# AUTOMATIC 3D BUILDING EXTRACTION FROM STEREO IKONOS IMAGES

**Zhen Xiong,** Post Doctoral Fellow
Department of Geodesy & Geomatics Engineering
University of New Brunswick,
15 Dineen Drive, PO Box 4400
Fredericton, NB, Canada E3B 5A3
Centre for Remote Imaging, Sensing & Processing (CRISP)
National University of Singapore
Lower Kent Ridge Road, Singapore 119260
zxiong@unb.ca

**Yun Zhang,** Associate Professor
Department of Geodesy & Geomatics Engineering
University of New Brunswick
yunzhang@unb.ca

## ABSTRACT

In 3D GIS research and development, how to effectively acquire and input a large amount of 3D buildings is always a tough challenge. To date, as the most stable and accurate method, through a photogrammetric instrument, manually measuring buildings one by one, is really a tiring and time-consuming work. Therefore, many researchers have been trying to develop semi-automatic building extraction method or even completely automatic building extraction methods. In this paper we introduce a new method of automatic 3D building extraction based on stereo Ikonos images. This method includes several steps, such as edge detection and connection, building block detection and corner detection, and building extraction and image matching. The final result is 3D coordinates of buildings. We used a scene of stereo images to test our program. Only about 15 minutes were needed to process a scene of Ikonos image (2613 by 3539 pixels). The final result is reported. Corresponding discussions are conducted and conclusions are given based on the test results and discussions.

## INTRODUCTION

Building extraction from digital images is needed for many tasks. But at the same time it is also a difficult task. Because building detection and recognition from images is such a complex problem that it can not be solved in a single step (Paparoditis, 1998).

Before the launch of Ikonos on September 24, 1999, most of the researchers use aerial photos to extract buildings (Paparoditis, 1998, Lu, 2004, Andre, 1998, Lin, 1998). Along with the development of remote sensing techniques, the spatial resolution of satellite image becomes higher and higher. Gradually more and more scientists tried to use high resolution image to extract buildings (Donna, 2004).

But no matter aerial photo or satellite image, the methods to extract buildings are not essential different. Because building is construction above ground, most of the researchers utilize this character to extract building. Basically we can summarize these methods into four categories.

(1) Use DEM and DSM to find building
    Uwe Weidner et al (1995) use high resolution digital elevation model (DEM) to extract buildings. After segmentation on this DEM, they get an outline of buildings. Based on these building outlines, through discretization noise elimination and minimum description length (MDL) -based polygon simplification, finally they can reconstruct the building.
(2) Use Laser scanning data to find building
    This method mainly uses image segment and reasoning techniques to extract building (Zhan, 2004. Michel, 2001. J. dash, 2004 ).
(3) Use image shadow to estimate building height

Another method uses the building shadow to estimate the building height (Lin, 1998). In this method, many researchers like to use building models to fit the image so as to estimate the building model's parameters.

(4) Photogrammetric method
Photogrammetric method generally uses a stereo pair to generate DSM firstly, and then use image segment technique to extract buildings (Paparoditis N., 1998 ).

Due to the complexity of the task, fully automatic building extraction is considered not yet operational. Many semi-automatic systems are currently being developed (Englert, 1998; Müller, 1998; Gruen and Wang, 1998; Veldhuis, 1998; Rottensteiner, 2000) which offer a compromise between the great demand for automation in data extraction and the fact that the problem has not yet been completely solved. In semi-automatic systems, recognition and interpretation tasks are performed by the human operator, whereas modeling and / or precise measurement is supported by automated tools.

On the other hand, some researchers have tried to develop fully automatic system for building extraction (Donna, 2004; Yi, 2002). Both of their research results are encouraging, but not perfect. Haverkamp can only extract the flat roof buildings with clear building outlines. Lu used DSM and multi-spectral classification to process the urban area, but most of the extracted houses are not complete, and some houses are lost. Our research may be the latest attempt. It is a fully automatic system and belongs to photogrammetric method. After some image preprocessing, we conduct edge detection and edge chaining, building block detection and corner detection, finally we use image matching method to look for the corresponding corner points of buildings on both images and calculate their coordinates.

## METHODOLOGY

The procedure of 3D building extraction contains several steps. From input image to output 3D building outlines, we first extract building edges from image, then conduct edge connection to make the building edges become a closed block. Later on detect the closed block and building corners, finally get coordinates of building corners based on stereo photogrammetry and image matching.

### Edge detection and connection

Basically we follow the Canny detector to extract edge. This procedure contains several steps.

(1) Image smoothing
It is necessary to reduce the affection of image noise to the edge extraction. The general method is image smoothing by Gaussian convolution (figure 1) (Dmitry, 1999).

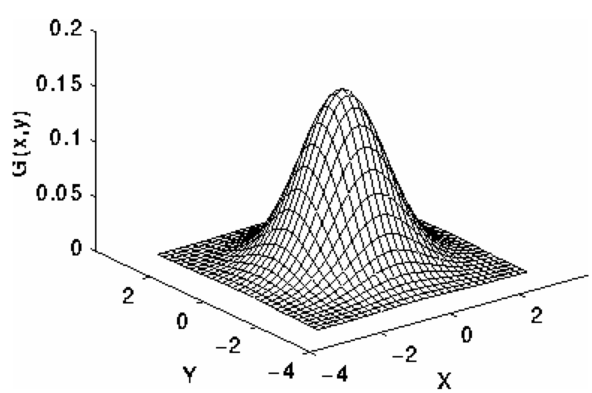



**Figure 1.** 2-D Gaussian distribution with mean (0, 0) and σ =1

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (1)$$

Where σ is the standard deviation of the distribution.

(2) Generate gradient image

Then a simple 2-D first derivative operator (somewhat like the Roberts Cross) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives (figure 2).

$$T1 = (g(i, j+1) - g(i, j) + g(i+1) - g(i+1, j))/2.0 \qquad (2)$$

$$T2 = (g(i, j) - g(i+1, j) + g(i, j+1) - g(i+1, j+1))/2.0 \qquad (3)$$

$$T3 = sqrt(T1^2 + T2^2) \qquad (4)$$

$$Angle = a\tan(T2/T1) \qquad (5)$$

Where T1 is the first derivative along x direction, T2 is the first derivative along x direction, T3 is the final gradient magnitude, and *angle* represents the gradient direction.

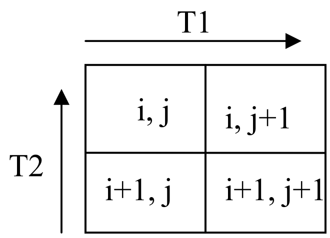



**Figure 2.** 2-D first derivative operator

(3) Non-maximal suppression

Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets zero to all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as *non-maximal suppression*. Figure 3 shows the ridge. This ridge is south north direction. For the central point, its gradient magnitude is 23. It is greater than its left pixel and right pixel. So this point is a ridge point.

| | | |
|---|---|---|
| 10 | 20 | 13 |
| 11 | 23 | 10 |
| 14 | 21 | 12 |

**Figure 3.** Non-maximal suppression

The tracking process is controlled by two thresholds: *Threshold1* and *Threshold2* with *Threshold1* > *Threshold2*. Tracking can only begin at a point on a ridge higher than *Threshold1*. Tracking then continues in both directions out from that point until the height of the ridge falls below *Threshold2*. This process helps to ensure that noisy edges are not broken up into multiple edge fragments. For example, we set threshold1 equals 22 and threshold2 equals 20, then we just can begin the tracking from 23 (figure 3), and later we can find the ridge points 20 and 21.

(4) Edge connection

After edge detection, we can not conduct building extraction immediately, because for some buildings their edges are not closed. If we find an edge point is connected with only one edge point, we called it "open" point.

The edge connected with “open” point is “open” edge. We should first connect the “open” edges together so that every building is a closed block.

Basically we conduct edge connection only if the edges meet the following conditions.

At least two edges have been detected

Every edge must be longer than a threshold

The intersection angle of this edges must be close to 90 degree

Once the detected edges meet our conditions above, we can chain the edges to a closed block according to following methods.

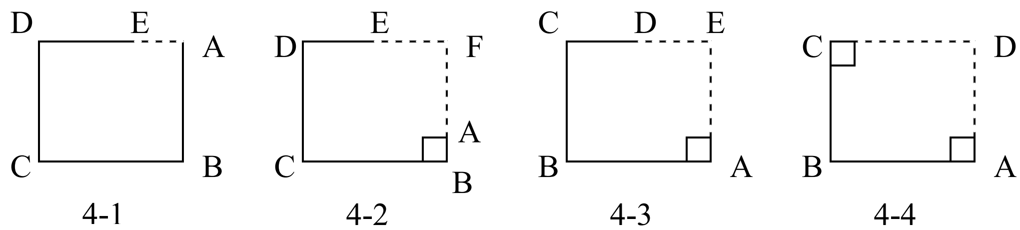



**Figure 4.** Edge connection

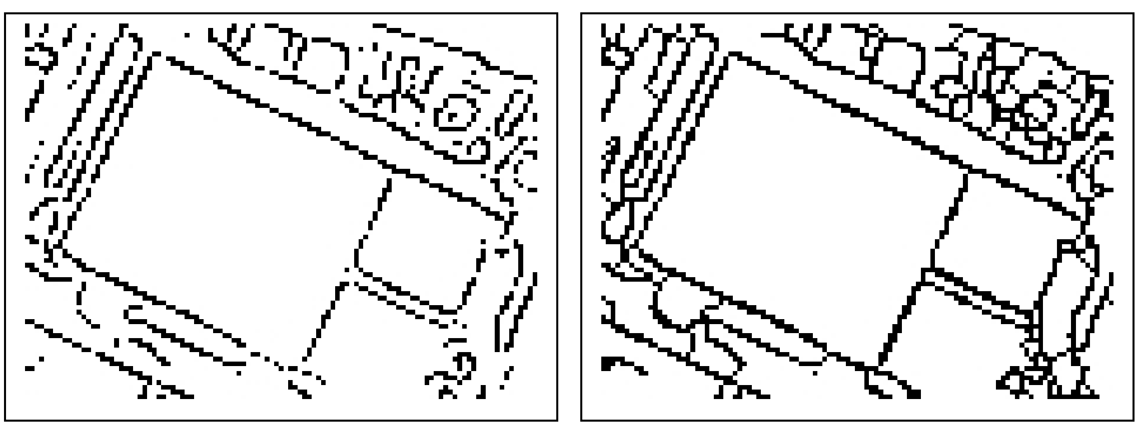
Figure 4-1 shows a case that there are two “open” points A and F. If the edge AB and DF are longer than a threshold, then extend these two edges to let them cross intersect.

If one edge is shorter than threshold (figure 4-2), say edge AB, then this edge can not be extended. We can draw a vertical line to edge BC from corner point B, and at the same time extend edge DE so that let them cross intersect at point F.

If two “open” edges are parallel (figure 4-3), for the longer one, draw a vertical line, for the shorter one, if it is longer than the threshold, extend it, if not, draw a vertical line, let them cross intersect at point E.

If two “open” edges are longer than threshold (figure 4-4), but their extended line can not intersect after extended for a distance threshold, draw vertical lines from the “open” point and let then intersect.

But actually for an edge, sometimes it is not a simple edge, it consists of several connected edges. In this case, we should detect its corners first, and then find how many edges is there on this “edge”, later check if this “edge” meets the conditions to be connected or not.



Before edge chaining After edge chaining

**Figure 5.** Edge chaining

Actually, many houses are very small, and each edge of the house sometimes is only several meters. In this case, the house edge just has several pixels in satellite image. Besides, the house edge sometime appears not in line but in curve. This makes our practical processing very difficult. Usually for an open point, we can hardly find that simple case in figure 3. Usually there are many edges and points around this point, so how to choose a point to connect it with the open point is really a problem. Is this case, we just select the nearest point for edge chaining. Figure 5 shows the detail of before connection and after connection.

### Block detection and corner detection

After edge connection, we first suppose building is a closed block in the edge connected image. So after edge connection, we only search the closed blocks.

For one point, first we check it whether it is edge point. If not, we suppose it is a building roof point. Later through region growing and point number counting, we determine whether the block is a building or not. Basically the size of a building on satellite image is within a suitable region. Here we set two thresholds for building size. After that, we detect building corners. Here for the number of building corners, we think it should be in a suitable region, so we set another two thresholds for corner number. After region growing and block searching, we conduct two steps of processing.

### Edge thinning and edge point sorting

The edge we detected usually is not a single pixel chained edge. But our later process for corner detection needs a single chained edge, so we first need to conduct edge thinning. We start this process from an edge point. Then search its 8 neighbor points with clockwise order. The first edge point is what we need. Then move to this edge point. Continue this process until we find the first edge point again. During this process, we give every edge point a sequence number according to its sequence being searched.

### Corner detection

Various corner detection algorithms have been developed. A number of frequently used approaches were discussed in the survey by Liu and Srinath (1990). Comparative experimental results were also given in the survey. The algorithms include Rosenfeld and Johnston (1973), Rosenfeld and Weszka (1975), Freeman and Davis (1977), and Beus and Tiu (1987). But these algorithms can not give accurate corner position and usually lose many corners (Dmitry, 1999).

Based on these algorithms, Dmitry Chetverikov and Zsolt Szabó developed a new algorithm: IPAN99 algorithm (1999). It is a fast and efficient algorithm for detection of high curvature points. But its fatal drawback of this algorithm is that it can only detect high curvature corners. For the low curvature corners, it is very difficult for this algorithm to detect them and usually this algorithm may give many wrong corners.

SUSAN and HARRIS are two corner detection algorithms generally mentioned by researchers. But SUSAN is only an ideal mathematical model, if we use it to detect corners in the satellite image, the result is ridiculous. Harris is not an accurate corner detector and usually gives many wrong corners.

According to the characteristics of digital image, we developed a new algorithm for corner detection. This method is very simple and effective. Figure 6 shows the result of corner detection. The black points on the edge are the corner points detected.

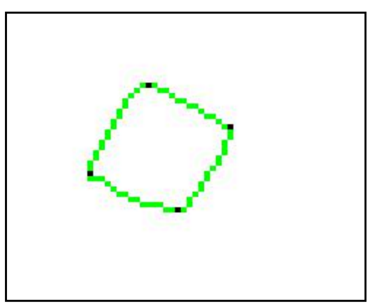


**Figure 6.** Detected corners

### 3D buildings extraction

After a building candidate is extracted in an image, we conduct image matching to look for its corresponding building in another image. Here we use a two step image matching strategy which contains roof matching and corner matching or edge matching and corner matching (Leong Keong Kwoh, Zhen Xiong, 2005). When the roof area is greater than a threshold, we adopt edge matching and corner matching. Otherwise, we use roof matching and corner matching.

The extracted building candidates are not all true buildings. Only those qualifying true building conditions can finally be used for 3D coordinates extraction.

## EXPERIMENT

(1) Data

We use a pair of level 1A IKONOS images which were acquired on June 25, 2004 in Penang, Malaysia (figure 7). We use the panchromatic image whose spatial resolution is 1 meter. The format of both left image and right image is 1039 by 958. These images are not full scenes. The detailed data clipping information is the following.

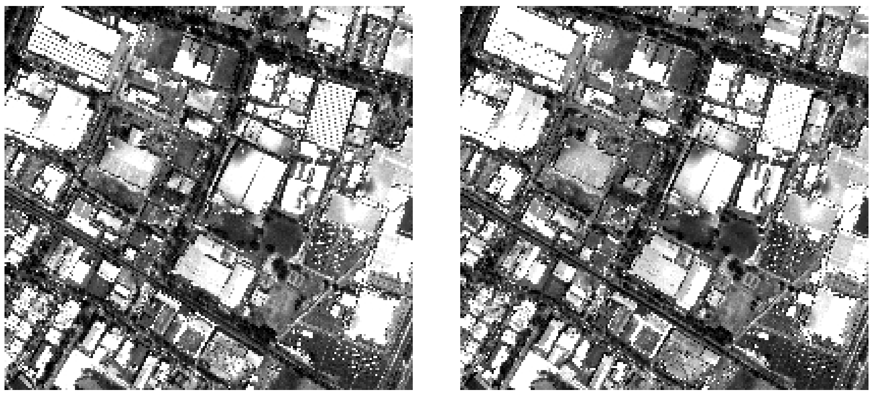


**Figure 7.** IKONOS stereo pair

Left image:
Upper Left:
P (column): 3878 L (row): 5926
Lower Right
P (column): 9125 L (row): 11891

Right image:
Upper Left:
P (column): 3029 L (row): 5143
Lower Right
P (column): 9023 L (row): 11982

(2) Satellite sensor model

Because the sensor model released by Space Imaging can not be directly used for photogrammetric task, so first we use 12 tie points to refine sensor model (Z. Xiong and Y. Zhang 2005). These tie points are selected in the full scene of images (table 1).

**Table 1. Image coordinates of tie points**

| Left | | Right | |
|---|---|---|---|
| P | L | P | L |
| 1063 | 157 | 238 | 55 |
| 12018 | 331 | 11168 | 341 |
| 12008 | 525 | 11159 | 532 |
| 6979 | 8503 | 6132 | 8509 |
| 6752 | 6629 | 5902 | 6642 |
| 2802 | 13726 | 1956 | 13729 |
| 1498 | 13847 | 650 | 13861 |
| 1050 | 13074 | 202 | 13084 |
| 12029 | 2840 | 11180 | 2852 |
| 5987 | 5929 | 5138 | 5940 |
| 1254 | 5566 | 408 | 5564 |
| 7824 | 3627 | 6973 | 3641 |

(3) The final result contains a full scene image of extracted buildings (figure 10) and a text file which records the coordinates of corners of all extracted buildings. The format of this text file is showed below as table 2.

**Table 2. Coordinates of building corners**

BLOCK NO: 1

| | Lat | Lon | T_h | B_h | cl(tl) | rw(tl) | cl(tr) | rw(tr) | cl(bl) | rw(bl) | cl(br) | rw(br) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5.330344 | 100.297638 | 8.71 | 2.00 | 4535.0 | 6946.0 | 3686.1 | 6956.2 | 4536.2 | 6942.2 | 3686.5 | 6956.2 |
| 2 | 5.330307 | 100.297737 | 8.71 | 2.00 | 4546.0 | 6950.0 | 3697.1 | 6960.2 | 4547.2 | 6946.2 | 3697.5 | 6960.2 |
| 3 | 5.330217 | 100.297707 | 8.71 | 2.00 | 4542.0 | 6960.0 | 3693.1 | 6970.2 | 4543.2 | 6956.2 | 3693.5 | 6970.2 |
| 4 | 5.330254 | 100.297607 | 8.71 | 2.00 | 4531.0 | 6956.0 | 3682.1 | 6966.2 | 4532.2 | 6952.2 | 3682.5 | 6966.2 |

BLOCK NO: 2

| | Lat | Lon | T_h | B_h | cl(tl) | rw(tl) | cl(tr) | rw(tr) | cl(bl) | rw(bl) | cl(br) | rw(br) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5.330312 | 100.297829 | 9.55 | 2.00 | 4556.0 | 6950.0 | 3707.2 | 6959.7 | 4557.3 | 6945.7 | 3707.6 | 6959.8 |
| 2 | 5.330230 | 100.298096 | 9.55 | 2.00 | 4585.0 | 6959.0 | 3736.2 | 6968.7 | 4586.3 | 6954.7 | 3736.6 | 6968.7 |
| 3 | 5.330112 | 100.298073 | 9.55 | 2.00 | 4583.0 | 6972.0 | 3734.2 | 6981.7 | 4584.3 | 6967.7 | 3734.6 | 6981.7 |
| 4 | 5.330094 | 100.298050 | 9.55 | 2.00 | 4580.0 | 6974.0 | 3731.2 | 6983.7 | 4581.3 | 6969.7 | 3731.6 | 6983.7 |
| 5 | 5.330077 | 100.297760 | 9.55 | 2.00 | 4548.0 | 6976.0 | 3699.2 | 6985.7 | 4549.3 | 6971.7 | 3699.6 | 6985.8 |

BLOCK NO: 3

| | Lat | Lon | T_h | B_h | cl(tl) | rw(tl) | cl(tr) | rw(tr) | cl(bl) | rw(bl) | cl(br) | rw(br) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5.330313 | 100.289818 | 15.42 | 2.00 | 3667.0 | 6955.0 | 2818.8 | 6961.3 | 3669.3 | 6947.3 | 2819.5 | 6961.4 |
| 2 | 5.330304 | 100.289864 | 15.42 | 2.00 | 3672.0 | 6956.0 | 2823.8 | 6962.3 | 3674.3 | 6948.3 | 2824.5 | 6962.4 |
| 3 | 5.329680 | 100.289818 | 15.42 | 2.00 | 3667.0 | 7025.0 | 2818.8 | 7031.3 | 3669.3 | 7017.3 | 2819.5 | 7031.4 |
| 4 | 5.329690 | 100.289764 | 15.42 | 2.00 | 3661.0 | 7024.0 | 2812.8 | 7030.3 | 3663.3 | 7016.3 | 2813.5 | 7030.4 |
| 5 | 5.329970 | 100.289787 | 15.42 | 2.00 | 3664.0 | 6993.0 | 2815.8 | 6999.3 | 3666.3 | 6985.3 | 2816.5 | 6999.4 |
| 6 | 5.330006 | 100.289780 | 15.42 | 2.00 | 3663.0 | 6989.0 | 2814.8 | 6995.3 | 3665.3 | 6981.3 | 2815.5 | 6995.4 |

In table 2, Block No is the number of extracted building. Lat is latitude and Lon is longitude, T_h is the height of building roof, and B_h is the height of ground. Cl(tl) and rw(tl) are column and row of roof corner in left image. Cl(tr) and rw(tr) are column and row of roof corner in right image. Cl(bl) and rw(bl) are column and row of bottom corner in left image. Cl(br) and rw(br) are column and row of bottom corner in right image.

## DISCUSSION AND ANALYSIS

As long as the building outline is correct, this building can be completely extracted. So in this situation, the question of 3D building extraction becomes a question of building outline detection. From the experiment result we can find that some extracted buildings are not completely extracted, some buildings have not been extracted, and some extracted buildings are not real buildings. We carefully checked these cases and found the reason.

(1) For the building not been completely extracted, in most of the cases, the gray distribution of building roof is not even. So during edge detection and connection, the building may be separated into several parts. Later the bright parts will be extracted and the dark parts will not, because the dark part does not meet the threshold of mean gray value for building roof.

(2) For the building not been extracted, generally there are two reasons. One is the mean gray value of building roof is too small and not meet the gray threshold for building roof. Another reason is the gray distribution of building roof is not even, some bright and some dark. So that the building roof is separated into several

parts and all these parts are too small and do not meet the threshold of building size and can not be extracted.

(3) For the extracted building is not a real building, it is an area of ground. This is because this ground area is bright and very high, meets all the conditions we set for building detection.

From the analysis above, we can find the ways to improve our building extraction result in our future work.

(1) Besides panchromatic image, we can use color information or spectral feature of multi-spectral image to segment image so that to remove grass, trees and other vegetables, reduce their affection to building extraction.

(2) Use a DSM or generate a DSM first, so that to avoid ground rectangular be detected as building and combine neighbor parts which have the same height to a big one based on suitable reasoning.

(3) Use more effective constraints to identify buildings.

## CONCLUSION AND FUTURE WORK

This research developed a full operational strategy for automatic 3D building extraction based on stereo Ikonos images. Although the result is far from perfect, but most of the flat buildings can be extracted. The program is fast and effective. Another most significant point is that it can automatically detect building corners and can deliver corner coordinates. Of course, there are obviously many drawbacks. In our experiment result, many buildings are not completely extracted, some extracted buildings are not real buildings and some buildings are not extracted. But we think these problems can be overcome gradually if we continually improve our techniques according to the analysis above.

**Figure 8.** Extracted buildings

## REFERENCES


Andre Fischer, Thomas H. Kolbe, Felicitas Lang, Armin B. Cremers, Wolfgang Forstner, Lutz Plumer and Volker Steinhage (1998). Extracting Building from Aerial Images Using Hierarchical Aggregation in 2D and 3D, *Computer Vision and Image Understanding*, Vol. 72, No.2, Novermber, pp.185-203.

Beus H.L. and Tiu S.S.H (1987). An improved corner detection algorithm based on chain-coded plane curves. *Pattern Recognition*, 20:291-296.

Braun C., Kolbe T. H., Lang F., Schickler W., Steinhage V., Cremers A. B., Forstner W., and Plumer L (1995). Models for Photogrammetric Building Reconstruction, *Computer & Graphics*, Vol.19, No.1, pp.109-118.

Donna Haverkamp (2004). Automatic Building Extraction From IKONOS Imagery, *Department of Research and Product Development, Space Imaging* LLC, Thornton, CO 80241, US

Englert, R. (1998). *Learning Model Knowledge for 3D Building Reconstruction*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Institute of Computer Science III, Bonn, Germany.

Freeman H. and Davis L.S (1977). A corner finding algorithm for chain-coded curves. *IEEE Trans. Computers*, 26:297-303.

Gruen, A. and Wang, X. (1998). CC-Modeler: A Topology Generator for 3-D City Models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5):286-295.

Lin Chungan and Ramakant Nevatia, (1998). Building Detection and Description from a Single Intensity Image, *Computer Vision and Understanding*, Vol.72, No.2, November, pp.101-121.

Liu H.-C. and Srinath M.D. (1990). Corner detection from chain-code. *Pattern Recognition*, 23:51-68.

Müller, H. (1998). Experiences with Semiautomatic Building Extraction. In *Third Course in Digital Photogrammetry*, Bonn, Germany. Institute for photogrammetry at Bonn University and Landesvermessungsamt Nordrhein-Westfalen.

Paparoditis N., Cord M., Jordan M., and Cocquerez J.-P. (1998). Building Detection and Reconstruction from Mid- and High-Resolution Aerial Imagery. *Computer Vision and Image Understanding*, Vol.72, No.2, November, pp.122-142.

Rosenfeld and E. Johnston (1973). Angle detection on digital curves. *IEEE Trans. Computers*, 22:875-878.

Rosenfeld and J.S. Weszka. 1975. An improved method of angle detection on digital curves. *IEEE Trans. Computers*, 24:940-941.

Rottensteiner, F (2000). Semi-automatic building reconstruction integrated in strict bundle block adjustment. In *Proceedings of the XIX$^{th}$ ISPRS Congress*, volume XXXIII-B2 of *International Archives of Photogrammetry and Remote Sensing*, pages 461-468, Amsterdam.

Uwe Weidner and W. Forstner (1995). Towards Automatic Building Extraction from High-Resolution Digital Elevation Models, ISPRS *Journal of Photogrammetry and Remote Sensing*, 50(4):38-49

Veldhuis, H. (1998). Performance Analysis of two Fitting Algorithms for the Measurement of Parametrised Objects. In Schenk, T. and Habib, A., editors, *Proceedings ISPRS Commission III Symposium*, volume XXXII-3/1 of *International Archives of Photogrammetry and Remote Sensing*, pages 400-408, Columbus, OH.

Yi Hui Lu, John Trunder and Kurt Kubik (2002). Automatic Building Extraction for 3D Terrain Reconstruction Using Interpretation Techniques, School of Surveying and Spatial Information Systems, University of New South Wales, NSW 2052 Australia

Zhen Xiong, Yun Zhang (2006). An Initial Study of Moving Target Detection Based on High Spatial Resolution Satellite Images. *PE & RS* (under reviewing).

Leong Keong Kwoh, Zhen Xiong (2005). Building Height Extraction from Ikonos Stereo Images, *IGARSS 2005*, Seoul, Korea.

Dmitry Chetverikov and Zsolt Szabó (1999). Detection of High Curvature Points in Planar Curves.

http://visual.ipan.sztaki.hu/corner/

http://www.geocities.com/ResearchTriangle/Campus/7971/susan.htm

http://gandalf-library.sourceforge.net/tutorial/report/node122.html

http://www.ipf.tuwien.ac.at/fr/buildings/diss/node16.html

J. dash, E. Steinle, R.P. Singh, H.P. Bahr (2004). Automatic building extraction from laser scanning data: an input tool for disaster management. *Advances In Space Research*.

Michel Morgan, Ayman Habib (2001). 3D Tin For Automatic Building Extraction From Airborne Laser Scanning Data. Civil and Envionmental Engineering and Geodetic Science Department, Ohio State University, USA.

Zhan Qingming, Martien Molenaar, Klause Tempfli (2004). Building Extraction From Laser Data By Reasoning On Image Segments In Elevation Slices. *International Institute for Geo-Information Science and Earth Observation* (ITC), The Netherlands.