Road Centerline Tracking From High Resolution Satellite Imagery By Least Squares Templates Matching

Seung-Ran Park, Taejung Kim, Soo Jeong**, Kyung-Ok KIM**
Satellite Technology Research Center, KAIST, 373-1 , Yusung-gu, Taejon, 305-701, ROK
Tel : (82)-42-869-8634 Fax : (82)-42-861-0064
E-mail : {srpark,tjkim}@satrec.kaist.ac.kr, URL:http://satrec.kaist.ac.kr

** Spatial Information Technology Center ETRI,161 Gajeong-Dong, Yuseong-Gu, Daejeon, 305-350, ROK
Tel: (82)-42-860-6652, Fax: (82)-42-860-4844
E-mail : {soo,kokim}@etri.re.kr

ABSTRACT: Road information is very important for topographic mapping, transportation application, urban planning and other related application fields. Therefore, automatic detection of road networks from spatial imagery, such as aerial photos and satellite imagery can play a central role in road information acquisition. In this paper, we use least squares correlation matching alone for road center tracking and show that it works. We assumed that (bright) road centerlines would be visible in the image. We further assumed that within a same road segment, there would be only small differences in brightness values. This algorithm works by defining a template around a user-given input point, which shall lie on a road centerline, and then by matching the template against the image along the orientation of the road under consideration. Once matching succeeds, new match proceeds by shifting a matched target window further along road orientation at the target window. By repeating the process above, we obtain a series of points, which lie on a road centerline successively. A 1m resolution IKONOS images over Seoul and Daejeon were used for tests. The results showed that this algorithm could extract road centerlines in any orientation and help in fast and exact head-up digitization/vectorization of cartographic images.

KEY WORDS: template matching, least squares correlation matching, road extraction, feature extraction

1. Introduction

Many authors have studied automated extraction of curvilinear features such as roads, railroads, and river boundaries [1] – [8]. In order to achieve full automation, various techniques have been tried. They include the combination of local analysis of features (such as edge and lines) and perceptual grouping [1][2][5][7], scale-space approaches [6], top-down approaches based on Divide-and-Conquer [4] and the use of neural network and classification [8].

However, full automation in extraction of curvilinear features is yet to be achieved. Often, fully automated algorithms produce incomplete results so that manual post processing is inevitable. They are somewhat lack of robustness in the sense that they often produce huge failures on unpredicted input.

The motivation of our work in this paper is to develop a methodology to generate a road map from high-resolution remote sensing images at around 1m ground sampling distance. The intention was to develop the methodology to replace head-up digitization of road centerlines. Robustness was given a highest priority. For this, we decided to include human interactions to delineate road centerlines. We instead focused on how to reduce human interactions for the task while assuring meaningful extraction of the centerlines.

2. Observation of Road Images In High-Resolution

As mentioned previously, our intension is to delineate road centerlines from satellite or aerial images at around 1m resolutions. Figure 1 shows road images at such a resolution over a typical dense metropolitan city of Seoul.

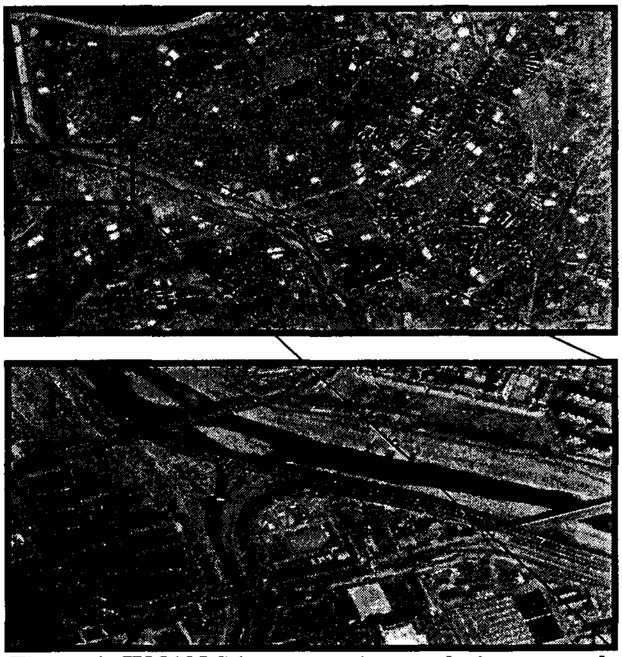

Figure 1. IKONOS images at 1m resolutions over the dense metropolitan city of Seoul in a small scale (upper image) and in a larger scale (lower image). (Courtesy of e-HD.com, Ltd.)

The first thing we can notice is that there are many different types of road: highways, interchanges, main streets with many vehicles, small roads between houses, etc., in various orientations. In most of large roads, centerlines are clearly visible and are appeared as curvilinear features. At 1m resolutions, the problem of road extraction can be replaced by the problem of road centerline delineation. Although there are noises, the brightness patterns of a road along its centerline appear quite similar.

We can define a part of a road with similar brightness patterns as a "road segment". Within a road segment, template matching may work. In other words, if we define a template on a certain point of a road centerline, other parts of the centerline within the same road segment can be matched against this template.

3. Algorithm Description

3.1. Adoption a Least Squares Correlation Matching

In real world, the surface of a road is in general flat. Therefore in road images at 1m resolutions, there are not severe geometric distortions around and along road centerlines. If we draw a rectangular window centered on a point of a road centerline, we can define the corresponding window at another point of a road centerline by translating and rotating the rectangle (see figure 2).

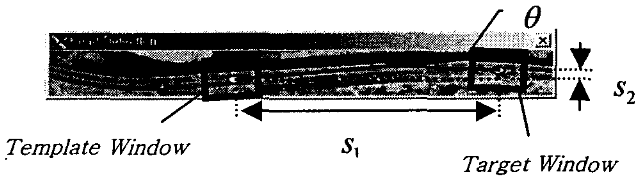

Figure 2. The relationship between the template and target windows

Here, we can model the relationship between a template window and target window as

$$x_{target} = x_{template}\cos\theta + y_{template}\sin\theta + s_1 - s\sin\theta_{template} \quad (1)$$

$$y_{target} = -x_{template}\sin\theta + y_{template}\cos\theta + s_2 + s\cos\theta_{template} \quad (2)$$

where ($x_{template}, y_{template}$) is a coordinate of a point on a template window, (x_{target}, y_{target}) of a point on a target window, θ the rotation angle between the template and target windows and (s_1, s_2) the shift in x and y directions. $\theta_{template}$ is the road orientation at the template point and s the shift distance along perpendicular to $\theta_{template}$. Here, s_1 and s_2 are set to indicate the distance between the template and initial guess. They remain constant through iteration. Least squares correlation matching assuming this similarity transformation can be achieved using the following matrix equation.

$$l = Ax$$

$$l = \left\{ f\left(x_{template}, y_{template}\right) - g\left(x_{target}, y_{target}\right) \right\}$$

$$X^T = \left[\Delta\theta \quad \Delta s \quad r_s\right]$$

$$A^T = \begin{Bmatrix} -(\delta_x \sin\theta + \delta_y \cos\theta)x_{template} + (\delta_x \cos\theta - \delta_y \sin\theta)y_{template} \\ -\delta_x \sin\theta_{template} + \delta_y \cos\theta_{template} \\ 1 \end{Bmatrix}$$

where f *is* the brightness value of the template window at ($x_{template}, y_{template}$), g of the target window at (x_{target}, y_{target}), δ_x and δ_y are the derivatives of brightness values at (x_{target}, y_{target}) in x and y directions, respectively and r_s is the difference of average brightness value between the template and target windows.

We can further improve the performance of least squares correlation matching for road centerline tracking by applying higher weights around the center of a template and target window and lower weights around the ridge. Figure 3 shows that the definition of initial target window around the position of initial guess and matched target window formed through iterative least squares correlation matching.

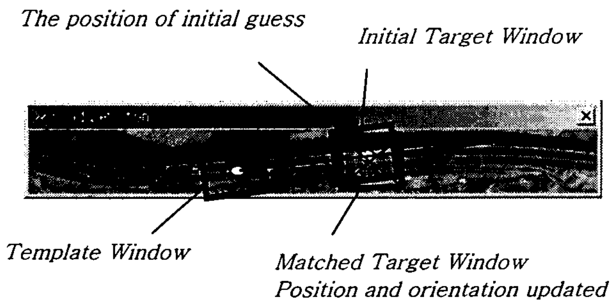

Figure 3. The definition of initial target window and matched target window

3.2. Execution Procedures

The procedures of our road centerline tracking algorithm are shown in Figure 4.

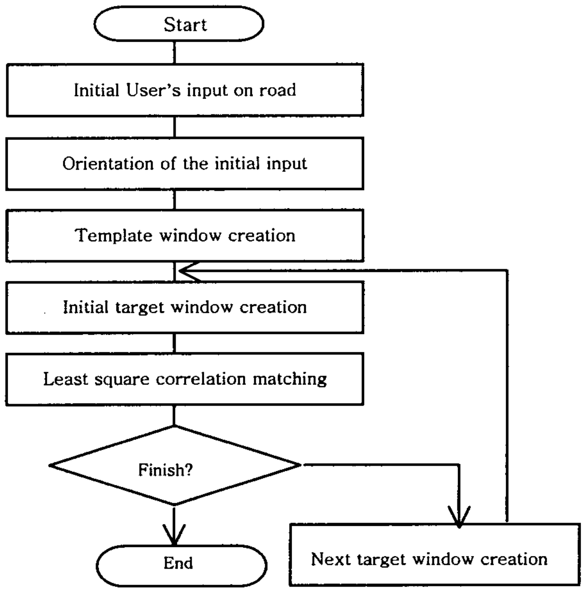

Figure 4. The procedures of road centerline tracking

First, an input point on a road centerline must be provided from a user. This point will be used as the center of a template window. It is important for a user to select a point on a road centerline in order to track valid road centerlines. This point will be discussed further in the next section. It may be possible to refine the input point by projecting it to a nearby linear segment, if extraction of linear segments has been applied beforehand. This idea, however, has not been used for our experiments.

The next step is to estimate the orientation of the road. The orientation of the road is important in our algorithm. It defines the orientation of the template and initial target window. More importantly, it defines the position of the initial target window and hence guides the direction of template matching. One of the major differences between our algorithm and previous work on template matching [3][9][10] is that others require additional constraints to guide matching for meaningful feature extraction whereas we guide matching only by the orientation information of match windows, which can be obtained through our least squares correlation matching.

We tried two methods for estimating the orientation of road at a user's input point: automatic and manual. The automatic method was to apply automatic line extraction method proposed by [11] to the image and then to calculate the orientation of the line segment nearby the initial input point. The manual method was to get another input point from a user and to calculate the orientation of the line connecting the two input points. Valid line segments were extracted and valid road orientations were estimated in most cases. However, in order to ensure our algorithm can work for all times, we need the manual alternative.

Based on the user input point on a road centerline and its orientation, a template and initial target windows are generated. A template window is defined whose center is at the user input point and whose orientation is aligned to the road orientation. An initial target window is generated by shifting the template window to the direction of road.

Least squares correlation matching is then applied. The position and orientation of target window is updated iteratively. Once matching is completed the location of a new road centerline and its orientation are achieved. Then our algorithm checks whether there are more points to consider or not. If so, a new target window is defined by shifting the matched target window further to the orientation of matched target window. The procedure of least squares correlation matching and the creation of new target window repeat. A series of least squares correlation matching generate successive points on a road centerline and in this way tracking road centerline is achieved through least squares correlation matching.

3. 3 Post Processing

Due to the nature of template matching scheme, one complete road is sometimes extracted as several road segments. To overcome this limitation, we have developed user-friendly post processing algorithm for connecting, deleting and editing road segments.

User can connect two split segments as one. Template matching sometimes splits one road segment into two or more segments. These segments are connected by user's manual selection and connected segments are saved as identical segment.

Occasionally, template matching exposes to matching fail points. In this case, a user selects a fail point, and then deletes them. A user can also edit a road segment by inserting and shifting additional match points to the segment.

4. Experiments And Discussions

The algorithm proposed here was tested with 1m resolution IKONOS image over Seoul and Daejeon area. Figure 5 shows two examples of road centerline extraction on a typical highway. In the first example, road orientation was estimated automatically by applying the line extraction algorithm proposed by [11]. Once a user point was given, a series of matching was applied rightwards and leftwards. It was set to stop tracking when least squares correlation matching failed (failure points are not shown in the figure). In order to match one point, it took 6.5 msec on average and to match the whole segment shown in the figure, 0.3 seconds on a Pantium-4 PC at 1.5GHz clock speed. The second example shows the road centerline extraction when a series of user inputs are given. Road orientation at each input point was estimated by the line connecting the point and one adjacent to it. In this case, we limited the matching to take place only between the input points.

This can be one way to control the beginning and end position of our road tracking algorithm.

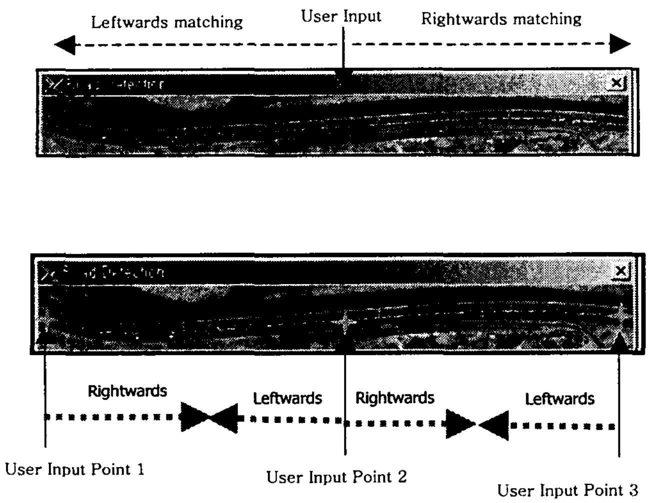

Figure 5. An example of road centerline extraction when one point is given and line extraction is carried out (upper image) and when a series of user input points are given (lower image).

Figure 6 shows the results of road extraction over the whole test area. In this example, an operator provided a series of input points and road orientation was estimated using the input points. Also matching was applied only between the input points. Extraction of road centerlines was carried out three times by different operators. On average 29 segments were required to track the whole roads as shown in the figure. Due to the small scale of the figure, each segment is not easily discernable. Within a segment, three to five input points were given on average, although the exact number varied between operators. Computation time taken for matching was 21.2 seconds on average. If we assume an experienced user, the total time taken to produce such a result including the time for user input provision would be only a few minutes. Figure 6 supports road centerlines can be successfully extracted within a reasonable processing time.

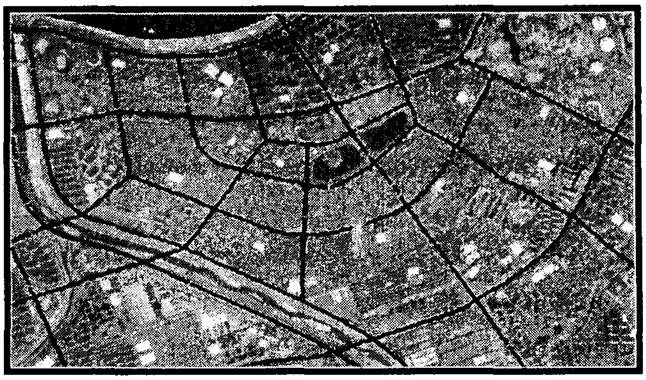

Figure 6. The result of road extraction on the dense metropolitan city of Seoul.

It is notable that there are still many roads in the figure that are not extracted by our algorithm. These are mostly small roads without centerlines and parking lots within apartment or industrial complexes. We can assume that such roads are of less importance. If we need to extract such a road, this can be done by simple manual measurements of start and end points of the road and by connecting the two points with a straight line.

5. Conclusions

In this paper, we proposed a new semi-automatic road extraction algorithm. From the high-resolution satellite image, road centerlines are extracted by using template matching based on least squares correlation matching. The experiments with an IKONOS image showed that the most of road centerlines of highways and major streets were extracted effectively given a few user input points.

One of the major contributions of this paper is that it proved that template matching could offer wider applicability of curvilinear feature extraction than energy minimization approaches.

6. References

1. Fischler, M. A., J.M., Tenenbaum, H.C. Wolf, 1981, Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique, Computer Graphics and Image

Processing, 15(1981): 201-223
2. Zlotnick, A. and P.D. Carnine, Jr., 1993, Finding road seeds in aerial images, CVGIP:Image Understanding, 57(2): 243-260
3. Gruen, A. and H. Li, 1995, Road extraction from aerial and satellite images by dynamic programming, ISPRS Journal of Photogrammetry and Remote Sensing, 50(4): 11-20
4. Geman, D. and B. Jedynak, 1996, An active testing model for tracking roads in satellite images, IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(1):1-14
5. Trinder, J. and Y. Wang, 1998, Automatic road extraction from aerial images, Digital Signal Processing, 8(1998): 215-224
6. Mayer, H. and C. Steger, 1998, Scale-space events and their link to abstraction for road extraction, ISPRS Journal of Photogrammetry and Remote Sensing, 53(1998): 62-75
7. Katartzis, A., H. Sahli, V. Pizurica and J. Cornelis, 2001, A model-based approach to the automatic extraction of linear features from airborne images, IEEE trans. on Geoscience and Remote Sensing, 39(9): 2073-2079
8. Doucette, P., P. Agouris, A. Stefanidis, and M. Musavi, 2001, Self-organised clustering for road extraction in classified imagery, ISPRS Journal of Photogrammetry and Remote Sensing, 55(2001): 347-358
9. Vosselman, G. and J. Knecht, 1995, Road tracing by profile matching and Kalman filtering, Automatic Extraction of Man-made Objects from Aerial and Space Images (A. Gruen, O. Kuebler and P. Agouris, editors), Birkhäuser, Basel, pp. 265-274
10. Hu, X., Z. Zhang, J. Zhang, 2000, An approach of semiautomated road extraction from aerial image based on template matcAhing and neural network, International Archives of Photogrammetry and Remote Sensing, XXXIII (B3/2): 994-999
11. Burns, J. B., A.R. Hanson and E.M. Riseman, 1986. Extracting straight lines, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 8, pp.425-445.