# AUTOMATIC URBAN ROAD EXTRACTION FROM DIGITAL SURFACE MODEL AND AERIAL IMAGERY

**Wonjo Jung**, Ph.D. Student
**Wonkook Kim**, Ph.D. Student
**Junhee Youn**, Ph.D. Student
**James Bethel**, Associate Professor
Geomatics Area, School of Civil Engineering, Purdue University, West Lafayette, IN 47907-2051, USA
wjung@purdue.edu
kim327@purdue.edu
younj@ecn.purdue.edu
bethel@ecn.purdue.edu

## ABSTRACT

Automatic linear feature extraction has a long history but still it is one of the challenging topics in photogrammetry and computer vision. In this paper, we describe an automatic road extraction process suitable for urban areas with high buildings. The proposed road extraction process has roughly three steps. The first step is generating an obstacle map to get road primitives from the digital surface model (DSM). In this step, it is assumed that roads are on the ground surface. The second step is road centerline extraction from the road primitives. Based on the assumption that roads are straight in dense urban areas, the Hough transformation is used to detect road segments and the line parameters are adjusted by the least squares method. In the third step, based on the extracted road centerlines, road edges are extracted by an adaptive snake algorithm from a true orthophoto. In this procedure, seed points for the adaptive snake algorithm are automatically generated by using the result of the previous step. The proposed road extraction process is tested on the downtown area of San Francisco, California.

## INTRODUCTION

Automatic road extraction is one of the challenging topics in photogrammetry and computer vision. The advent of GIS (Geographic Information System) forces a more frequent update of spatial feature information, and the relatively easy access to remotely sensed data makes an automatic road extraction algorithm play a more important role than before. From prior work, road extraction algorithms could be categorized into semi-automatic and automatic approaches. Within recent decades, much research effort has been expended to automate existing manual road extraction processes which were very time consuming. In semi-automatic approaches, several interactions of a human operator are allowed to ensure the quality of the final product. The profile matching method is one of the most frequently used algorithms in semi-automatic road extraction (Vosselman and Knecht, 1995; Gruen *et al*., 1995; Wang and Zhang, 2003; Eker and Seker, 2004; Hu *et al*., 2004; Kim *et. al*., 2004; Zhou *et. al*., 2005). In these algorithms, geometric constraints or a model of the road shape are implemented to guide a target window. An active contour model, known as a Snake algorithm, is also applied to semi-automatic road extraction algorithms based on manually given seed points (Gruen and Li, 1997; Youn and Bethel, 2004). Using a the classified image which requires a manual process, automatic road network verification and linking approaches have been studied (Cristianini and Shawe-Taylor, 2000; Doucette *et al.*, 2001; Song and Civco, 2004; Udomhunsakul, 2004;Valadan and Mokhtarzade, 2004). Also, moving beyond images taken by traditional optical sensors, three dimensional LIDAR data or SAR(Synthetic Aperture Radar) data are tested in several recent projects (Wessel *et al*., 2003; Hu *et al.*, 2004). Based on those semi-automatic algorithms, fully automated road extraction algorithms have been developed (Agouris *et al.,* 2001; Doucette *et al.*, 2001; Hinz and Baumgartner, 2004). In these algorithms, local information like intensity profiles and pixel gradients are mainly used to extract road segments. Also road extraction algorithms can be classified by the area of interest (specifically rural and urban areas). Existing semi-automatic algorithms can be powerfully performed on rural areas and even automatic algorithms have shown good results. However, dense urban areas are still challenging places to adapt an existing road extraction algorithm without modifications.

The main object of this study is to develop an automated road extraction process suitable for dense urban areas with high buildings. The proposed road extraction process has roughly three steps. The first step is generating a road primitive map from the digital surface model (DSM). The second step is the centerline extraction from the road primitives. In the last step, based on the extracted road centerlines, road edges are extracted by an adaptive snake algorithm from a ture orthophoto. In this procedure, seed points for the adaptive snake algorithm are automatically generated from the result of the previous step. In the next part, detailed procedures and experimental results will be described and discussed.

## DETECTION OF ROAD CENTERLINES FROM DSM

In this paper, two kinds of data sources, DSM and aerial photographs, are used. Aerial photographs contain very detailed radiometric and spatial information about features. For this reason, aerial photographs have been widely used for road feature extraction. However, due to the central projection geometry, relief tilt displacements are inevitable. In rural areas, the absence of occluding features causes road features to be usually clearly seen. In dense urban areas, large portions of the road network are occluded by buildings. To recover areas hidden by buildings, we need to produce a true orthophoto. Such an orthophoto is a good data source for road extraction because it contains both radiometric and spatially registered information. However, the presence of extraneous features also makes a challenge for automatic algorithms to detect roads. For example, cars running on the road or shadowed areas of the road act like noise. Before using the detailed radiometric information in the true orthophoto, we employed DSM containing only spatial information to reduce the effect of the noise. The test area (3072m ×3072 m) is the downtown section of San Francisco, having several skyscrapers and lots of high buildings. The grid spacing of the DSM is 1 meter and its coordinate system is UTM (Zone 10). The GSD (Ground Sample Distance) of the true Orthophoto is 40 cm with the same coordinate system of the DSM.

### Obstacle Area Map Generation

In this paper, two characteristics of roads are assumed in the detection of the road network from DSM. The first one is that roads are exactly on the ground surface, and the second one is that roads are straight in dense urban areas. From the first assumption, road primitives and candidates can be obtained by generating an obstacle area map. To explain the term, “obstacle”, first the occlusion area in an aerial photograph should be explained. The main idea of the obstacle originates from occlusion area designation for aerial photographs. Every aerial photograph has its exposure station. If we have both DSM and an aerial photograph of the same area, based on the assumption that rays are aligned straight from a ground feature to the exposure station, we can determine whether a specific location in the scene can be seen or not from this exposure station. An occlusion area in an aerial photograph is invisible from its exposure station and can be detected when a certain feature obstructs the ray from the point in question. Therefore, the term “obstacle” can be defined as the feature obstructing rays from other points to the exposure station. Figure 1(a) shows that the basic relation between occlusion areas and obstacle areas.

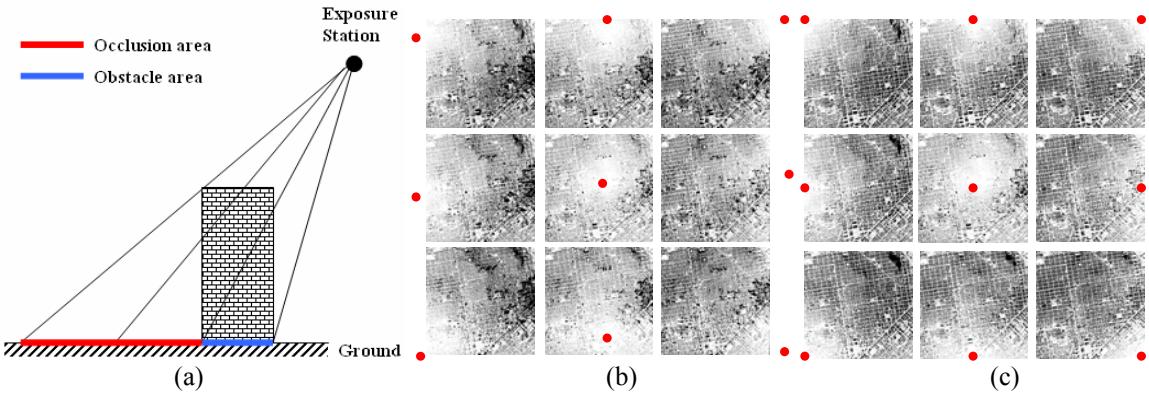



**Figure 1.** Obstacle area map (a) concept of the occlusion and obstacle area, (b) actual image exposure stations (red dots) and detected obstacle areas (black pixels) over the area of interest, (c) fictitious image exposure stations and detected obstacle areas (black pixels) over the area of interest.

To implement this obstacle detection algorithm using the DSM, nine exposure stations (red dots in Figure 1(b)) of the aerial photographs are used. As seen in Figure 1(b), near the nadir of each exposure station fewer obstacle areas are present compared to radially distant areas. The main purpose of detecting obstacles is to get road primitives from the DSM. Since we assumed that roads are right on the ground surface, obstacles are definitely not roads. Since obstacles are marked as black in this figure, white pixels in the obstacle area map are road primitives. In case of using different images like a high resolution satellite image, the narrow field of view causes a different occlusion pattern and we would possibly have to revise the algorithm. In Figure 1(c), nine fictitious exposure stations are generated based on the knowledge of a typical flight plan for aerial photography. Figure 1(c) shows the selected nine fictitious exposure stations and detected obstacle areas. Height of the exposure station is chosen as 1500m above mean sea level. It is clearly seen that occlusion areas shown in figure 2(b) are generated by obstacles in figure 2(a); the higher obstacles, the longer occlusion areas. For the later steps, we used the obstacle map generated by fictitious exposure stations.

## Reducing Noise Using Morphological Operations

Once obstacle areas are detected, non-obstacle areas which are designated by white pixels in figure 2(a) will be used as road primitives, based on the assumption that roads are on the ground surface. Note that we are currently unable to detect overpass roads under this assumption. As presented in the enlarged view of figure 3(b), severe noise is found on the road and in open areas like parking lots. This is mainly due to small objects like cars, trees and small height variations of DSM itself. Before detecting road elements, noise is reduced by applying dilation and erosion methods to the obstacle map consecutively. As you see in Figure 3, building edges are well preserved even after the noise reduction process. When reducing noise of the obstacle map, we need to be careful not to lose boundary information of the roads. In this experiment, a 3 by 3 structuring element is used (see Figure 3(a).)

(a) obstacle areas (black) (b) detected occlusion areas (black)

**Figure 2.** Obstacle areas versus occlusion areas; fictitious exposure station lies on the left side of images.

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

(a) structuring element (b) before noise reduction (c) after noise reduction

**Figure 3.** Noise reduction using dilation and erosion method.

## Estimating Dominant Directions of Road Networks

As previously mentioned, roads are assumed to be straight objects in dense urban areas. The main idea of this section is extracting road centerlines using the Hough transformation from the obstacle area map. When the Hough

transformation is directly applied to the obstacle area map, roads are poorly extracted. This is mainly due to the proportion of white pixels. Since the proportion of white pixels is almost half (49%), the Hough transform gave us a lot of non-road lines which pass through obstacles but contain a large proportion of white pixels. For example, the line having the maximum value in parameter space was the diagonal line of the image; longer lines contain more white pixels than true road lines. This problem might be solved by modifying the Hough transform so that it detects only continuous white pixels. However, not only non-road lines will be detected but also road lines with noise will not be detected if we use this approach. To overcome this problem, the concept of dominant road directions is employed in our process. Since road networks often have grid form in dense urban areas, the number of directions of roads is usually small with many roads mutually parallel. To estimate dominant road directions, main roads are detected from the 16 times zoomed (downsampled) obstacle map (192 ×192 pixels, GSD is 16 meters.) In this experiment, a road longer than one kilometer is considered as a main road and detected by using a rotating bar template with width of 1 pixel and length of 64 pixels. The concept of the rotating bar is shown in Figure 4(a). When the rotating bar finds a line consisting of more than 90 percent white pixels, that line becomes a main road (Figure 4(b).). Once main roads are detected, the angle of each main road is measured and grouped. Grouping of angles is done by the following rules. When grouping angles, clusters are grouped into one angle and the representative value of a group equals its mean. In this experiment, four groups are detected and their values are 41°, 80°, 135°, and 171° with respect to column direction of the image (clockwise). Once grouping is done, road primitives are divided into four images (see Figure 4 (c)-(f)) based on the grouped angles. Based on these four road primitives, the Hough transformation is applied to detect road segments.

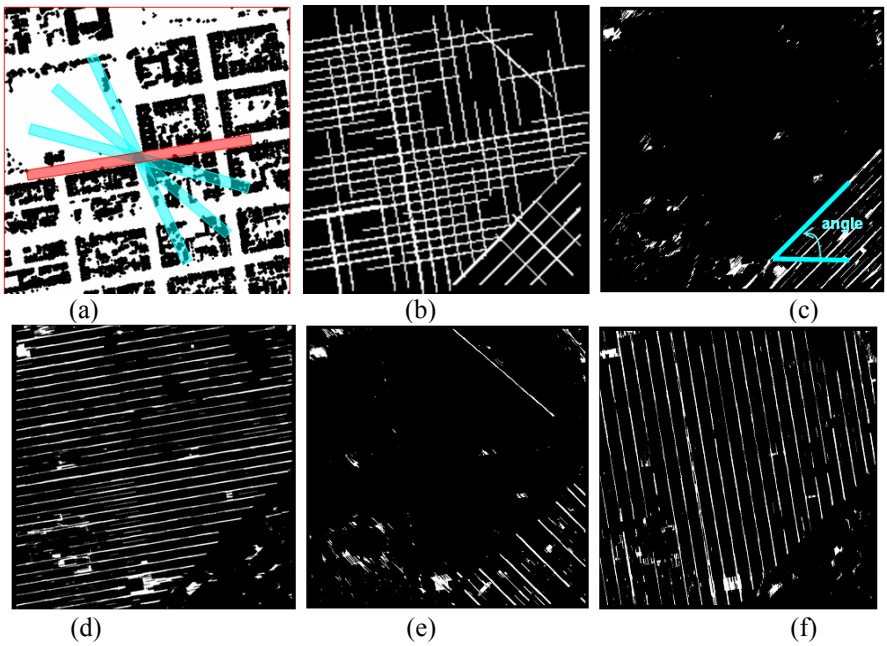



**Figure 4.** Rotating bar results and detected dominant roads ; (a) the concept of the rotating bar; the red bar consists of more than 90 percent white pixels but not other bars, (b) dominant roads detected by rotating bar, (c)-(f) road primitives generated by dominant angles

## Extraction of Road Centerlines Using the Hough Transformation

In our process, road centerlines are extracted by the Hough transformation algorithm. The basic idea of the Hough transformation is that a line corresponds to a certain point in a parameter space. Usually, the equation of the line is represented in two ways for the Hough transformation. Figure 5(b) describes the parameter space when a line is represented with Equation 1; $a$ is the slope of the line and $b$ is the y-intercept of the line. The problem with Equation 1 is that both the slope and intercept approach infinity as the line approaches the vertical (Gonzalez, 1992). Figure 5(c) describes the parameter space when a line is represented with Equation 2; $\theta$ ($-90° < \theta \leq 90°$) is an angle from the $x$ axis and $\rho$ is the perpendicular distance from the origin to the line. If we represent lines using Equation 2, we could expect a finite parameter space for a finite input.

$$y = ax + b \quad \text{(Equation 1)}$$

$$x\cos\theta + y\sin\theta = \rho \quad \text{(Equation 2)}$$

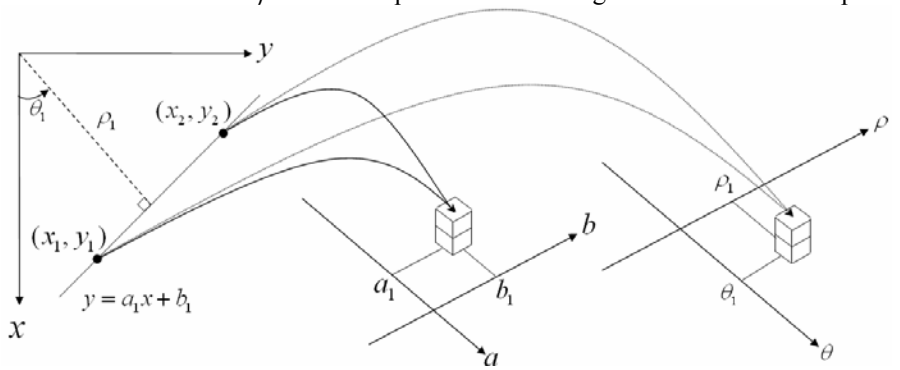
Once the road primitives are transformed into the parameter space, we need to decide how many roads we will detect for each road primitive image shown in Figures 4(c)-(f). To decide the number of roads, the distances between parallel lines are detected. For each road primitive image, five peak points were initially detected in its parameter space. Since each road primitive image contains lines with the same direction, five peak points have almost the same $\theta$ values but different $\rho$ values. Then a minimum distance, $d_{min}$, is calculated using Equation 3. We select the minimum distance between all pairs of the five lines. Assuming that the distance between two adjacent roads is equal to or greater than $d_{min}$, the maximum possible number of roads , $N_{max}$, for a road primitive image is calculated using Equation 4.

$$d_{\min} = \min\{|\rho_i - \rho_j| \in Real : i \neq j,\ 1 \le i, j \le 5,\ i, j \in Integer\} \quad \text{(Equation 3)}$$

where $\rho_i$ is the element of $\rho$ values of the five peak points in parameter space.

$$N_{\max} = \frac{\rho_{\max}}{d_{\min}} \quad \text{(Equation 4)}$$

where $\rho_{max}$ is the maximum value of the $\rho$ which is equivalent to the diagonal distance of the input image.




(a) a line of equation (b) parameter space I (c) parameter space II

**Figure 5.** The Hough transformation and its graphical representation

Based on $N_{max}$, the road segments are extracted from the parameter space and short lines are removed. For this data set, the number of extracted roads is 62 and is depicted in Figure 6 (a).

It is worth mentioning that the Hough transformation works well on salient and thin edges. In this experiment, road primitives are frequently discontinuous and thick (see Figure 4). The problem is that thick line segments produce multiple high responses around the highest peak. There are several approaches to solve this problem such as a multi-resolution strategy which can make roads become thin lines, or a proximity filter which will allow only one point within the cluster areas in parameter space. However, widths of roads are variable and it is hard to find a proper scale for all road widths even within this single test area. So, the multi-resolution approach is not applied to this problem. An estimation approach can give us a good approximation of the center line of each road segment. However, we are not sure whether the estimated peak point represents the centerline of the road segment or not if we choose the peak point using an interpolation method. In this paper, the least squares adjustment method is implemented to solve this problem.

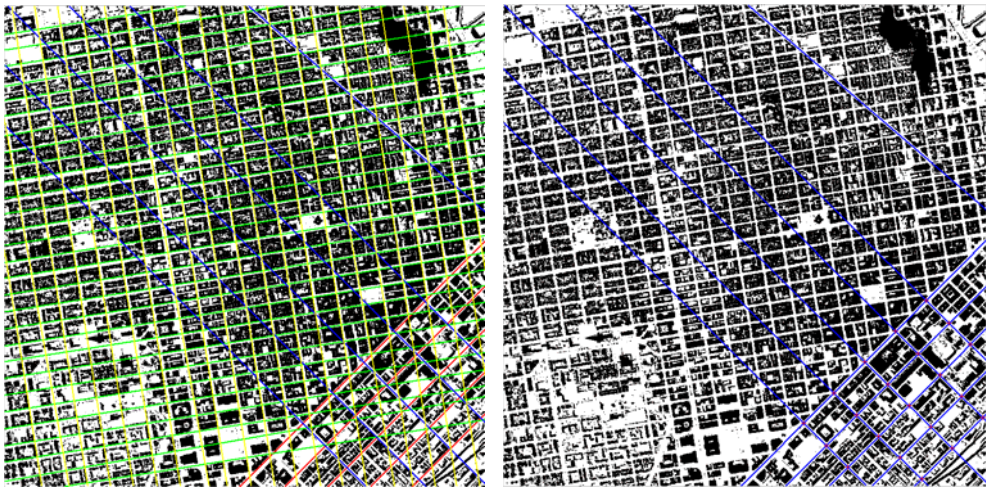
## Refining Road Centerlines Using The Least Squares Method

This problem can be considered as a line regression problem. Once a road is extracted as a single line from the previous step, usually the line is not at the center of the road primitives as previously explained. To refine an extracted road line, the coordinates of the road primitive pixels within 10 meters from the line are observed. Then

based on the observed coordinates of pixels, condition equations are constructed. The condition equations are equations 5 and 6 for the line models of equations 1 and 2 respectively. In equations 5 and 6, the observations are ($x$, $y$) coordinates and the line parameters are ($a$, $b$) and ($\rho$, $\theta$) respectively. Initial values of the parameters for the nonlinear model can be obtained from parameter space in the Hough transformation. Through this refinement process, we can estimate road centerlines. If erroneous road primitives act as outliers in the adjustment process, we could use the L1 norm instead of the L2 norm as the objective function; the L2 norm is used in this experiment.

$$F = y - ax - b = 0 \qquad \text{(Equation 5)}$$

$$F = \rho - x\cos\theta + y\sin\theta = 0 \qquad \text{(Equation 6)}$$

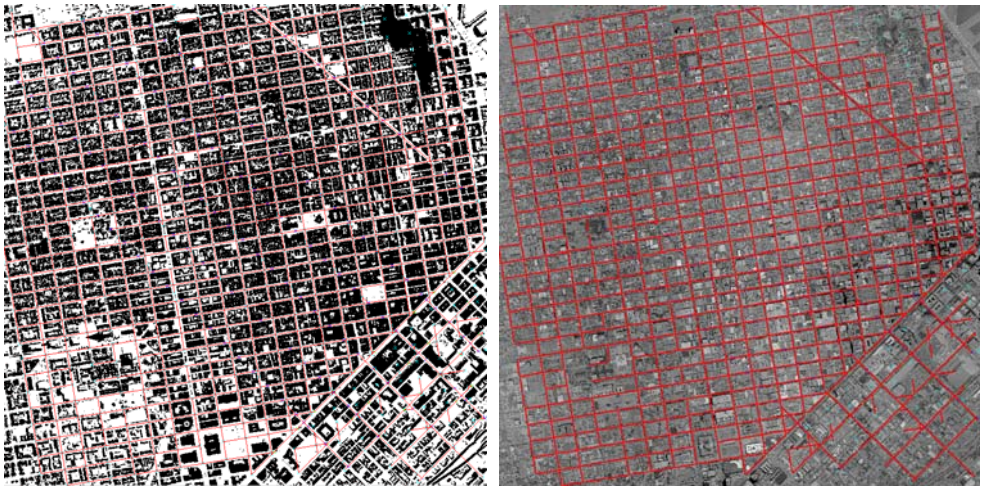


(a) (b)

**Figure 6.** Extracted road centerlines and road intersections; (a) extracted roads using the Hough transformation, (b) refined road lines using least squares adjustment and intersections

## Road Centerline Verification and Evaluation

In the previous step, a total of sixty two lines are extracted and refined. Using these refined line parameters, we can find possible intersections within the image. The number of possible combinations of two among sixty two lines is 1891 ($C_2^{62} = (62\times61)/(2\times1)$) and, at most, 1891 intersections are determined. Once intersections are determined, a road segment which is defined by two adjacent intersections is verified by simple algorithms. If more than eighty percent of the road segment consists of white pixels, the road segments is verified; the threshold value, eighty percent, is acquired by experience. The result is shown in Figure 7. To evaluate our process, reference road regions are manually extracted by using the ArcMAP module in ESRI ArcGIS software. For counting purposes, extracted centerlines and reference road regions are converted into a binary raster format. The counting rule is that a correctly extracted road centerline pixel is a pixel within the reference road regions. As a result, the suggested automatic road extraction process showed 84% “completeness” and 93% “correctness”. The completeness is the percentage of the reference data covered by extracted data. And the correctness represents the percentage of the extracted data covered by reference data.



(a) (b)

**Figure 7**. Extracted road segments from DSM; (a) Extracted road segment overlaid on obstacle areas (b) Extracted road segments displayed on true orthophoto.

## CITY BLOCK DELINEATION WITH SNAKES

Now the problem we have is how to delineate the city blocks using the derived intersection candidates and the road centerline candidates. The intersection points are the most salient features which make it possible to characterize the city block. The initial approximation to the city block is considered as a polygon having the intersection points as its vertices and the connected lines between those points. The initial approximation is used as an initial contour for the Snakes. Since there tends to be large change in image intensity along the perpendicular direction to the side area of city block, we try to use an intensity variation as an image force for the Snakes.

### The Solution For The Snakes

In this study, the snake algorithm is used for obtaining the shape of the city blocks. First the general concept of the snake algorithm is described and in the second part, a small alteration in the algorithm is introduced to make it better fit this specific problem. Our objective is to find the shape and location of the city blocks from the collected intersection points. Since the actual city block consists of several buildings and small roads, its boundary cannot be a straight line. It can be very irregular. For simplicity, however, we would like to obtain so refined a shape that we could think of it as a polygon. We also expect the polygon to have a smooth shape and to be located at the boundary of a city block as exactly as possible. This end can be successfully accomplished by the snake algorithm by employing the concept of energy minimization. Kass *et al*.(1988) unified the contributions from the physical shape of the contour, the external constraint on the contour and the influence of the image intensity on the contour under the measurable quantity, energy. Once all those three criteria are expressed in one equation, the problem is now to find the optimal contour that gives minimum total energy.

The usual measure of contour smoothness is called internal energy. The degree of how elastic and how stiff the contour is, is formulated by a first and second derivative of the contour vector respectively. So the internal energy can be expressed as

$$E_{\text{int}} = \int_0^1 \alpha(s)\left|v_s(s,t)\right|^2 + \beta(s)\left|v_{ss}(s,t)\right|^2 ds \qquad \text{(Equation 7)}$$

where $s$ is a space (curve) parameter, $t$ is a time (iteration) parameter, and $\alpha(s)$, $\beta(s)$ are coefficients for elasticity and stiffness. The larger the elasticity, the tighter the contour will be and the larger the stiffness is, the harder to bend the contour. We can control the behavior of the contour by adjusting these coefficients.

The measure of the influence of the image intensity is called image energy. It could be any function that is mapped from the image intensity.

$$E_{img}(v) = -\int_0^1 P(v(s,t))ds \quad \text{(Equation 8)}$$

where $P(v(s,t))$ is function value corresponding to the feature of interest. Since we focus on salient edges between a city block and a road, the corresponding function $P(v(s,t))$ can be expressed as a norm of the image gradient with respect to the image coordinates x and y.

$$P(v(s,t)) = \left|\nabla I(x,y)\right|^2 \quad \text{(Equation 9)}$$

As a result, the total energy of the snake is written as

$$E_{total} = \int_0^1 \alpha(s)\left|v_s(s,t)\right|^2 + \beta(s)\left|v_{ss}(s,t)\right|^2 - P(v(s,t))ds \quad \text{(Equation 10)}$$

There are many ways to minimize the energy of an active contour model such as dynamic programming proposed by Amini *et al*.(1988) and the conjugate gradient descent method. In this study, we use the Euler-Lagrange equation and solve it with finite difference analysis. The equation can be written as

$$\gamma\frac{\partial v(s,t)}{\partial t} + \alpha v_{ss}(s,t) + \beta v_{ssss}(s,t) = -\frac{\partial P(v(s,t))}{\partial v} \quad \text{(Equation 11)}$$

where $\gamma$ represents the viscosity of the medium. The higher viscosity makes the evolution of the curve slower. $v_{ss}$ is the second derivative with respect to s. Finite difference analysis with iterations gives the optimal position of the contour curve with this energy minimization technique.

### City Block Delineation and Local Coefficient Snakes

To delineate the urban area city blocks, Youn and Bethel (2004) used a local energy coefficient for the Snakes (local energy coefficients means that coefficients vary locally). In this paper, we apply the same concept for block delineation. In Equation 7, $\alpha$ controls the elasticity of the contour and $\beta$ controls the stiffness of the contour. With the energy minimization, large $\alpha$ makes the consecutive points closer to one another and large $\beta$ makes the shape of line, which is composed of discrete points, smooth. Considering the city block, the side area should be straight, although extraneous features can cause ambiguity. So, larger $\alpha$ and small $\beta$ for the side area compared to the corner area makes the side area of city block straight.

In the previous section, we made the intersection candidates and the road centerlines and they compose the initial approximation for the Snakes with two steps. As a first step, we group the intersections near each city block among the all intersection candidates and they are used as vertices of the block polygon. Secondly, we add the points between the grouped intersection candidates. Also, points near the intersection are considered as having corner attributes and other points are considered as having side attributes. Applying local energy coefficients to the Snake energy minimization, we extract the city blocks. The result of the extracted city blocks is presented in Figure 8. Note that these figures require rendering in color to see all significant aspects.

## RESULT AND CONCLUSION

In this paper, the automatic road extraction process is tested on the downtown area of San Francisco. The suggested automatic road extraction process showed 84% completeness and 93% correctness. Since we assumed straight roads and only surface streets, several curved roads and overpass roads are not detected. Also, several non-road segments within open areas like parking lots were incorrectly detected as roads.

After extracting road centerlines and intersections, the city blocks are delineated by using an adaptive snake algorithm. Initial seed points for the Snake are automatically generated by grouping intersection candidates together with road centerlines.

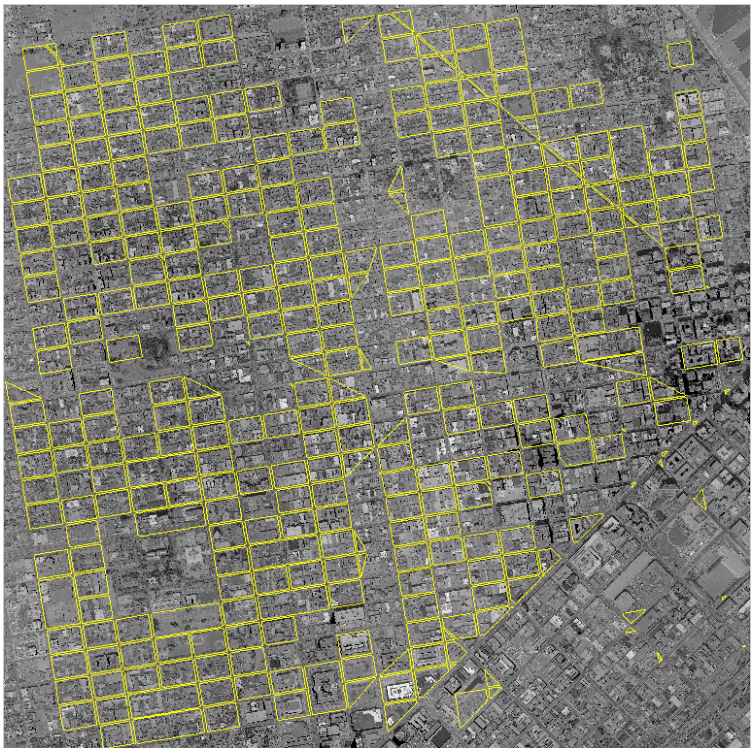


**Figure 8.** The Delineated City Blocks with the Local Coefficient Snakes

## REFERENCES


Agouris, P., A. Stefanidis, and S. Gyftakis, 2001. Differential snakes for change detection in road segments. *Photogrammerict. Engineering and Remote Sensing* 67(12):1391–1399.

Amini, A A, S. Tehrani, T.E. Weymouth, 1988. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. *Second ICCV*, pp. 95-99.

Cristianini, N. and J. Shawe-Taylor, 2000. An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, Cambridge, UK.

Davis, D.N., K. Natarajan, and E. Claridge, 1995. Multiple energy function active contours applied to CT and MRI images of the brain, *Proceedings of the 5th IEE Conference on Image Processing and its Applications*, Edinburgh, UK, pp. 114-118.

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(5-6):347–358.

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), Birkhaeuser Verlag, Berin, Germany, pp. 83-94.

Gruen, A. and H. Li, 1997. Semi-automatic linear feature extraction by dynamic programming and LSB-Snakes. *Photogrammetic Engineering and Remote Sensing* 63(8):985–995.

Gonzalez, C. R. and R. E. Woods, 1992. Digital Image Processing, Addison Wesley, pp.432-438.

Eker, O. and D. Z. Seker, 2004. Semi-automatic Road Extraction from Digital Aerial Photograph, *XXth ISPRS Congress*, Commission 3, 12-23 July 2004 Istanbul, Turkey, unpaginated CD-ROM.

Hinz, S. and A. Baumgartner, 2004. Automatic extraction of urban road networks from multi-view aerial imagery, *Photogrammetric Engineering and Remote Sensing*, 70(12):1405-1416.

Hu, X., Z. Zhang and C. V. Tao, 2004. A Robust Method for Semi-Automatic Extraction of Road Centerlines Using a Piecewise Parabolic Model and Least Square Template Matching, *Photogrammetric Engineering and Remote Sensing*, 70(12):1393-1398.

Kass, M., A. Witkin, and D. Terzopoulos, 1988. Snakes: Active contour models, *International Journal of Computer Vision*, 1(4):321-331.

Kim, T., S. Park, M. Kim, S. Jeong, and K. Kim, 2004. Tracking Road Centerlines from High Resolution Remote Sensing Images by Least Square Correlation Matching, *Photogrammetric Engineering and Remote Sensing*, 70(12):1417-1422.

Song, M. and D. Civco, 2004. Road Extraction Using SVM and Image Segmentation, *Photogrammetric Engineering and Remote Sensing*, (70)12:1365-1371.

Udomhunsakul, S., 2004. Semi-automatic Road Detection from Satellite Imagery, *Proceedings of the 2004 International Conference on Image Processing (ICIP 2004)*, Singapore, October 24-27, 2004, IEEE, pp.1723-1726.

Valadan Zoej, M.J. and M. Mokhtarzade, 2004. Road Detection from High Resolution Satellite Images Using Artificial Neural Networks, *XXth ISPRS Congress*, Commission 3, 12-23 July 2004 Istanbul, Turkey, unpaginated CD-ROM.

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), Birkhaeuser Verlag, Berin, Germany, pp. 265-274.

Wang, R. and Y. Zhang, 2003. Semi-automated Road Extraction from QuickBird Imagery, *ISPRS Joint Workshop on Spatial Temporal and Multi-Dimensional Data Modeling and Analysis*, October, 2-3, 2003, Quebec, Canada, unpaginated CD-ROM.

Wessel, B., C. Wiedemann and H. Ebner, 2003. The role of context for road extraction from SAR imagery, *Proceeding of IEEE International*, *Geoscience and Remote Sensing Symposium*, 21 – 25 July, 2003, Toulouse, France, unpaginated CD-ROM.

Youn, J., and J. S. Bethel, 2004. Adaptive Snakes for Urban Road Extraction, *XXth ISPRS Congress*, Istanbul, Turkey, July 12-23, 2004, CD-ROM

Zhou, J., W.F. Bischof, and T. Caelli, 2005. Robust and Efficient Road Tracking in Aerial Images, *Joint Workshop of ISPRS and the German Association for Pattern Recognition, Vienna, Austria, 29-30 August* 2005*, IAPRS, Vol. XXXVI, Part 3/W24*, pp.35-40.