

# SEMI-AUTOMATED MAP OBJECT EXTRACTION FROM 1M RESOLUTION SPACE IMAGES

Taejung Kim\*, Young-Jae Lim<sup>a</sup>, Soo Jeong<sup>a</sup>, Kyung-Ok Kim<sup>a</sup>

Department of Geoinformatic Engineering, Inha University, 253 Yonghyunn-Dong, Nam-Gu, Incheon, ROK, [tejid@inha.ac.kr](mailto:tejid@inha.ac.kr)

\* Telematics Research Division, Electronics and Telecommunications Research Institute, 161 Gajeong-Dong, Yuseong-Gu, Daejeon, ROK, (yjl, soo, kokim)@etri.re.kr

**KEY WORDS :** Building Detection, Road Extraction, Least Squares Matching, IKONOS

## ABSTRACT:

Extraction of map objects such as roads, railroads, rivers and building boundaries from 1m resolution space images is one of the important research issue. Automation of this task is crucial for the success of the application but full and reliable automation is yet to be achieved. This paper describes the development of algorithms to extract two major map objects, roads and buildings. We adopt the “semi-automatic” approach for reliability and efficiency. For road extraction, we designed a new least squares template matching algorithm. For buildings, we combined line analysis and template matching for semi-automatic extraction. Our algorithms were tested with IKONOS images over a very dense urban scene. The algorithms developed showed promising results. The major contribution of this paper is the development of monoscopic algorithms little human intervention that produces a fair amount of information.

## 1. INTRODUCTION

Since the first commercial earth observing satellite with better than 1m spatial resolution has been launched in 1999, a number of follow-on satellites are already in orbit or in the process of development. For example, the Korea Aerospace Research Institute is developing the second satellite of its earth observation satellite series, KOMPSAT-2, with 1m resolution imaging capability and the planned launch year of 2005. These satellites are offering spaceborne images with high quality for various applications on ground.

Among many technologies needed to handle 1m resolution images, automated extraction of map objects from images seems one of the most essential and urgent ones. This technology is required not only for mapping but also for urban planning, environmental research, logistics, and etc. Currently, the task of map object extraction is done manually through head-up digitization. This process consumes too many resources and causes the usage of spaceborne images not very cost effective in real applications.

Therefore, research should be carried out to extract map objects such as roads, rivers, building boundaries from 1m resolution spaceborne images with an automated manner. This paper will report techniques developed for extracting two map objects, roads and building boundaries. While fully automated techniques are ideal, they often have accuracy and reliability problems. Instead, we focus on semi-automated approach. The work reported here is done as a part of operational SW development work. We design the extraction process such that users will guide and interact with automated algorithms for better results.

The first part of this paper is about the extraction of roads. So far, various methods have been proposed for this theme, including perceptual grouping, (Trinder and Wang, 1998; Katartzis *et al.*, 2001), scale-space approaches (Mayer and Steger, 1998), neural network and classification (Doucette *et al.*, 2001), “snakes” or energy minimization (Gruen and Li, 1997), and template matching (Vosselman and Knecht, 1995; Gruen *et al.*, 1995; Hu *et al.*, 2000). Although many authors have focused on the development of fully automated algorithms, it

seems that semi-automated algorithms such as “snakes” and template matching seemed to gain acknowledgement.

In this paper we introduce a new semi-automatic road extraction algorithm based on template matching. Our work was motivated by the previous work of Gruen *et al.* (1995) but our algorithm differs from the previous work in the following ways: we focused on tracking road centerlines from high resolution images while the previous work focused on tracking roads from mid or low resolution images; and we eliminated the need to have additional constraints for match guidance by designing a new least squares correlation matching scheme. The next section will describe this scheme step-by-step.

The second part of this paper is about the extraction of building boundaries. Previous approaches for this task includes line analysis and perceptual grouping (Shufelt and McKeown, 1993; Kim and Muller, 1999), the use of shadow information and perspective geometry (Huertas and Nevatia, 1988). To improve the quality of building extraction, several approaches used 3D information. Shufelt and McKeown (1993) and Kim and Muller (1998) combined stereo matching and line analysis for building extraction. Cochran and Medioni (1992) and Kim and Muller (1996) tried to use building detection results to improve stereo matching process. Some authors assumed 3D information on buildings was available and used this information for building extraction (Baltasavias *et al.*, 1996).

In this paper, we will focus on extraction of relatively large and rectangular shaped buildings such as apartment or industrial buildings. We will propose a monoscopic algorithm, which extracts buildings from a single image without any additional information. For better results, we may assume other data sources such as 3D building heights obtained from a LIDAR sensor or building footprints from digital maps. However, we intent to maintain with this “image only” approach to find out the maximum amount of information we can retrieve from one single image only. We, however, decided to include manual interaction for the process. We realize the building extraction by line analysis and least squares template matching with a manually-given input point. Section 3 will describe this process step-by-step.

---

\* Corresponding Author

## 2. LINEAR FEATURE EXTRACTION

We designed a new template matching algorithm based on the following considerations. In images at 1m resolution, road centerlines are appeared as curvilinear features and have distinctive brightness patterns compared to their surroundings. And hence, we could apply template matching along the road centerlines. Also we assumed that although there are geometric distortions we could model the geometric transformation between one point and the other on a road centerline as similarity transformation. Hence, 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 1)

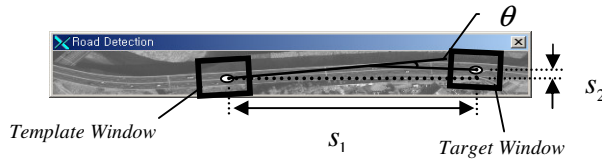


Figure 1. The template and target window for road detection

We can express the relationship between a template and a target window along a road centerline as below

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

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

where  $\theta_{template}$  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 in x and y direction, respectively, between the template and initial guess. They remain constant through iteration. Least squares correlation matching assuming this similarity transformation can be derived (Kim et al., 2004)

Our road centerline tracking algorithm works as follows. First, a user provides an input point on a road centerline. 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. Next, we estimate the orientation of the road. 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 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.

Next, 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.

Next, least squares correlation matching is 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 (see figure 2). 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.

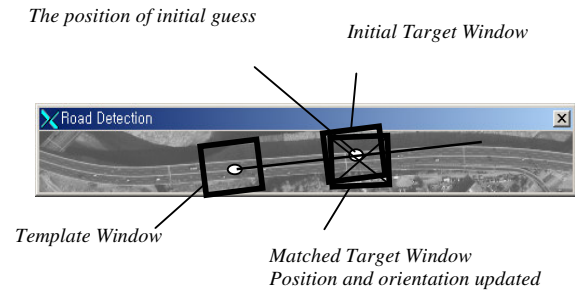


Figure 2. The definition of initial target window and matched target window

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 (see figure 10).

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.

## 3. BUILDING EXTRACTION

We designed a new building extraction algorithm based on line analysis and template matching. We assume that a user will provide a starting position of our algorithm. A user must click one point which lies inside of a building object to be extracted. This is not a very sophisticated requirement. But this is indeed very useful for eliminating many ambiguities and false alarms associated with fully automated approaches.

Our algorithm works in three parts. The first part is the extraction of lines from the subimage centred on the user-given initial position (as explained in section 3.1). The second part is the estimation of the position and orientation of building boundaries from lines and the generation of building rectangles (see section 3.2). The third part is the template matching (section 3.3). The previously extracted building boundaries are defined as a template and a least squares matching is applied to extract buildings with similar shapes.

### 3.1. Line Extraction

To extract building boundaries, we first extract lines from the image. A subimage of 100 x 100 pixels is defined centred on

the user input position. For line extraction, we used the algorithm proposed by Burns et. al(1985).

The following images are an example of the subimage centred on a building roof and lines extracted.

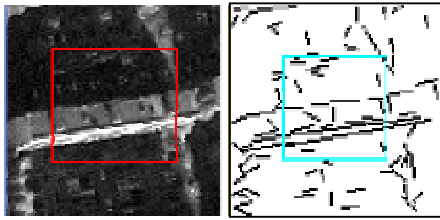


Figure 3. A subimage on a building and line image

### 3.2 Estimation of Building Position and Orientation

The orientation and position of the building within the subimage are estimated by the lines extracted. Building orientation is estimated first by voting the orientation of individual line segments. The most popular angle is assumed as the initial value of building orientation.

The estimation of building position is also done by voting. This time, we first slice the subimage into small linear sections along the direction of building orientation (see figure 4) and vote the number of line elements within each image slice. The slice with the maximum vote is assumed as the initial position where one long side of a rectangular-shaped building is located.

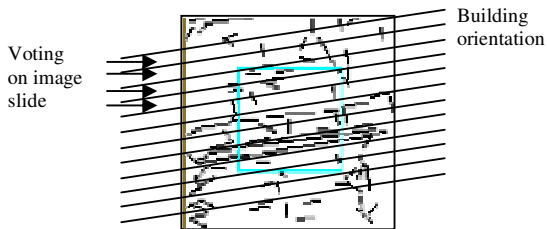


Figure 4. Estimation of the position of a building

Once we determine the initial position of building position and orientation, we refine these values by least squares template matching as the one describe in section 2. We first define a line template whose orientation and position are set as the initial values (see figure 5). Again, we assume the relationship between the line template and the true line element from a building as similarity transformation. The equations 1 and 2 also hold in this case. Based on these transformations we can design a least squares template matching. Through matching, we refine the position and orientation of the line element and hence those of a building to be extracted.

After the matching, we then have identified the position and orientation of one long side of building boundaries. We name this as the first long side. We can then estimate the position and orientation of the other long side (second long side) of building by slicing the line image with the new orientation and finding the slice of the largest vote located opposite to the first long side with respect to the subimage centre (see figure 6).

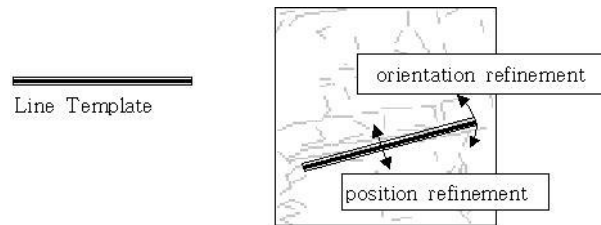


Figure 5. The line template and template matching

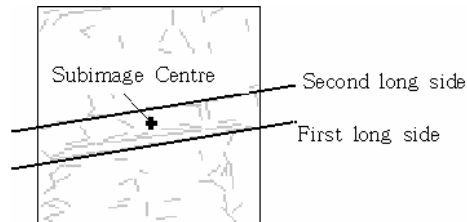


Figure 6. Estimation of second long side

Next, we need to find the other (short) sides of a rectangular-shaped building. For this purpose, we again slice the line image. But this time, we slice the image along the direction perpendicular to the orientation of the long side. And we slice not the whole image but only for the part between the first and second long sides. The first short side is determined as the slice with the largest vote. The second short side is determined as the slice with the largest vote among those lie opposite to the first short line (see figure 7).

In reality, however, it is not uncommon that short sides of a building produce very weak edge patterns. As a result, lines for those parts are often not detected at all. The line image we used here also has this problem. We cannot see meaningful line edges from the short sides of the building. Although we devised an algorithm to extract lines from the all four sides of a building, the location of short sides would not be very correct. Figure 7 shows this problem.

Nevertheless, we can retrieve the orientation and position of long sides quite accurately. Instead of devising a more sophisticated algorithm to correctly locate short sides (which is “virtually” impossible), we have assigned this task to human operators. With relatively simple operations such as rotation, scaling and translation, we can edit the building rectangle to correctly describe the underlying building. Figure 8 explains this process.

### 3.3 Template matching of a building rectangle

In most urban areas, we can observe a collection of buildings with a very similar shape. Typically in Korea, the apartment is the most popular and preferred type of residence. There are indeed many large apartment complex in any major cities in Korea and all apartment buildings do have a very similar shape with each other.

For such cases, building extraction may be done easily if we can use the previously extracted building rectangle. For this purpose, we use the least squares template matching as before. We define the building rectangle extracted previously as the template of the building to be extracted. Again, we can assume the relationship between the building template and the real buildings as the similarity transformation (equations 1 and 2).

The same least squares matching process can be applied. Through matching, the position and orientation of the building template is refined. Figure 9 explains this template matching process.

#### 4. RESULTS AND DISCUSSION

The road detection algorithm proposed here was tested with 1m resolution IKONOS images. Figure 10 shows one example of road centerline extraction on a typical highway. In the example, road orientation was estimated automatically by applying the line extraction algorithm. Once a user point was given, a series of matching was applied rightwards and leftwards. It shows that the least squares template matching we designed works.

Figure 11 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. By the series of operator's input points, the centrelines from all major roads are extracted. There are still many roads in the figure that are not extracted. These are mostly small roads without centerlines. Such roads can be extracted by measuring start and end points of the road and by connecting the two points with a straight line.

Figure 12 shows the results of building extraction from an IKONOS image by the proposed method. The left image is the results of the estimation of building rectangles by orientation and position voting process. By clicking one point on building roof, building rectangles are extracted automatically. No manual edition was applied. As mentioned before, short sides of buildings are not well detected. Sometimes, we observe false detections of even long sides of buildings, typically for those vertically aligned buildings in the left part of the image. This is due to the line extraction failure. Our method works only if there are valid line responses corresponding to the true building sides. For most cases our method correctly found the orientation and location of long sides of buildings. We argue that producing such results by only one input point from a user is very promising.

The right image in figure 12 is the result of template matching using previously extracted building rectangle. First we apply the automated extraction and manual editing. We then used this result as building template. By clicking one point on building roof, template matching is initiated and the results shown in the image are obtained. These results are without manual rotation and translation. Only the manual editing of scaling was performed to adjust the different building length.

We can see that the orientation of some buildings is slightly wrong. It seems the template matching we designed did not work properly as we intended for such cases. This is the current limitation of our algorithm. Nevertheless, by reusing the previously extracted building rectangles we generated those results almost automatically, which is also very promising.

There are other limitations of our building extraction algorithm. It was not designed to work on small house buildings. Figure 13 illustrates this. When we click on small buildings, sometimes a group of small buildings were extracted and sometimes arbitrary rectangles. Also, a user should click a point within building roof in order to get valid results. The right image in figure 13 says that even if a point is clicked on a road, the

building extraction algorithm is still initiated and somehow generate rectangles.

So far, we described the algorithm we developed to extract two major map objects from 1m resolution images. Due to the limitation of page length, we could only briefly mention the theory, procedures and performance of the algorithms developed. We believe, nevertheless, that we have shown a fair amount of information can be retrieved from only one single image with very little manual intervention with carefully devised line analysis and template matching.

#### REFERENCES

- Baltsavias, E., S. Mason, and D. Stallmann, 1996 "Use of DTMs/DSMs and Orthoimages to Support Building Extraction", *Automated Extraction of Man-made Objects from Aerial and Space Images*, (edited by A. Gruen et al.), pp.199-210, **Birkhauser**
- Burns, J.B., A.R. Hanson and E.M. Riseman, 1986, "Extracting Straight Lines", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8(4):425-445**
- Cochran, S.D. and G. Medioni, 1992, "3-D Surface Description from Binocular Stereo", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **14(10):981-994**
- 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**
- Gruen, A. and H. Li, 1997, Semi-automatic linear feature extraction by dynamic programming and LSB-snakes, *Photogrammetric Engineering and Remote Sensing*, **63(8):985-995**
- Gruen, A., P. Agouris and H. Li, 1995, Linear feature extraction with dynamic programming and globally enforced least squares matching, *Automatic Extraction of Man-made Objects from Aerial and Space Images* (A. Gruen, O. Kuebler and P. Agouris, editors), **Birkhauser, Basel**, pp. 83-94
- Hu, X., Z. Zhang, J. Zhang, 2000, An approach of semiautomated road extraction from aerial image based on template matching and neural network, *International Archives of Photogrammetry and Remote Sensing*, **XXXIII(B3/2):994-999**
- Huertas, A. and R. Nevatia, 1988, "Detecting Buildings in Aerial Images", *Computer Vision, Graphics, and Image Processing*, **41:131-152**
- 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**
- Kim, T. and J-P Muller, 1999, "Development of a Graph-based Approach for Building Detection", *Image and Vision Computing*, **17(1):3-14**
- Kim, T. and J-P Muller, 1998, "A Technique for 3D Building Extraction", *Taejung Kim and Jan-Peter Muller, Photogrammetric Engineering and Remote Sensing*, **64(9):923-930**
- Kim, T. and J-P Muller, 1996, "Automated Urban Area Building Extraction from High Resolution Stereo Imagery", *Taejung Kim and Jan-Peter Muller, Image and Vision Computing*, **14(2):115-130**
- Kim, T., S-R Park, M-G. Kim, S. Jung, K-O Kim, Tracking road centerlines from high resolution remote sensing images by least squares correlation matching, *Photogrammetric Engineering and Remote Sensing*, (in press), 2004

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  
 Shufelt, J. and D.M. McKeown, 1993, Fusion of Monocular Cues to detect Man-made Structure in Aerial Imagery, *Computer Vision, Graphics, and Image Processing: Image Understanding*, 57(3):307-330

Trinder, J. and Y. Wang, 1998, Automatic road extraction from aerial images, *Digital Signal Processing*, 8(1998):215-224  
 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), Birkhauser, Basel, pp. 265-274

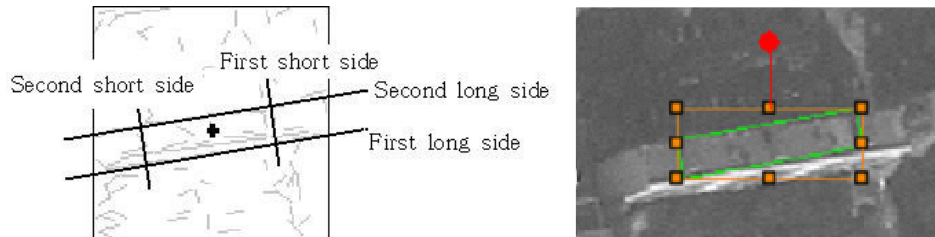


Figure 7. Four sides extracted in the line image (left) and the extracted building boundaries in the subimage (right)

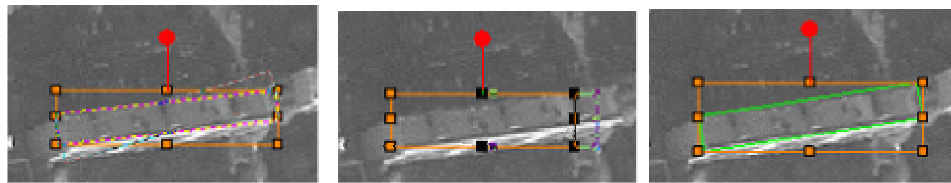


Figure 8. Manual editing of the extracted rectangle by rotation(left), scaling (middle) and translation (right)

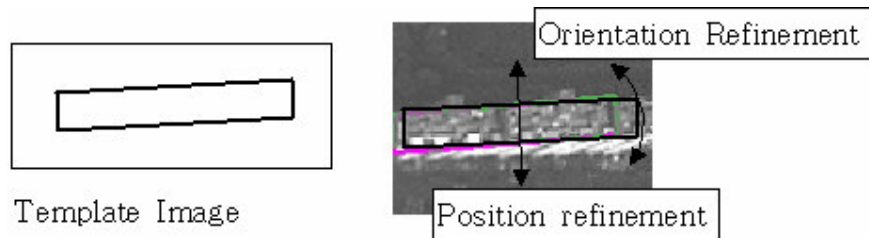


Figure 9. Building Template Matching Process

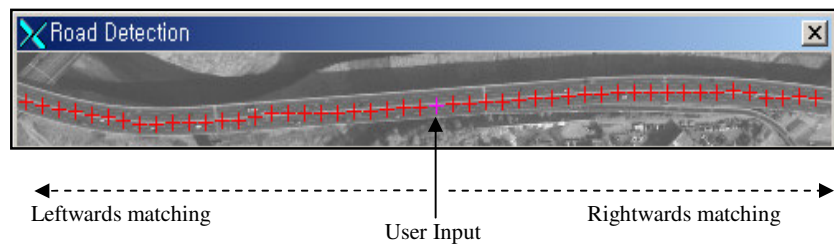


Figure 10. The results of road centreline tracking

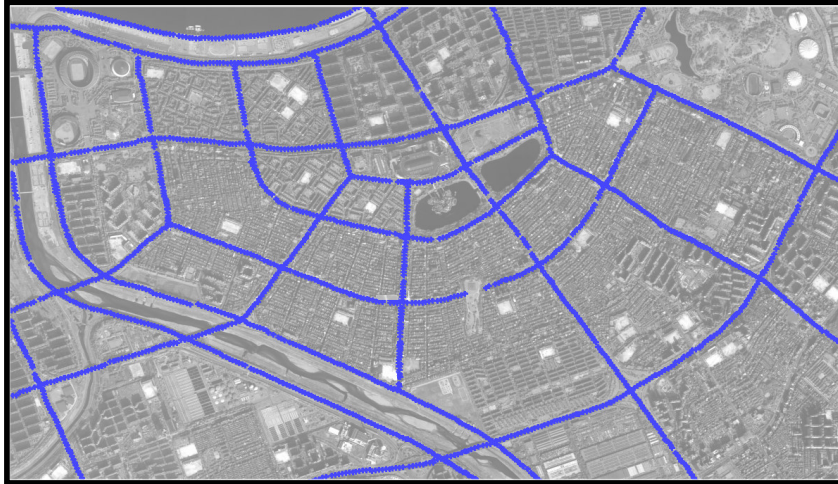


Figure 11. The result of road extraction on the dense metropolitan city of Seoul.



Figure 12. Results of building extraction. The left image is the results of building rectangle estimation line analysis and template matching only (without manual editing). The right image is the results of building template matching.

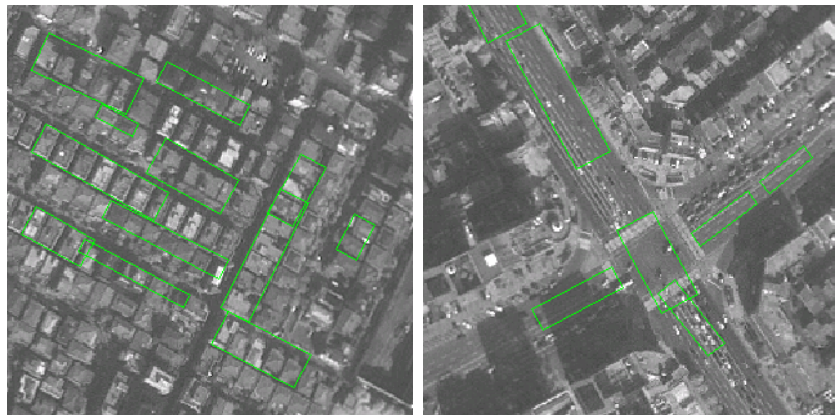


Figure 13. The limitation of the building extraction method