

AUTOMATED CORRECTION AND UPDATING OF ROADS FROM AERIAL ORTHO-IMAGES

Marie-Flavie AUCLAIR FORTIER¹, Djemel ZIOU¹, Costas ARMENAKIS² and Shengrui WANG¹

¹Département de mathématiques
et informatique
Université de Sherbrooke
2500 Blv Université, Sherbrooke
Québec, Canada, J1K 2R1
auclair@dmi.usherb.ca

² Center for
Topographic Information
Geomatics Canada
615 Booth Str, Ottawa
Ontario, Canada, K1A 0E9
armenaki@NRCan.gc.ca

KEY WORDS: Road Database Correction, Aerial Image Analysis, Snakes, Line Junctions, Road Following.

ABSTRACT

Our work addresses the correction and update of road map data from georeferenced aerial images. To positionally correct the existing road network location from the imagery, we use an active contour ("snakes") optimization approach, with a line enhancement function. The initialization of the snakes is based on the existing vector road data coming from the National Topographic Database of Geomatics Canada, and from line junctions computed from the image by a new detector developed for this application. To generate hypothesis for new roads, a road following algorithm is applied, starting from the line intersections, which are already in the existing road network. Experimental results are presented to validate the approach and to demonstrate the interest of using line junctions in this kind of applications.

1 INTRODUCTION

Road detection is a time consuming operation when performed manually. Fortunately, aerial imagery tends to facilitate this process, but complete automatization is still far off with the methods used to date. Road detection is a type of feature extraction which is almost always context-dependent. In our case, high-resolution georeferenced images, that is, the geographic position of each pixel is known, and the existing road topographic network from the National Topographic Database of Geomatics Canada are used. There are two problems with this database: 1) it is not accurate enough (Figure 2a) and 2) new roads are not included. The aim of our work is to automatically correct and update this road network.

Much works has been done on automating road detection (Auclair-Fortier et al., 2000, Guindon, 1998, Heipke et al., 1997). Among the methods developed, we can mention particularly the work of Klang (Klang, 1998). He developed a method for detecting changes between an existing road database and a satellite image. First he used the road database to initialize an optimization process, using a snake approach to correct road location. Then, he ran a line-following process using a statistical approach to detect new roads, starting from the existing network. It should be noticed that none of the reviewed methods make use of junctions in the image. These junctions are generally reliable information and since roads form networks, they are very relevant in this context. There are significant differences between our approach and Klang's. First, we add line junctions to his correction scheme to improve the matching between the road database and the image. Second, we generate hypothesis for new roads by following lines starting from line junctions near the known road network. We describe our approach in Section 2. In Section 3, we present line and line-junction detection. In Section 4, we describe the correction of road location. In Section 5, we present a hypothesis generation scheme for new roads. In Section 6, we present results obtained from a georeferenced image and a road database, both supplied by Geomatics Canada. Finally, the conclusions are discussed in Section 7.

2 APPROACH

Geometric correction of the database is the first part in updating road maps. Because the road database has been obtained from various sources including scanning road maps, the location of roads is not precise. However, this information allows a good distinction between image features which are roads and those which are not. In aerial images, roads are generally long, thin features, built in networks. In the case where the resolution is not low enough to have roads as lines, the image is resampled to reduce its resolution. Then, we want to find lines which are near the initialization location provided by the road database. This prompts us to use active contour models (snakes), which are optimization curves requiring an initialization close to the solution. To bring this initialization closer to the solution, we match each intersection in the road database with at most one junction found in the image. Thus, we re-localize road segments according to the difference between database intersections and image junctions. After re-localization, we apply a line optimization process, using the

snake approach. The correction part can be summarized by these five steps: 1) resampling the image 2) extraction of lines 3) extraction of line junctions 4) displacement of road segments according to the difference between their extremities and image junctions matched, and 5) correction of road segment location with snake process.

The second step in updating road maps is the addition of new roads. We know that roads are built in networks and that new roads are linked with old ones. We thus generate hypothesis for new roads by following lines from line junctions near the known network.

3 DETECTION LINES AND OF LINE JUNCTIONS FROM THE IMAGE

In Aerial and Satellite images, roads are long, thin elements. Usually, they are bright lines against a dark background. To detect these, we use Steger's line detector (Steger, 1998) which can be summarized as follows: A bright line is a maximum in the gray-level image in a particular direction. Thus, in this direction, the second directional derivative has a negative minimum and its first directional derivative is 0. This detector provides a line plausibility ($\lambda_{max}(x, y)$) and a line direction ($\vec{n}(x, y)$) function.

In the context of road detection, road intersections are represented in aerial images by line junctions. Image junctions are very useful features, particularly in solving matching problems, because they are generally reliable and stable elements. Unfortunately, we have noticed that they are not used for road detection. One reason is that, to the best of our knowledge, before the beginning of this project, no line-junction detector was existing. We use a line-junction detector developed in our Computer Vision Laboratory, based on a measure of curvature between the direction vectors of the line pixels within a given neighborhood (Deschênes and Ziou, 1999). This detector also finds line terminations. This measure is estimated by computing the mean of dissimilarity between the orientation vector of the line pixels:

$$\frac{1}{N} \sum_{\Delta x=-M}^M \sum_{\Delta y=-M}^M \beta(\Delta x, \Delta y) [1 - |\vec{n}(x, y) \cdot \vec{n}(x + \Delta x, y + \Delta y)|] \quad (1)$$

where $\vec{n}(x, y)$ is the normalized line direction at point (x, y) , \cdot is the dot product,

$$\beta(\Delta x, \Delta y) = \begin{cases} e^{-(\Delta x^2 + \Delta y^2)} & \text{if a line exists between } (x, y) \text{ and } (x + \Delta x, y + \Delta y) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and N the number of points $(x + \Delta x, y + \Delta y)$ such that $\beta \neq 0$. The junction is localized in a high curvature region using a non-maxima suppression. For more details, the interested reader can see the technical report on this detector (Deschênes and Ziou, 1999). Figure 1a presents the results of the line-junction detector.

4 CORRECTION OF ROAD LOCATION

The correction of road location involves two steps. The first is re-localization of the known roads to bring them closer to the final snake's solution. The second step concerns the snake process itself. These two steps are presented in the next two sections.

4.1 Matching the Road Database with Image

The active contour model (snake) is an iterative process which needs an initialization close to the final solution. This initialization must fall within the attraction zone of the snake. We propose to use line junctions to re-localize road location from the road database, because they indicate the position of road intersections in the image. For this purpose, we use a correlation measure between certain points in the road database and line junctions. The road database provided by the Center for Topographic Information of Geomatics Canada is made up of lists of georeferenced points defining line segments. We call these line segments vectors. These points are in georeferenced coordinates, thus we have to convert them in coordinates relative to the beginning of the image. Then, we group the vectors into road segments in order to reduce the number of snakes. These road segments must be as straight as possible because this allows us to add a constraint of high rigidity to the snake. This high rigidity will help avoid noise and shadowing problems, because the optimized contour does not follow small intensity variations caused by these phenomena. Two vectors are linked if they are adjacent, if the cosine of the angle between them is below a threshold (in our experiments, we always set this threshold to $\cos 138^\circ$) and if there is no other vector starting at the extremity common to the two vectors. In other words, we find road segment extremities 1) where there are more than two vectors ending at a single point 2) where the angle between two vectors is lower than a given value 3) at terminations. A road segment is described by all vectors between its two extremities. Figure 1b shows an example of road segments from the database. There are five road segments in this image.

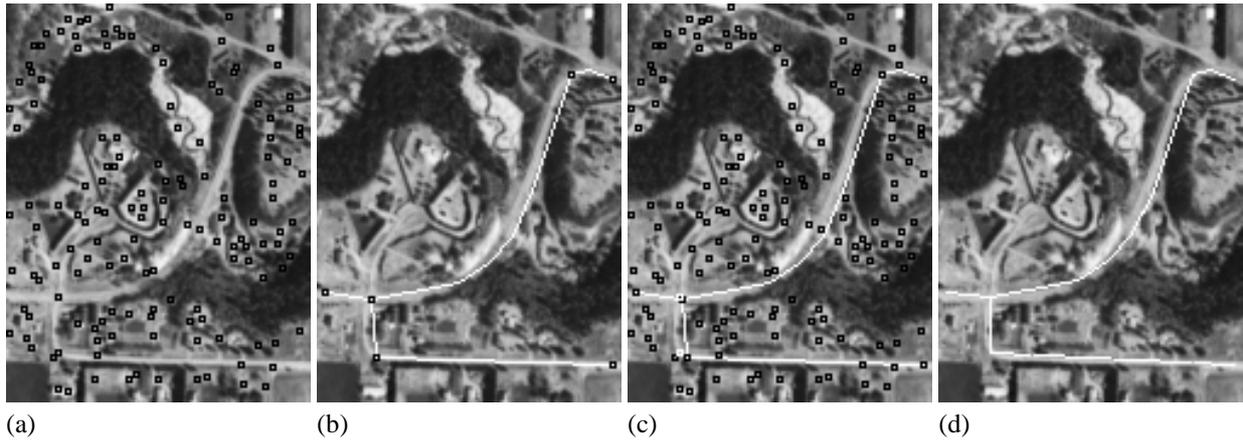


Figure 1: a) Line junctions. b) Road segments. Black squares are extremities. c) Matching between image junctions and road-segment extremities. White squares are junctions matched with extremities, while black squares are either extremities or other junctions ($N_m = 13$). d) Road segments after re-localizing the two extremities matched with junctions.

Having the list of all road segments, each extremity is matched with a junction, if possible. For this purpose, a correlation measure is computed between the extremity and each junction localized in a $N_m \times N_m$ neighborhood around this extremity. This measure is done on a $M_m \times M_m$ neighborhood around each such junction. Let (x, y) be the extremity and (u, v) be the junction; then, the correlation is:

$$\text{corr}(x, y, u, v) = \sum_{i=-M_m/2}^{M_m/2} \sum_{j=-M_m/2}^{M_m/2} \text{RoadSeg}(x+i, y+j) \text{line}(u+i, v+j), \quad (3)$$

where

$$\text{RoadSeg}(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ belongs to a road segment} \\ 0 & \text{if not.} \end{cases} \quad (4)$$

and $\text{line}(x, y)$ is the line plausibility (λ_{max}) as defined in Sect. 3. (x, y) is matched with the junction (u, v) having the highest correlation. Figure 1c presents the results of matching between extremities of Figure 1b and junctions of Figure 1a. We used a M_m of 25 in all of our trials.

After matching, each segment re-localized according to the difference between its extremities and the junctions matched. As the differences at the two extremities may not be the same, each vector extremity in the road segment is moved according to the difference between the nearest road segment extremity and its matched junction. That means that every vector keeps the same orientation except the one containing the midpoint of the road segment, because its two extremities will move according to the two different extremities of the road segment Figure 1c presents road segments after re-localization.

4.2 ACTIVE CONTOUR MODELS - SNAKES

As we mentioned earlier, we want to find lines near the initialization provided by the road database. Active contour model (snake) is an optimization process which needs an initialization near the solution (Berger, 1991, Kass et al., 1988). Aerial images can be affected by noise and roads can be partially hidden by shadows of trees, houses or clouds. Snakes manage these problems with constraints on rigidity and elasticity.

In the continuous domain, snakes (or active contours) are parametric curves $(v(s, t) = (x(s, t), y(s, t)), s \in [0, 1])$ that minimize an energy function based on internal and external constraints (Berger, 1991) at time t :

$$E_{tot} = \eta E_{int} + E_{ext} = \eta \int_0^1 (\alpha_1 |v'(s, t)|^2 + \beta_1 |v''(s, t)|^2) ds + \int_0^1 \lambda_{max}(v(s, t)) ds, \quad (5)$$

where E_{ext} is the external energy, E_{int} the internal energy and η a regularisation parameter.

The internal energy represents the snake's capacity to distort. $\alpha_1, \beta_1 \in \mathbb{R}^+$, v' and v'' are the first and second partial derivatives in s . The term $|v'(s, t)|^2$ influences the snake's length (or tension) and $|v''(s, t)|^2$ influences the curvature (or elasticity) of the snake. The parameter η can be absorbed by α and β ($\alpha = \eta\alpha_1$ and $\beta = \eta\beta_1$).

The external energy represents the image force ($f(x, y)$) attracting the snake. In our case, this force is the bright-line plausibility ($f(x, y) = -\lambda_{max}(x, y)$).

Equation 5 is an evolution problem (snake position evolves over time). The solution to this minimization problem is the simplified Euler evolution equation:

$$\gamma \frac{\partial v(s, t)}{\partial t} - \alpha v''(s, t) + \beta v''''(s, t) = -\frac{\partial E_{ext}(v(s, t))}{\partial v}, \quad (6)$$

where v'''' is the fourth derivative in s . The parameter γ represents the viscosity of the curve. The higher the viscosity, the greater the inertia and the slower the evolution of the curve.

The discrete version of Equation 6 is obtained by setting a discrete version of the parametric curve at moment t : $\mathbf{V}_t = \{v_{i,t} = (x_{i,t}, y_{i,t}), i = 1..n\}$, where n is the number of points in the curve. The solution can be written in the following vectorial form:

$$\gamma \frac{\partial \mathbf{V}_t}{\partial t} - \alpha \mathbf{V}_t'' + \beta \mathbf{V}_t'''' = -\frac{\partial \mathbf{E}_{ext}(\mathbf{V}_{t-1})}{\partial v}, \quad (7)$$

which can be rewritten in matrix form:

$$(\mathbf{K} + \gamma \mathbf{I}) \mathbf{V}_t = \gamma \mathbf{V}_{t-1} - \frac{\partial \mathbf{E}_{ext}(\mathbf{V}_{t-1})}{\partial v}, \quad (8)$$

where I is the $n \times n$ identity matrix and \mathbf{K} is a $n \times n$ matrix. The elements of \mathbf{K} are determined by Equation 7 and the initial conditions; the snake has free extremities (Berger, 1991), i. e. $v''(0, t) = v''(n, t) = v''(1, t) = v''(n, t) = 0$. γ is given by (Berger, 1991):

$$\gamma = \frac{|\frac{\partial \mathbf{E}_{ext}}{\partial v}|}{\sqrt{n\Delta}}, \quad (9)$$

where Δ is the mean progression of the curve. The initial Δ must be determined by the user. We keep the same Δ until the total energy of the curve increases and then we decrease Δ by a factor of 0.75. We can approximate the total energy of the curve at time t by:

$$E_{tot,t} = \sum_{i=1}^n (E_{ext}(v_{i,t}) + \alpha |v'_{i,t}|^2 + \beta |v''_{i,t}|^2) \quad (10)$$

To summarize the correction step, for each road segment:

1. Initialize a snake curve (\mathbf{V}_0) with the re-localized road segment found in the matching step. Each pixel belonging to the curve is a $v_{i,0}$. Set t to 1.
2. Compute \mathbf{V}_t from Equation 8.
3. If the total energy of the curve decreases then increment t and go to step 2.
4. If Δ is not too low, decrease Δ , increment t and go to step 2.

5 GENERATION OF HYPOTHESIS FOR NEW ROADS

As mentioned earlier, the second problem in updating road database is the detection of new roads. These new roads are also bright lines on a dark background, so the result of the line detector can be used.

Our work is based on the supposition that new roads are connected with the existing road network. This hypothesis is realistic because roads are built to allow people to travel from one place to another. For this reason, we consider only lines connected to the road network as given in the existing database. The line-following algorithm starts from junctions in a $N \times N$ neighborhood of the roads, corrected using the snake approach (Figure 4a). We follow all lines originating from these junctions. We consider each line pixel in a 5×5 window around a junction as a starting pixel for the line-following process.

For each line, we proceed as follows: Initially, we take one of the starting pixel (anyone). Let p be the last chosen pixel (we call it the parent). First, we consider three neighbors of this pixel in the line direction. For example, if the line

direction at point $p = (p_x, p_y)$ is in $]\pi/8, 3\pi/8]$, the three neighbors to consider are $(p_x + 1, p_y + 1)$, $(p_x, p_y + 1)$ and $(p_x + 1, p_y)$. Each of these three neighbors which is marked as a line pixel is considered as a child. The next pixel selected is the child that minimizes:

$$|\theta_c - \bar{\theta}_p| \quad (11)$$

where θ_c is the line direction of the child and $\bar{\theta}_p$ is the mean line directions of the parent respectively. The mean line direction is computed on the last L pixels visited, as follows:

$$\bar{\theta}_a = \frac{\sum_{i=0}^{L-1} \theta_{a-i}}{L}, \quad (12)$$

where θ_{a-i} is the line direction of the i th pixel visited before pixel a .

In the case where none of the three neighbors are marked as line pixels, we look in a neighborhood further away in the line direction; that is, when the direction of the vector between the parent and the pixel falls in the interval $[\theta_p - \pi/8, \theta_p + \pi/8]$, we consider this pixel as a child. In this case, the next pixel is the one minimizing:

$$(1 - \omega) \|pt_c - pt_p\|_2 + \omega |\theta_c - \bar{\theta}_p| \quad (13)$$

where pt_c and pt_p are the locations of child and parent, respectively and $\omega \in [0, 1]$ is a weighting parameter. We do it iteratively from the beginning with the retained child. If a parent has no child, the line-following process is terminated. If the new road is shorter than a given threshold (S), it is dropped.

Let a line junction near the existing network, for each line pixel in a 5×5 neighborhood of this junction, we can summarize the line following algorithm as follows:

1. Set the pixel as parent.
2. Compute the three neighbors in line direction of the parent. If they are line pixels, set them as children.
3. If there is at least one child:
 - (a) Set the child that minimizes Equation 11 as parent.
 - (b) Return to step 2.
4. If there is no child:
 - (a) Compute further children in the line direction.
 - (b) If there is now at least one child:
 - i. Set the child that minimizes Equation 13 as parent.
 - ii. Return to step 2.
5. If the length of the new line is less than S , then drop it.

Figure 4b shows the results of the line-following process with $L = 10, S = 7, \omega = 0.9$.

6 EXPERIMENTAL RESULTS

Our algorithm has been implemented in C++ on a Sun platform and in Visual C++ on a Windows platform. It was experimented on various sections of a georeferenced ortho image (11334×10166 pixels) that we re-sampled in order to reduce the initial spatial resolution by a factor of two. The digital georeferenced ortho image mosaic used has initial ground spatial resolution of 1.5m and is from the Edmonton, Alberta area. It was generated by scanning 1 : 60000 scale 1995 aerial photography at 1000ppi (0.025mm) with 8-bit quantization. The Helava/Leica digital photogrammetric workstation was used for the automatic DEM generation, while the exterior orientation parameters for each image were determined through automatic aerotriangulation using control points from the Aerial Survey Database (ASDB). The road vector data were obtained from the National Topographic Database (NTDB). These data were revised and enhanced in 1996 using GPS methods and have an estimated accuracy of about 10m.

Figure 2 presents the results obtained at each step of our correction algorithm for an image section. Figure 2a is the original image, on which the vectors from the road database are superimposed. Visually, we can see that in many places,

vectors do not correspond to roads. Figure 2b shows junctions (threshold is 0.1) and road-segment extremities, represented by black squares. The white squares are junctions matched with extremities ($N_m = 13$). We note that except for dead ends, there is only one extremity which is not matched (upper part of central curve), because there is no junction in the neighborhood of the extremity. If the nearest junction had been matched, the central segment would have shifted to the right, away from the road. Figure 2c shows the initialization after re-localizing road segments. Figure 2d presents the final position of the snakes ($\alpha = 100$ and $\beta = 4$). We note that the final snake positions are closer to the center of the lines than the original vectors. The mean displacement of road segment points is 2.76 pixels and the variance of this displacement is 4.09. We had a mean number of iterations of 31 per pixel in the snake step, for a total of 641 pixels treated.

The results without junction detection (Klang's approach with the same parameters as in Figure 2) are presented in Figure 3. The mean displacement of road segment points is 3.63 and the variance of this displacement is 4.95. The mean number of iterations was 34 per pixel. Comparing Figure 2f and 3b, we can see why the mean displacement in Klang's approach is higher than ours. Some snake sections are displaced further from the road. Our results shown an improvement in the road's localization over Klang's results.

Now we will present the results concerning generation of hypothesis for new roads. Figure 4 presents the new-road hypothesis generation step for the image in Figure 2a. Figure 4a shows the line junctions near the known network ($N = 7$) and Figure 4b shows the final result ($L = 10$ and $S = 7$). The interested reader will find more results and a study of the parameters in (Auclair-Fortier et al., 1999).

7 CONCLUSION

We have presented our approach for automatically updating road maps from high-resolution aerial images. The updating of road database involves the addressing of two problems, i.e., the correction of existing road location and the addition of new roads. Since roads are long and thin elements in high-resolution images, the method is based on the detection of line. Existing road location is not precise, but can provide a good indication of where roads are located as opposed to other elements such as rivers or railways. For correction, we use a snake approach, with the existing location as the line initialization. To bring the initialization closer to the road, we re-localize road segments on the basis of line junctions from the image. We include a line junction detector proposed by (Deschênes and Ziou, 1999). We have presented some results obtained with an image and a database provided by Geomatics Canada. For the addition of new roads, we apply a line-following algorithm starting from image junctions which are in the existing network. We have shown that there is an advantage for using line junctions by comparing with Klang's approach.

REFERENCES

- Auclair-Fortier, M.-F., Ziou, D., Armenakis, C. and Wang, S., 1999. Automatic Correction and Updating of Road Databases from High-Resolution Imagery. Technical Report 241, Département de mathématiques et d'informatique, Université de Sherbrooke.
- Auclair-Fortier, M.-F., Ziou, D., Armenakis, C. and Wang, S., 2000. Survey of Work on Road Extraction in Aerial and Satellite Images. Technical Report 247, Département de mathématiques et d'informatique, Université de Sherbrooke.
- Berger, M.-O., 1991. *Les contours actifs : modélisation, comportement et convergence*. Thèse de Doctorat, Institut National Polytechnique de Lorraine.
- Deschênes, F. and Ziou, D., 1999. Detection of Line Junctions and Line Terminations Using Curvilinear Features. Technical Report 235, Département de mathématiques et d'informatique, Université de Sherbrooke.
- Guindon, B., 1998. Application of Spatial Reasoning Methods to the Extraction of Roads from High Resolution Satellite Imagery. In: IGARSS.
- Heipke, C., Mayer, H., Wiedemann, C. and Jamet, O., 1997. Evaluation of Automatic Road Extraction. Proceedings of International Archives of Photogrammetry and Remote Sensing XXXII, pp. 47–56.
- Kass, M., Witkin, A. and Terzopoulos, D., 1988. Snakes : Active Contour Models. The International Journal of Computer Vision 1(4), pp. 321–331.
- Klang, D., 1998. Automatic Detection of Changes in Road Databases Using Satellite Imagery. In: Proceedings of International Archives of Photogrammetry and Remote Sensing, Vol. 32, pp. 293–298.
- Steger, C., 1998. An Unbiased Detector of Curvilinear Structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(2), pp. 113–125.

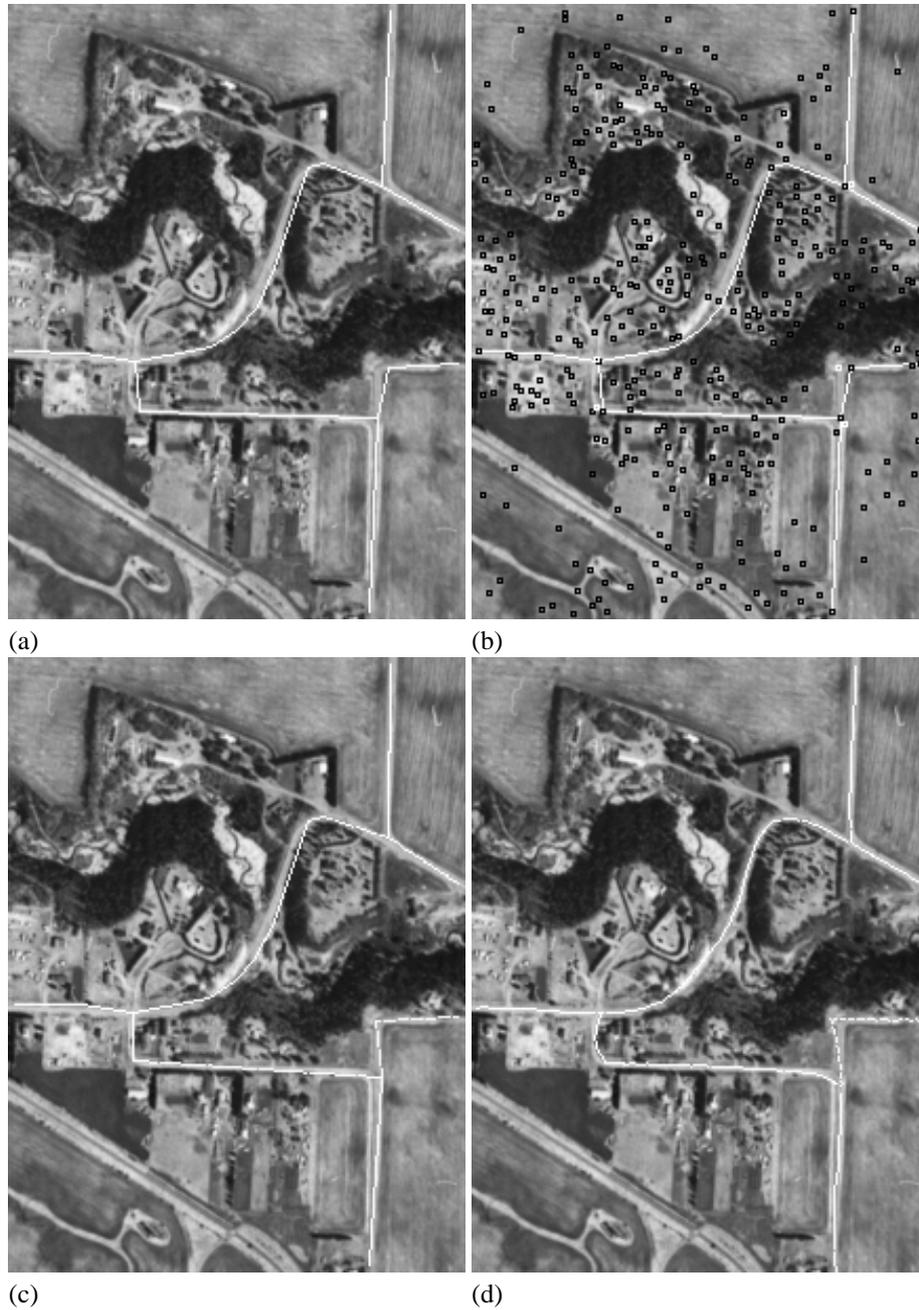


Figure 2: a) Original image (217×295) with vectors from the road database. b) Junctions (threshold is 0.1) and extremities (black squares). White squares are matched junctions ($N_m = 13$). c) Initialization of snakes after re-localization. d) Final result ($\alpha = 100$ and $\beta = 4$).

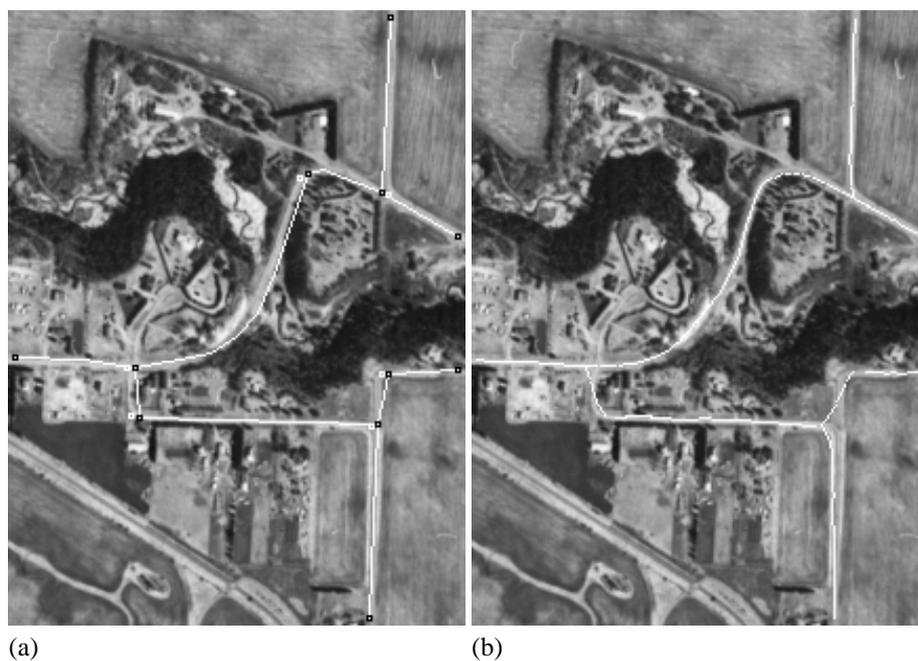


Figure 3: Results without junction detection. a) Matching of road segments with image. White squares are points matched. b) Final result ($\alpha = 100$ and $\beta = 4$).

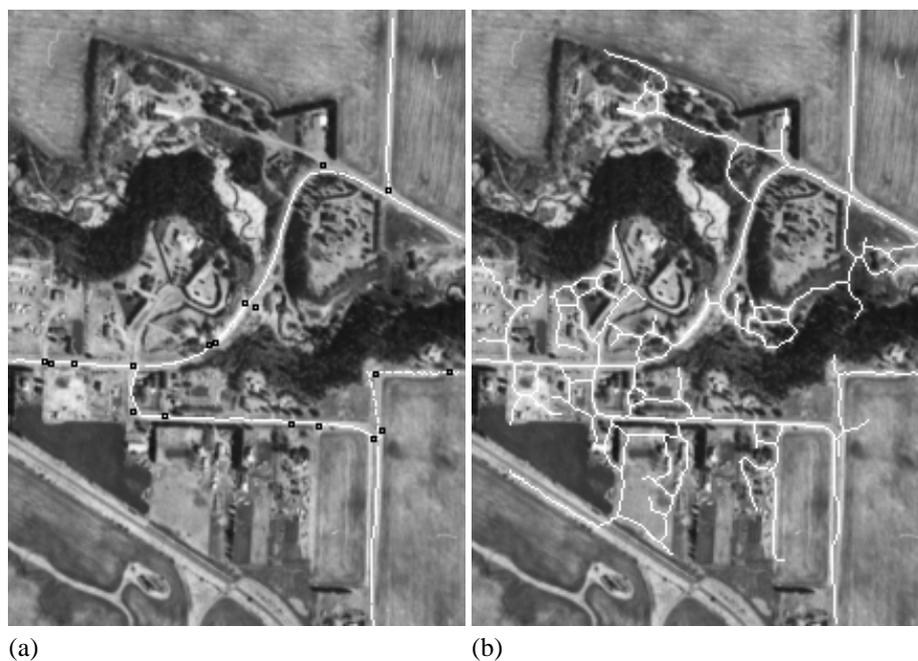


Figure 4: Detection of new roads. a) Line junctions near the corrected network. b) Result ($L = 10$ and $S = 7$).