An Unsupervised Road Extraction Algorithm for Very Low-Resolution Satellite Imagery *

Fatih Porikli

Electrical Engineering Department
Polytechnic University
Brooklyn, NY 11201

Trish Keaton

Information Sciences Lab
HRL Laboratories, LCC
Malibu, CA 90265

Abstract

In this paper, we present a novel approach for the automatic extraction of roads from very low-resolution satellite imagery, such as images found in the SPOT data set. First, the input image is preprocessed to amplify possible road regions, while suppressing the low spatial frequency areas. Then, the road features, which are chosen to be curvilinear structures, are found by evaluating the local responses from a set of nonlinear directional line filters. A mapping from the line domain to the vector domain (LVT) was devised to compute the line strength and orientation for each image point. This transformation enables us to combine the multi-channel data into a single aggregated response. From the line orientations and strengths, the candidate road points are traced and linked in a recursive manner. The iterative process consists of finding the road segments, passing them through the directional line filter set and computing new strengths and orientations, and then fusing these values with those obtained in the previous iteration. The tracing results are updated at each iteration, and the process continues until there are no further changes in the roads extracted. Experimental results obtained by processing panchromatic satellite images from the SPOT data set, demonstrate the success of the proposed algorithm.

1 Introduction

Unsupervised extraction of roads eliminates the need for human operators to perform the time consuming and expensive process of mapping roads from satellite imagery. As increasing volumes of imagery become available, fully automatic methods are required to interpret the visible features such as roads, railroads, drainage, and other meaningful curvilinear structures. There exists an even greater need for a mechanism that handles very low resolution images. The essence of detecting curvilinear elements is also related to the problem of deriving anatomical structures in medical imaging as well as locating material defects in quality control systems, and other geomorphologic and cartographic applications.

Typical road extraction algorithms consist of two stages: the detection of road points, and the concatenation of those points into road segments. The detection step is supported by *a priori* information based on the topological constraints, e.g., offsetting water bodies, limiting extraction within the same isobar. Radiometric road primitives such as maximum curvature, constant width, intensity smoothness, provide additional rules for the concatenation step. Many approaches combine a local criterion based on radiometry within some small neighborhood to discriminate roads from the surrounding background, and a global criterion to introduce priori information about the structures. Detection is often performed by an edge or line operator [1], differential geometry [2], or an analysis of the road profile [3]. In the simplest case, a straightforward connection of the detected line pixels is used to describe the road. Dynamic programming as in [4] can be used to minimize a global cost function, and heuristics included to the minimum cost path estimation framework [6], Hough transform based curve detection approaches, parametric curve models such as snakes and B-splines, and Bayesian networks [5] are used to aggregate low-level road pixel detection into road segment estimates.

Most of the proposed road detection algorithms were designed to extract roads from high-resolution images and often require user assistance to mark both starting and ending points of road segments. By using various road features as confidence measures, a minimum cost path is derived between the start and end nodes. Due to the noise sensitivity, asymmetry of the

[*] This research was supported by a contract from Raytheon Systems Company, Science & Technology, Garland, TX.

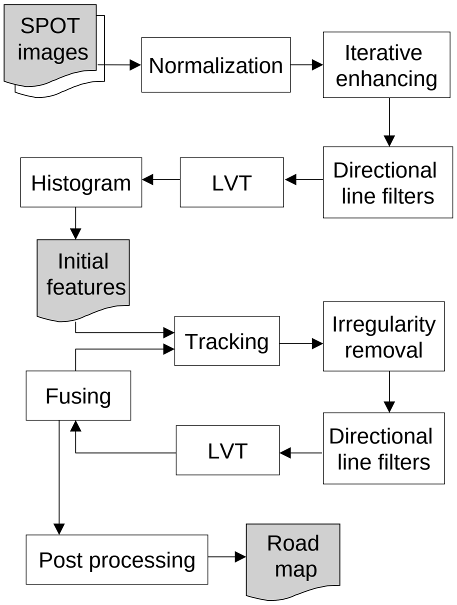


Figure 1: Flow diagram.

contrast at the both sides of the edges, and the difficulty of obtaining precise edge directions, edge based methods are inadequate for very low-resolution imagery.

We propose an unsupervised detection algorithm that finds road confidences as well as associated orientations by using the responses from nonlinear directional line filters. Section II discusses the deriving of the road confidences, and the line-to-vector transformation process. In Section III, the tracing algorithm, feedback loop, and post processing operations are explained. Finally, the later section presents experimental results obtained from images in the SPOT data set.

2 Derivation of Road Confidences

In this section, we discuss the preprocessing of the input image, and the automatic extraction of curvilinear structures that are used as road features for tracing the road segments, and for assigning road confidence measures to the image points. A general flow diagram of the algorithm is shown in Fig. 1.

2.1 Preprocessing

Although the orbit of the SPOT satellite is synchronous with the sun so that the lighting of each image is supposed to be identical, we still found that the panchromatic band of the images are substantially different from each other. Our aim is to develop an unsupervised road extraction scheme that is adaptive to the input images. Thus, a preprocessing stage was necessary to equalize the intensity variations of the input images acquired at different illuminations conditions before proceeding with the extraction of roads.

The input images are first intensity normalized such that the spectrum of each input image covers the maximum intensity levels that is 2^n levels for n-bits coded imagery. Normalization helps to compensate possible illuminance differences between the various images. After normalization, we enhance the images. One important observation on the nature of the road points is that such points tend to have a higher intensity value than the surrounding region in the panchromatic channel of the SPOT data set. Thus, the feature extraction stage is adapted to detect high intensity linear structures. In contrast, water bodies exhibit a darker appearance. In order to detect water streams instead of roads, we simply modify the preprocessing stage to reflect the desired contrast we wish to find. The detection of water bodies would require that the normalized image be inverted in intensity. To segment roads, we iteratively increase the intensity value of a pixel if its value is much higher than f_{mean} the average intensity value within a circular window around the pixel, and still less than f_{max} the maximum in the window. The constraint on f_{max} is included to prevent amplification of the noise. Thus, the intensity difference between a road point and an immediate adjacent background point is amplified as well as any noise, which is filtered in the later stages. Let $f(x, y)$ be the normalized intensity function of an image. Then the enhancing is done as

$$f_n = \begin{cases} f_{n-1} + \varepsilon & f_{max} \gg f_{n-1} > f_{mean} \\ f_{n-1} & otherwise \end{cases} \qquad (1)$$

where n is the current iteration index, (x_k, y_k) is a point in the window, and ε is used as convergence step size. The results of preprocessing are shown in Fig. 5,a-b.

2.2 Directional Line Filters

A line segment can be characterized as an elongated rectangular region having a homogeneous intensity level that is bounded on both its longer sides by homogeneous regions having a different intensity level. The assumption that road segments will have the same contrast on both sides is rarely true for real images. Therefore, a semi-linear structure that finds step edges on the either side of the line, was devised as inspired by Vanderburg [8]. An adequate line detector should also be able to detect very narrow lines (1-2 pixels),

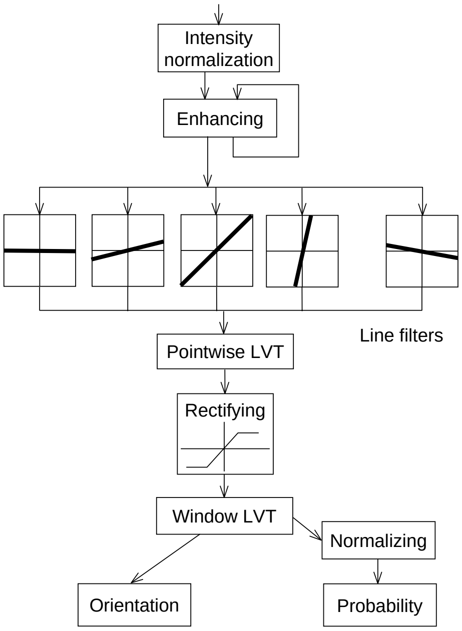


Figure 2: Line detection by nonlinear directional filters.

as well as wider ones (5-6 pixels). Unlike speckles and edges, a line point is generally bordered by other adjoining line points located on opposite sides. Hence, to filter out image noise and avoid detecting sparks as road points, we use a filter which produces higher scores for the longer line structures. The filter template is stretched along its detecting orientation. However, using only two of such templates, as done by orthogonal pairs in edge detection, limits the accuracy of line detection especially for lines that are diagonally oriented. This is accomplished by extending the filter template on the direction perpendicular to its detection orientation to include the distant points from the line center. However, such an extension neglects the continuity property of the lines and introduces false errors especially in presence of speckle noise. To prevent errors, i.e. catching sparks as road points, the filter template is stretched along its detecting orientation. Yet, using only two of such templates, as done by orthogonal pairs in the edge detection, limits the accuracy of the line detection especially for lines that are diagonally oriented. Therefore, we employ a compass type directional filter set containing multiple line filters tuned at the different orientations. Here, compass means each separate filter in the bank operates at a different orientation θ_i and the entire set covers the full orientation spectrum $[0, \pi]$. The basic filter g_i consists of two zero-padded Gabor filter type half templates g_i^a, g_i^b to detect the step edges on the either side of the line. The basic filter is shaped as a Gabor filter to give more weight to the center pixels, to achieve the scalability for different line widths, and to simplify design of each directional filter.

$$g_i = \cos\left(c_1(x \cos\theta_i + y \sin\theta_i)\right) e^{-\frac{x^2+y^2}{c_2}}. \qquad (2)$$

The constants c_1, c_2 determine the shape and width of the matching template. From half template responses, a strength s_i at each pixel position is calculated for each direction θ_i as

$$s_i = \begin{cases} g_i^a + g_i^b & g_i^a + g_i^b \geq 0 \\ 0 & g_i^a + g_i^b < 0 \end{cases} \qquad (3)$$

2.3 Line-to-Vector Transform

A problem of fusing all of the above line strengths immediately arises. One cannot directly sum up and average the orientation angles because of the ambiguity at the limits of the angular spectrum $[0, \pi)$. For example, two lines with orientation angles $\pi - \epsilon$ and ϵ lie in similar directions, however averaging their orientation angles gives $\frac{\pi}{2}$ which is almost perpendicular to both lines. Essentially, having relatively significant strengths for any orthogonal filter pair is an ambiguity.

To eliminate incompatible filter outcomes, and to fuse any number of line strengths, a novel line-to-vector transform (LVT) was developed. LVT is a mapping from $1\frac{1}{2}$-D direction domain to 2-D vector domain such that the perpendicular line contrasts become reciprocal to each other. As a line orientation becomes more similar to a directional filter, its response from the perpendicular filter should attenuate. This property can be exploited to find the orientation of the lines which lie between compass filter orientations instead of just selecting the direction of the filter having the maximum magnitude. If the filter directions are represented such that perpendicular responses cancel each other's effect out, then it is possible to fuse all the filter responses to derive an aggregated line orientation and strength. Thus, the angular spectrum of orientation is extended from $[0, \pi)$ to $[0, 2\pi)$ and

$$s_i(\theta_i) \overset{LVT}{\rightarrow} \vec{s}_i(\omega_i) = s_i e^{j2\theta_i} \qquad (4)$$

where $\omega_i = 2\theta_i$, s_i is the response and direction for the i^{th} directional template.

Thus, perpendicular filter pairs are converted to inverse directions, and likewise non-perpendicular ones are correlated. By adding the transformed vectors, we subtract the responses of the perpendicular filters and amplify those of non-parallel filters

$$\vec{s} = \sum_{i=0}^{N-1} \vec{s}_i(\omega_i) \qquad (5)$$

The resulting vector is transformed to a line by halving the phase component.

LVT also enables us to fuse filter responses for multi-spectral channels in a similar way. Let M is the number of the multi-spectral band, i.e. the red, green, near infrared images. If $\vec{s}_{i,j}$ is the LVT for the i^{th} filter in the j^{th} image of the multi-spectral image set, then the resulting response is going to be

$$\vec{s}_{M,N} = \sum_{j=1}^{M} \sum_{i=0}^{N-1} \vec{s}_{i,j}. \qquad (6)$$

2.4 Consistency Evaluation

Obtaining line strengths as explained in the previous section can be corrupted by extreme responses that cause a global attenuation of the correct line strengths. Moreover, the small valued line strengths mix up and divert the tracing algorithm. Therefore, a histogram based thresholding filter was applied to the strength and orientation data. First, the histogram of the line strength is calculated on a linear scale. The strength values of the points larger than an upper threshold is rectified to the upper threshold value. The strength values smaller than the lower threshold are zeroed. The lower and upper thresholds are the values of the histogram variant such that the sum of the histogram values up to those points are equal to the certain percentages of the total number of the image points.

The above analysis provides point-wise evaluation of the line likelihood. To achieve local consistency, LVT is performed again within a window, preferably circular, at each pixel position. Application of the LVT attenuates the line strengths if the computed line orientations within the window exhibit a high variance. After normalization to unity, the confidence of being a road point and the line orientation becomes as

$$p[x,y] = \frac{|\vec{s}(x,y)|}{|\vec{s}_{max}|}, \;\; \theta(x,y) = \frac{\angle \vec{s}(x,y)}{2}. \qquad (7)$$

The initial line strengths and orientations computed are presented in Fig. 5 (c-d).

3 Tracking and linking

3.1 Tracking

Given the confidences, the road points that have a relatively higher probability are linked into curvilinear segments. The road tracing algorithm is designed such that it can recursively link line points possessing similar characteristics. Even when the road confidences are not spatially continuous, the tracing algorithm can overcome this problem, and fill in gaps when a consistent link exists in the constrained neighborhood. First, a road point is chosen if its confidence value is high enough to initialize a new road segment. Each line segment is characterized by its end-to-end global direction, local tangential direction, total length, average curvature, and the variation of its global direction. A point in the window that produces the highest similarity score is chosen as a valid connector, and the segment descriptors are updated with each included extension to the segment. The search window is shaped as a folium-like quadratic, and is approximated by a polygon. The parameters a, b of the adaptive window are determined by the descriptors of the segment. For long segments the search window length parameter a is larger. The window width is inversely proportional to the parameter b, and is reciprocal to the segment length, and directly proportional to the global direction variance. Both the local and global directions of the segment determine the orientation of the search region. The span angle β depends on both a and b. To simplify indexing, the search region is mapped to a unit shape as in Fig. 3.

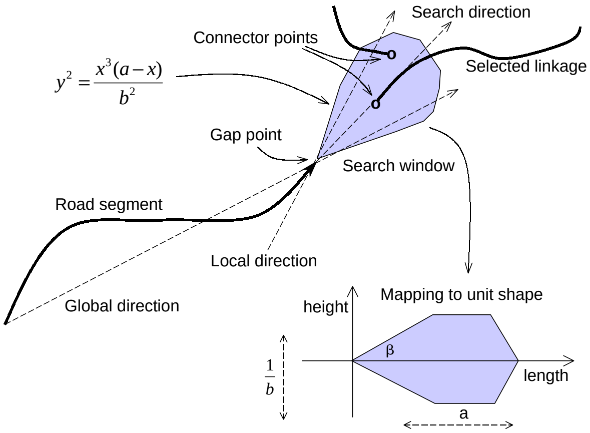


Figure 3: Tracing search window.

The point that gives the maximum similarity score is selected as the next connector point. Closer a point in the search window to the current point, its similarity score becomes larger. For each possible road point, the LVT calculates a magnitude by using the confidences and orientations of the current and the candidate points, and a third term which has the orientation of the line connecting the spatial positions of the current and the candidate points. The magnitude of the last term is the average of the magnitudes of the first two points. Let the $\vec{m}_{cur}$, $\vec{m}_{can}$ and $\vec{m}_{spa}$ stand for the LVT transformed values of the current, candidate and spatial terms such that $\vec{m}_{cur}$ is

$$\begin{aligned} |\vec{m}_{cur}| &= p[m_{cur}(x,y)] \\ \angle\vec{m}_{cur} &= 2\theta(m_{cur}(x,y)), \end{aligned} \tag{8}$$

$\vec{m}_{can}$ is

$$\begin{aligned} |\vec{m}_{can}| &= p[m_{can}(x,y)] \\ \angle\vec{m}_{can} &= 2\theta(m_{can}(x,y)), \end{aligned} \tag{9}$$

and $\vec{m}_{spa}$ is defined as

$$\begin{aligned} |\vec{m}_{spa}| &= \frac{p[m_{cur}(x,y)]+p[m_{can}(x,y)]}{2} \\ \angle\vec{m}_{spa} &= \tan^{-1}(\frac{m_{can}(y)-m_{cur}(y)}{m_{can}(x)-m_{cur}(x)}). \end{aligned} \tag{10}$$

Given the transformed values for each point within the search window, we determine a similarity score τ_{dir} that is equal to the magnitude of the sum of three vectors:

$$\tau_{dir} = \frac{1}{3}|(\vec{m}_{can} + \vec{m}_{cur} + \vec{m}_{spa})|. \tag{11}$$

A second similarity term τ_{line} is derived by accumulating the confidences of the points on the line N_{line} connecting the gap and the connector points:

$$\tau_{line} = \frac{1}{M} \sum_{(i,j)\in N_{line}} p[i,j] \tag{12}$$

where M is the number of points on the N_{line}.

Thus, the similarity score is the weighted average of the three values

$$\tau(i,j) = c_1\tau_{dir} + c_2\tau_{line} + \tau_{dist}. \tag{13}$$

If c_1 is assigned larger than c_2, the tracing algorithm will choose points such that their line orientations are consistent with the gap point regardless of the confidences of the points between them. Therefore, points forming a discontinuous road segment but are still spatially separated, will be neglected if they have noisy direction estimates. On the other hand, by selecting c_2 larger, the road segment will be matched with the underlying road points even if the orientations of points are not consistent as in the case of no roads present. The third term τ_{dist} is a linear weight that is proportional to the distance from the center point. The tracing process outputs a binary road map.

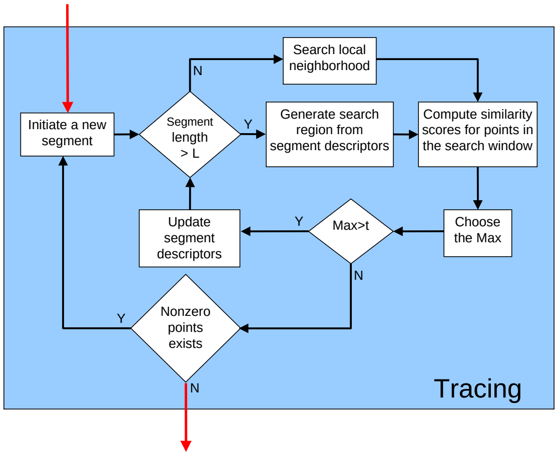

Initiate a new segment
Segment length > L
N
Y
Search local neighborhood
Generate search region from segment descriptors
Compute similarity scores for points in the search window
Choose the Max
Max>t
Y
N
Update segment descriptors
Nonzero points exists
Y
N
Tracing

Figure 4: Tracing stage flow diagram.

3.2 Irregularity Removal

The road maps generated by tracing process contain topological irregularities such as singular points, spurs, loops and blobs. Before feeding back the output road map to the directional filter set, these deviations are removed by morphological and contextual operators. First very small loops are filled, then a connected component labeling algorithm is used to calculate the areas of non-road regions bounded by the road segments. The algorithm uses recursion to give the same label to the non-road points in 8-neighborhood of each other's. If any of the labeled regions has a small area that physically cannot correspond to separate roads, it is assigned as a road. Thinning of road segments is required, since the tracing stage may produce wide road segments, and the removal of very small loops may causes blobs. A morphological operator using the hit-and-miss transform [7] is utilized to accomplish the thinning of the road map.

3.3 Feedback

The initial tracing is done without using any priori information to validate the accuracy of the obtained segments. Any additional road information supplied to the feature detection stage, which is the directional filter set in our case, evidently improves the quality of the estimation of road confidences and thus the tracing results. Unlike the initial image, the extracted road segments are clean, e.g., do not contain speckle type of noise or similar spatial impurities, which cause excessive spurs and diversions. For that reason, the orientation values computed by using the extracted road segments will be more accurate. Since the tracing stage overcomes road discontinuities in the input

	Image set A		Image set B	
	DR	TR	DR	TR
5 × 5 Proposed	%95.26	%83.68	%97.65	%89.42
5 × 5 Steger's	%85.01	%43.22	%78.31	%46.28
3 × 3 Proposed	%92.17	%79.08	%93.39	%86.47
3 × 3 Steger's	%77.72	%42.29	%72.29	% 40.58

Table 1:

image by filling in the gaps in the road confidence map as explained before, the new line strengths and as a result the new road confidences will have less discontinuities. Also, using the extracted road segments as a feedback can restrain the confidence values of the background points.

Hence, the initial road map is used as an input to the directional line filters, and new road confidences and orientations are computed. The processed intensity road map provides a better estimate of the road orientations and confidences. However, at the both ends of the road segments, fuzzy road regions are produced. Employing LVT once again in a neighborhood and comparing the LVT result to the orientation value of the target point filters such fuzzy regions. The points that have nonnegative confidences and not in the previous road map, are compared with the LVT orientation result. If the difference is small, they are kept, otherwise they are accepted as outliers. As a result, new confidence and orientation maps are obtained. These maps are fused with the previous maps. At an image point, linear weighting blends the new and the previous confidence values, and applying the LVT accordingly gives the orientation value. The progress of the feedback stage is presented in Fig. 5, d-f.

4 Results and Conclusion

We tested our road extraction algorithm on panchromatic images of the SPOT data set. The input images are presented in Fig. 5-a and Fig. 6-d. In these very-low resolution satellite images, the average road width corresponds to 1-5 pixels, and both of the images are slightly distorted by speckle noise. The input images were first normalized to extend the gray level distribution of image intensity to the entire visible spectrum, in our case 2^8 levels. The enhanced images are shown in Fig. 5-b. We used 8 pair-wise orthogonal directional filters to extract line segments. The intermediate results of the feedback loop are shown in Fig. 5, d-f. The final results are shown in Fig. 6, c-f.

We compared our results to Steger's [2] curvilinear structure extraction algorithm. We optimized the parameters of his algorithm so that it can derive the maximum number of road segments without introducing spurious segments. Two metrics were devised to compute their accuracy, the first metric is the ratio of the detected points that are ground truth to the total number of ground truth points. The second is the percentage of the ground truth points that were detected to the total number of the detected points. The ground truth points were annotated by hand. Because both of the algorithms output a skeleton of detected roads, the percentages were calculated in a search window around the pixels rather than using the exact location for comparison. As visible in Table. 1, our unsupervised method acquires more road points than the user optimized algorithm by Steger, and the amount of erroneous points is much less.

References

[1] R. Nevatia and R. Babu, "Linear feature extraction and description", *Comp. Graphics and Image Proc.*, Vol. 13, 1980.

[2] C. Steger, "An unbiased detector of curvilinear structure", *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. 2, 1998.

[3] W. Groch, "Extraction of line shaped object from aerial images using a special operator to analyze the profiles of function", *Comp. Graphics and Image Proc.*, Vol. 18, 1982.

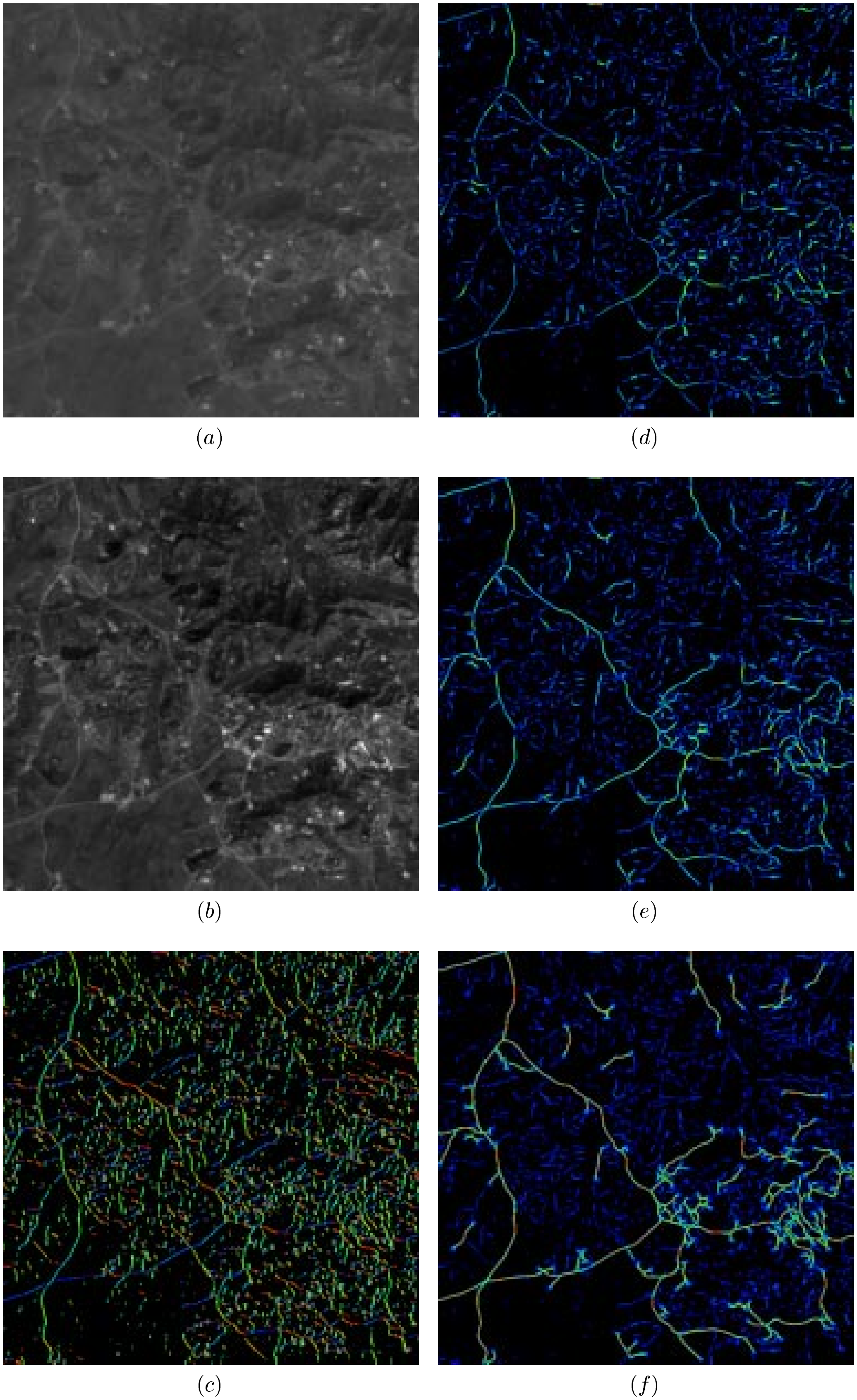
[4] M. Fischler, J. Tenenbaum and H. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique", *Comp. Graphics and Image Proc.*, Vol. 15, 1981.

[5] F. Tupin and H. Maitre, "Detection of linear features in SAR images: application to road network extraction", *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 36, March 1998.

[6] J. Trinder and Y. Wang, "Automatic road extraction from aerial images", *International Workshop on Image Analysis and Information Fusion*, 1998.

[7] B. Jang and R. Chin, "Analysis of thinning algorithms using mathematical morphology", *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. 12, 1990.

[8] G. Vanderburg, "Line detection in satellite imagery", *IEEE Trans. on Geoscience Electronics*, Vol. 14, 1976.



(a) (d)

(b) (e)

(c) (f)

Figure 5: The original gray level image (a), intensity enhanced result (b), and line filter orientation response (c). Because orientation spectrum is cyclic, red and blue represents similar angles, blue-0 and red-π. The initial confidence (d), road confidence after 2^{rd} iteration of the feedback loop (e), and after 5^{th} iteration(f).

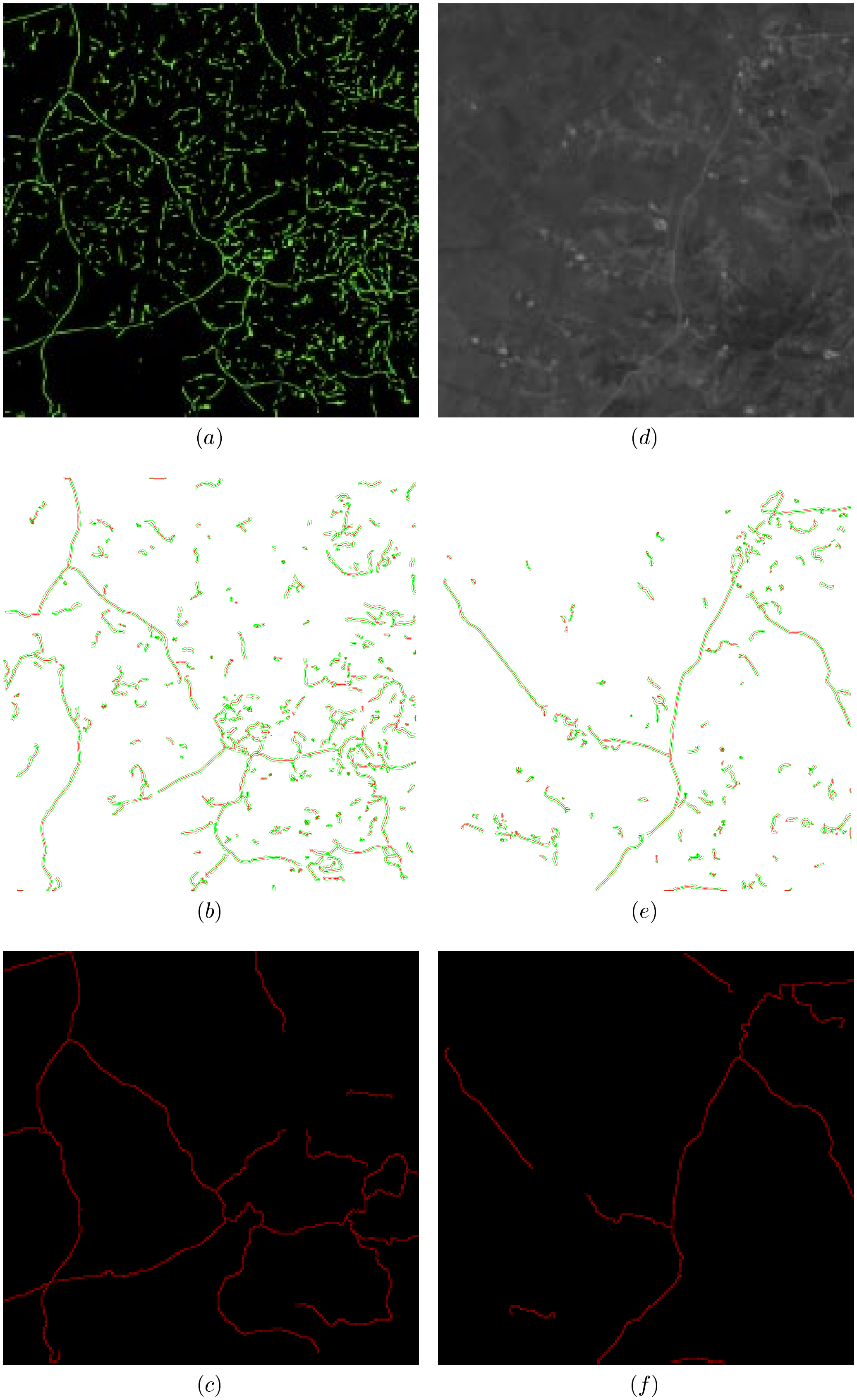


Figure 6: Extraction results; the initial tracing road map (a), Steger's results (b), the final road maps after spur removal (c). A second set of results for a different input image (d-f).