Road Extraction by Point-wise Gaussian Models

Fatih Porikli

Mitsubishi Electric Research Laboratories, Murray Hill, 07974 USA

ABSTRACT

We present a Gaussain model based approach for robust and automatic extraction of roads from very low-resolution satellite imagery. First, the input image is filtered to suppress the regions that the likelihood of existing a road pixel is low. Then, the road magnitude and orientation are computed by evaluating the responses from a quadruple orthogonal line filter set. A mapping from the line domain to the vector domain is used to determine the line strength and orientation for each point. A Gaussian model is fitted to point and matching models are updated recursively. The iterative process consists of finding the connected road points, fusing them with the previous image, passing them through the directional line filter set and computing new magnitudes and orientations. The road segments are updated at each iteration, and the process continues until there are no further changes in the roads extracted. Experimental results demonstrate the success of the proposed algorithm.

Keywords: Relevance feedback, road extraction, line filters, Gaussian models

1. INTRODUCTION

The huge amount of satellite image data requires robust and fully automatic methods to interpret the meaningful image 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. 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 usually supported by topological constraints, e.g., offsetting water bodies, limiting the detection process within the same isobar. Radiometric road primitives such as maximum curvature, constant width, intensity smoothness, may 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[1] or line operator,[6] 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[4] can be used to minimize a global cost function, and heuristics included to the minimum cost path estimation framework,[7] 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[13] 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 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 a model based road detection algorithm that finds road confidences as well as associated orientations by using the responses from a set of directional line filters. We previously proposed a road extraction algorithm.[6] Here, we improve upon the road detection by integrating road model functions. The detection results are further improved by a relevance feedback mechanism. We also provide a solution to recover poorly visible road segments in the original images. Section II discusses intensity normalization, texture removal, line filtering, and computation of the road confidences. Section III presents extraction of road segments, morphological operators, feedback mechanism, and fusion stages. The later section presents experimental results obtained from images in the SPOT data set.

Further author information: (Send correspondence to fatih@merl.com)

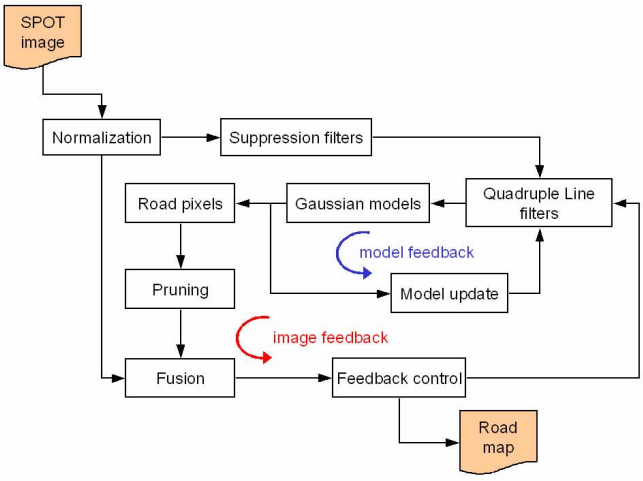

Figure 1. Flow diagram: The model feedback loop propagates the road features to the neighboring pixels, and the image feedback loop enhances the derived road pixels on the input image for the next iteration.

2. LINE FILTER AND ROAD MODEL

In this section, we discuss the filtering of the redundant spatial texture from the input image, computation of line strength and orientation, fitting Gaussian models to points, and iterative updating of the point models. A general flow diagram of the algorithm is shown in Fig. 1.

2.1. Normalization and Texture Removal by Suppression Filter

Depending on the atmospherical and geographical conditions, the lighting of the satellite images substantially differ form each other. Thus, the dynamic intensity range of the input image requires adjustment before proceeding with the extraction of road features. We normalized the input image such that its spectrum covers the maximum available intensity range 0-2^n levels for an n-bits coded image. Normalization process reduces the need for fine-tuning of the filter parameters, and helps to compensate for possible illuminance differences between the various images.

One important observation on the nature of the road points is that such points tend to have a higher intensity values than their surrounding area. In other words, if a point has similar or lower intensity value than the points within its immediate neighborhood it is less likely to correspond to a road point. By using a suppression filter, we smooth the intensity distribution and remove the low spatial texture without disturbing the edge structure for such regions. The suppression filter involves very simple operators. Let I denotes the input image. We compare the intensity value $I(p)$ of a point p with its neighbors. If the distance is less than a threshold δ, the point's intensity value is updated by the average of its neighbors within a local window as

$$I(p) = \begin{cases} \mu(p) & d \leq \delta \\ I(p) & d > \delta \end{cases} \quad (1)$$

$$d(p) = |I(p) - \mu(p)| \quad (2)$$

$$\mu(p) = \frac{1}{K} \sum_{q \in L_p} I(q) \quad (3)$$

where L_p is the local neighborhood that contains K neighbors of the center point p. We observed that $\delta \simeq 4$ is a good compromise for 2^8 bits coded images after the normalization. The iterative application of the suppression filter would effectively remove the small texture while preserving the roads due to the fact that small differences are suppressed more at each cycle. However, we found the amount of the change in the image decreases significantly after the first pass.

2.2. Line Filter

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.[9] An adequate line detector should also be able to detect very narrow lines (1-2 pixels), 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 and the entire set covers the full orientation spectrum $[0, \pi]$. For a $M \times N$ kernel, the basic filter operating at orientation θ is given as

$$g(i,j) = \cos\left(\frac{\pi(i\cos\theta - j\sin\theta)}{2M}\right)\cos\left(\frac{\pi(i\sin\theta + jcos\theta)}{N}\right) \tag{4}$$

where $i = 0,..,M$ and $j = 0,..,N$. The constants kernel size M, N determine the shape and width of the matching template. The basic filter is designed such that it has higher values towards the center to make it less sensitive to the noise. Since a line is basically constructed by two opposite edges, the basic filter g consists of two zero-padded edge filter templates g^a, g^b to detect the step edges on the either side of the line such that

$$g^a(i,j) = \begin{cases} g(i,j) & i > 0 \\ 0 & i < 0 \end{cases}, \quad g^b(i,j) = \begin{cases} g(i,j) & j > 0 \\ 0 & j < 0 \end{cases}$$

From the half template responses, a line strength $s(p, \theta_i)$ at each point p for each filter g_i is calculated as

$$s(p,\theta_i) = \begin{cases} g_i^a + g_i^b & g_i^a + g_i^b \geq 0 \\ 0 & g_i^a + g_i^b < 0 \end{cases} \tag{5}$$

A problem of fusing all of the these 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, we use a mapping from line directions to vectors such that the perpendicular line strengths become opposite to each other. Our reasoning is that 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 $s(p, \theta_i) \rightarrow \vec{s}(p, \omega_i) = s(p, \theta_i)e^{j2\theta_i}$ where $\omega_i = 2\theta_i$, $s(p, \theta_i)$ 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-perpendicular filters as $s(p) = \sum_{i=0}^{3} \vec{s}(p, \omega_i)$ for quadruple filters. The resulting vector is transformed to a line by halving the phase component.

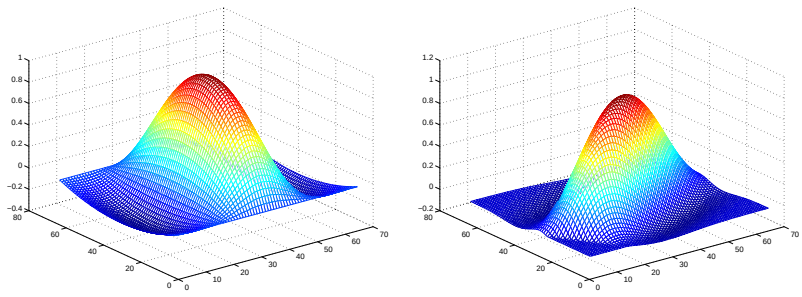

Figure 2. Gaussian-based road model and its rotated version.

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

$$m(p) = \frac{|\vec{s}(p)|}{|\vec{s}_{max}|}, \quad \theta(p) = \frac{\angle \vec{s}(p)}{2}. \tag{6}$$

where $\vec{s}_{max}$ is the maximum line strength computed for the current image.

2.3. Gaussian-Based Road Model

A major problem of road extraction algorithm is the disconnected road segments due to the poor visibility of the roads in the original image. Often roads are divided into several short segments, or completely missing from the image. For such cases, the possible road points that are not perceptible by the line filters should be enhanced using both the underlying image intensity distribution and the line detection results. We fit a Gaussian based model (Fig.2) at each image point to represent the likelihood of being a road point. These models are evaluated recursively to determine the correlation between the neighboring points. At each iteration, the local neighborhood of the point is searched for possible road models that match the current model. The model confidence is amplified if there is a neighboring road model at the same orientation, similarly in case no such neighbor exists the current model confidence is attenuated. The road model is defined as

$$r(p) = (1 - \frac{x_r^2}{M^2})(e^{-\frac{2|y_r^2|}{N^2}})(\cos\frac{\pi y_r}{N})m(p) \tag{7}$$

where M, N is the window size, and $x_r = x\cos\theta(p) + y\sin\theta(p), y_r = -x\sin\theta(p) + y\cos\theta(p)$. Then the models are simply updated using the K neighboring points models within the window L_p as

$$r^k(p) = \frac{1}{K}\sum_{q\in L_p} r(q). \tag{8}$$

At each update, the current road models are extended along their orientations if there is even a weak line model present at the extension points. The update operation is repeated until the missing road segments at a certain preset length can be recovered.

3. ROAD SEGMENTS AND FUSION

3.1. Connected Component Analysis

Given the confidences, the road points that have a relatively higher probability are linked into curvilinear segments. A common two-pass connected component analysis is designed such that it can recursively link line points possessing similar characteristics.

3.2. Irregularity Removal

The road segments generated by the connected component analysis may be contaminated by topological irregularities such as spurs and closed loops. Before feeding back the road segments to the input image, these deviations are removed by the morphological operators. In addition, thinning of road segments is required since the connecting stage may produce wide road segments. A morphological operator using the hit-and-miss transform[8] is utilized to accomplish the thinning of the road map. Like other morphological operators, the behavior of the thinning operation is determined by its structuring elements that are given as

$$T_1^{thin} = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline & & \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad \ldots, \quad T_8^{thin} = \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & & 0 \\ \hline & 0 & 0 \\ \hline \end{array}. \tag{9}$$

Thinning is done by translating the origin of the structuring element T_i^{thin} to each point position in the binary road image, and then at each position we compare the structuring element with the underlying image points. A total of eight comparisons is performed for all eight clockwise rotated versions of the original structuring element. During the iterative passes of the algorithm, if the foreground (1) and background (2) points in the structuring elements exactly match the foreground (road) and background (non-road) points in the road map, then the point underneath the origin of the structuring element is set to the background (non-road). Otherwise it is left unchanged. Thinning is repeated until there is no change detected.

$$T_1^{spur} = \begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad \ldots, \quad T_8^{spur} = \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 1 & & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}. \tag{10}$$

Spurs are mainly caused by the noise in the input image and are removed by using modified thinning operators. Using a modified structuring element T_i^{spur} in the thinning algorithm causes an erosion from the start points of the segments. The structuring elements are applied in a similar way as explained above, but the points are only marked. A marked road segment is considered a spur if the total number of points in the segment is much larger than the length of the main chord of the segment. This constraint distinguishes straight segments from the curved ones. Before the feedback stage, the shorter road segments and the segments that are considered to be spurs are removed from the road map.

3.3. Fusion and Relevance Feedback

The initial connected component analysis 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 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 fused with the input image, the directional line filters are applied to the fused image, and new road confidences and orientations are computed as described in the previous section. At an image point, linear weighting blends the magnitude values of the road map and the previous image intensity values as

$$I^t(p) = (1-\alpha)I^{t-1}(p) + \alpha m(p) \tag{11}$$

and the new orientation is computed after passing the result image from the filter set again. The fusion loop is executed until the change between the extracted road segments at two consecutive iteration becomes insignificant.

4. RESULTS AND CONCLUSION

We tested our road extraction algorithm on panchromatic images. Sample images are presented in Fig. 3-a. In these very-low resolution satellite images, the average road width corresponds to 1-3 pixels, and the images are contaminated by speckle noise. Since the dynamic intensity range is very low, the input images are required to be normalized to warp the intensity values to the 2^8 levels. We used 4 line filters operating at $0, \pi/4, \pi/2, 3\pi/4$ to compute the line strength and orientation. The line filter window size was set to 5×5 due to the narrow structure of the roads. In case of detection wider road segments, the original image may be subsampled such that the average road width becomes 1-3, or the line filter window is enlarged accordingly. Although the first approach is computationally faster, it is also more sensitive to the sampling impairments such as roads becoming disconnected. The road model window is set to 5×5 similarly. The model update mechanisms is executed twice to fill the gaps of 5 pixels long. After spurs and small loops are removed, the road segments are blended into the previous image by weighting by $\alpha = 0.05$. The enhanced images at the second iteration are also shown in the Fig. 3-b. The intermediate road detection results of the feedback loop are shown in Fig. 3-c. The final results are shown in Fig. 3-d. The fusion loop was iterated 15 to 20 on average times depending to the density of the roads in the input image. As visible in the enhanced images, the feedback mechanism significantly improved the visibility of the road segments in the original images, and the proposed road models achieved to recover most of the road gaps due to poor intensity range.

For the presented sample images and the other SPOT test images, no fine-tuning of the filter, road model, fusion, or other system parameters was necessary. The proposed algorithm extracted the most visible road segments accurately without requiring additional user interaction. The localization of the road segments are also more precise than the existing parametric line/curve fitting approaches since our method strictly traces roads instead of approximating them.

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. F. Porikli and T. Keaton, "Unsupervised road extraction algorithm for very low-resolution satellite imagery", *International Conference of Pattern Recognition Remote Sensing*, 2000.
7. J. Trinder and Y. Wang, "Automatic road extraction from aerial images", *International Workshop on Image Analysis and Information Fusion*, 1998.
8. B. Jang and R. Chin, "Analysis of thinning algorithms using mathematical morphology", *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. 12, 1990.
9. G. Vanderburg, "Line detection in satellite imagery", *IEEE Trans. on Geoscience Electronics*, Vol. 14, 1976.
10. M. Barzohar and D. Cooper, "Automatic finding of main roads in aerial images by using geometric-stochastic models", *Pattern Recognition and Machine Intelligence*, Vol. 7, 1996.
11. V. Cleynenbreguel and J. Osinga, "Road extraction from multi-temporal satellite images by an evidential reasoning approach", *Pattern Recognition Letters*, Vol. 12, 1991.
12. N. Netenyahu and V. Philomin, "Robust detection of straight and circular road segments in noisy aerial images", *Pattern Recognition*, Vol. 10, 1997.
13. B. Ma and S. Lakshmanan, "Detection of curved road edges in radar imaes via deformable templates", *International Conference of Image Processing*, 1997.

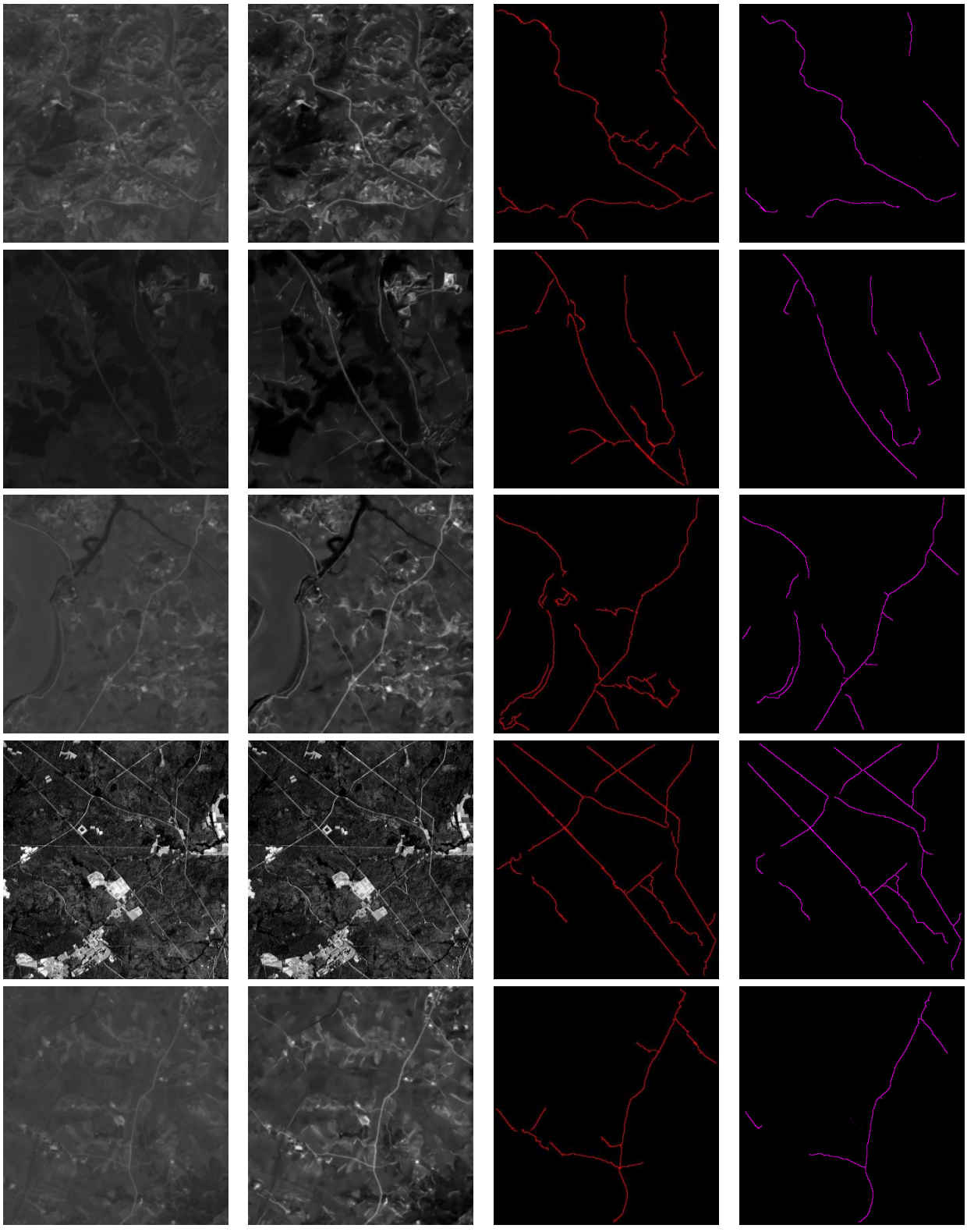

Figure 3. (a) Sample input images. (b) Enhanced images at the 2^{nd} iteration of the fusion mechanism. (c) Detected road segments at 4^{th} iteration. (d) Final road segments.