road Extraction from Aerial Images by Probabilistic Contour Tracking", Proc. of IEEE Int. Conf. on Image Processing, Vol. III, 2003, pp. 585-588.

[11] R. Bajcsy and M. Tavakoli, “Computer Recognition of Roads from Satellite Pictures”, IEEE Transactions Systems, Man, and Cybernetics, Vol. 6, No. 9, September 1976, pp. 623-637.

[12] A. K. Shackelford and C. H. Davis, “A Combined Fuzzy Pixel-Based and Object-Based Approach for Classification of High-Resolution Multispectral Data over Urban Areas”, IEEE Trans. Geosci. Remote Sens., Vol.41, No.10, October 2003, pp.2354-2363.

[13] L. H. Quam, “Road Tracking and Anomaly Detection in Aerial Imagery”, Proc. Image Understanding Workshop, May 1978, pp 51-55.

[14] D. Gabor, “Theory of Communication”, J. IEE (London), Vol.93, 1946, pp.429-457.

[15] D. Dunn, W. Higgins and J. Wakeley, “Texture Segmentation using 2D Gabor Elementary Functions”, IEEE Trans. Pattern Anal. Mach. Intell., Vol.16, February 1994, pp.130-149.

[16] B. S. Manjunath and W. Y. Ma, “Texture Features for Browsing and Retrieval of Image Data”, IEEE Trans. PAMI, Vol.18, No.8, August 1996, pp.837-842.

[17] D. Marr and E. Hildreth, “Theory of Edge Detection”, Proc. Roy. Soc. London, B207, 1980, pp. 187-217.

[18] J. Canny, “Computational Approach to Edge Detection”, IEEE Trans. PAMI, Vol. 8, No. 6, November 1986, pp. 679-698.

[19] D. W. Paglieroni, W. G. Eppler and D. N. Poland, “Phase Sensitive Cueing for 3D Objects in Overhead Images”, SPIE Defense and Security Symposium: Signal Processing, Sensor Fusion and Target Recognition XIV, Proc. SPIE, Vol .5809, 28-30 March 2005, Orlando FL.

[20] J. Prewitt, “Object Enhancement and Extraction”, in Picture Processing and Pyschopictorics, B. S. Lipkin and A. Rosenfeld, eds., Academic Press, New York, 1970.

[21] S. Sengupta, A. Lopez, J. Brase and D. W. Paglieroni, “Phase-Based Road Detection in Multi-Source Images”, Proc. IGARSS04, September 2004.

[22] I. E. Sobel, “Camera Models and Machine Perception,” Ph.D. dissertation, Stanford University, Palo Alto, CA, 1970.

[23] K. J. Turner, “Computer Perception of Curved Objects Using a Television Camera”, Ph.D. Dissertation, Univ. Edinburg, 1974.

[24] S. Vasudevan, R. L. Cannon and J. C. Bezdek, “Heuristics for Intermediate Level Road Finding Algorithms”, Computer Vision, Graphics, and Image Processing, Vol. 44, No. 1, November 1988, pp. 175-190.