AUTOMATED 3D BUILDING MODELS GENERATION USING MULTIDIRECTIONAL SCANNING MODEL

Ismail Rakip Karas[1], Fatmagul Batuk[2] and Alias Abdul-Rahman[3]

[1] Department of Computer Science,
Karabuk University, Karabuk, Turkey
ismail.karas@karabuk.edu.tr

[2] Department of Geodesy and Photogrammetry Engineering,
Yildiz Technical University, Istanbul, Turkey
batuk@yildiz.edu.tr

[3] Department of Geoinformatics,
Universiti Teknologi Malaysia, Johor, Malaysia
alias@utm.my

Abstract

In this paper, the automated 3D model of buildings generation from raster plans by using MUSCLE Model (Multidirectional Scanning for Line Extraction) is described. The algorithm of the model generates the line thinning and the simple neighborhood techniques for vectorization processes. Then, the 3D building model is configured according to the floor numbers. The results indicate that the method successfully generate the models of the buildings and creates a 3D geo-database. The outcome of this research could be utilized for future 3D GIS applications.

Key words: MUSCLE, 3D, Building Model, 3D GIS, Vectorization, Line Extraction, Line Thinning

1. Introduction

Interiors of buildings are often represented as two-dimensional spaces with attributes attached to them. Therefore, most of the available GIS programs use primarily 2D plans for visualization and features for indoors. These programs have not been structured with respect to the functionality of the building, but largely with respect to the navigation/visualization routes. However, in the last few years, there has been an increasing interest in modeling based on functionality of interiors (Meijers et al, 2005). Pu and Zlatanova (2005) have pointed out that automatically extracting geometry and logic models of a building is difficult and the nodes and links have to be created manually or half-manually.

Data collection used to be the major task which consumed over 60% of the available resources since geographic data were very scarce in the early days of GIS technology. In most recent GIS projects, data collection is still very time consuming and expensive task; however, it currently consumes about 15-50% of the available resources (Longley et al, 2001). In order to reduce the cost on data collection, data generation from existing archives and plans has been widely applied. Scanning method - where analogue data from the archives can be transformed into digital raster format data.

Most of geographical information systems work with raster images data such as scanned maps and engineering drawings in CAD format. In order to manipulate, for example transform or select the lines and the other features from such raster images, these features must be extracted through a vectorization process (Nieuwenhuizen et al, 1994). Vectorization is quite important in document recognition, line detection, mapping, and drawing applications (Zhong, 2002). For advanced vectorization applications, the raster images must have high accuracy to preserve the original shapes of the graphical objects with the highest extent possible (Dori and Wenyin, 1999).

Line is one of the most fundamental elements in graphical information systems. Line detection is a common and essential task in many applications such as automatic navigation, military surveillance, and electronic circuits industry (Shpilman and Brailovsky, 1999 - Climer and Bhatia 2003). In previous studies, there are a large number of algorithms developed for detecting lines from raster images (Miao et al, 2002 - Lagunovsky and Ablameyko, 1999 - Madhvanath et al, 1999 - Hori and Tanigawa, 1993). The vectorization methods implemented in these algorithms can be categorized into following six classes; (1) Hough Transform (HT) based methods, (2) thinning based methods, (3) contour based methods, (4) run-graph based methods, (5) mesh pattern based methods, and (6) sparse pixel based methods (Wenyin and Dori, 1999). After the scanning, thresholding, and filtering stages, a traditional vectorization process consists of three stages (except HT based methods); (1) line thinning, (2) line following and chain coding, and (3) vector reduction (i.e. line segment approximation). In order to determine only the important points representing the medial axis, the lines on the image are to be thinned to one pixel wide by using the kernel processing (Treash and Amaratunga, 2000). Once line thinning stage is performed, the second stage is following and chain coding the medial axis. In this stage, tracing process starts at an end pixel and continues based on the chain code directions until the last pixel in the line is reached. At the third stage, the medial axis coded in the second stage is examined and the vectors in the chain code are identified. In this process, the long vectors that closely represent the chain codes are formed while considering a user defined maximum deviation of the vectors from the chain codes (Jennings, 1993).

In this paper, the automated 3D model of buildings generation from raster plans by using MUSCLE Model (Multidirectional Scanning for Line Extraction) is described. The algorithm of the model generates the line thinning and the simple neighborhood techniques for vectorization processes. Unlike traditional vectorization process, this model generates straight lines based on a line thinning algorithm, without performing line following-chain coding and vector reduction stages.

Architectural plans of the buildings are generally either drawn on a paper as blueprints or stored in vector format. Blueprints can be stored in raster formats after scanning process. Whether the floor plan information is in raster or vector format, or even drawn on a form by a user, this model can be applied in any case. The key point is that the images of the floor plan should be represented as image on the GUI. Then the models are generated automatically by analyzing and processing those images. Therefore, the method is basically an image processing method. The 3D building model is configured according to the floor numbers. The results indicate that the model successfully generate the models of the buildings and creates a 3D geo-database. The outcome of this research could be utilized for future 3D GIS applications.

2. The Methodology

The following main stages in the model are described in this section:

1. Threshold processing
2. Horizontal and vertical scanning of the binary image
3. Detecting wrongly vectorized lines
4. Correcting wrongly vectorized lines by using diagonal scanning
5. Generation of 3D Building Model
6. Creating 3D Geo-Database

2.1. Threshold Processing

In greyscale images, the objects may contain many different levels of grey tones. In this study, the objects are separated by using the threshold processing technique, with the assumption that the grey values are distributed over the image nearly homogeneous (Belkasim et al, 2003). In the threshold process, a predetermined grey level (threshold value) is to be determined and every pixel that is darker than this level is assigned black, while every lighter pixel is assigned white. Therefore, the greyscale image was converted into a binary image (Jennings, 1993).

2.2. Horizontal and Vertical Scanning of the Binary Image

In this stage, the horizontal and vertical lines were extracted from the binary image. The nearly vertical lines were obtained by scanning the images horizontally, while the nearly horizontal lines were obtained by scanning the images vertically. The forms of nearly vertical and nearly horizontal lines are shown in Fig-1. In Fig-1a, the lines which pass through the region 1 and 2 are defined as the nearly vertical lines and the nearly horizontal lines, respectively. Fig-1b and Fig-1c indicates the sample drawings for nearly vertical and nearly horizontal lines, respectively. In other words, if the slope (tangent) of the line is between -1 and +1, it is defined as “nearly horizontal line”. If the slope (tangent) of the line is less than -1 or greater than +1, it is defined as “nearly vertical line”.

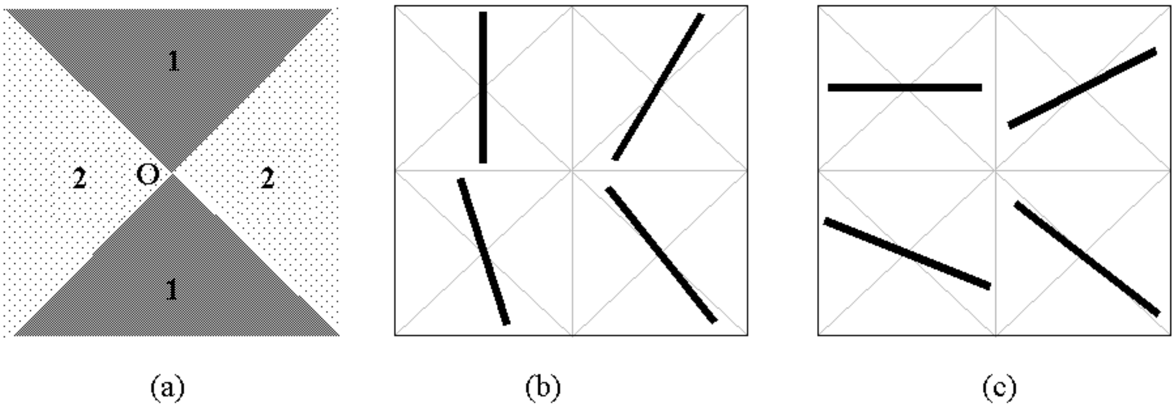


Figure 1. Samples for nearly vertical and nearly horizontal lines.

At the first step, each row on the binary image was scanned horizontally to determine the thickness of the lines and the position of the pixels, which were located in the mid-point of the lines. During this process, the value (black or white) of each pixel was checked by moving from left to right. Once the first black pixel was met, its column number was stored into the algorithm. While continuing to scan pixels, the column number of the first white pixel was also stored into the algorithm. Thus, the position of the middle pixel in the mid-point of the line could be determined by using the following equation, based on the image coordinate system:

$$\textit{The position of the middle pixel} = m + \textit{Absolute Value}\ ((n - m) / 2) \qquad (1)$$

m : column number of the first black pixel
n : column number of the first white pixel

For example, assuming that 8^{th} pixel is the first black pixel and 13^{th} pixel is the first white pixel in Fig-2a. Using Equation 1, position of the middle pixel can be calculated as 10^{th} pixel, which is then colored with red. After performing the same process for each row on the image, distribution of the red pixels for nearly vertical and nearly horizontal lines are indicated in Fig-2a and Fig-2b, respectively. In these figures, the distribution of the red pixels indicates that the red pixels have continuity for nearly vertical lines; however they have discontinuity for nearly horizontal lines.

After the horizontal scanning processes were completed, only the red pixels were selected. Then, a neighborhood analysis was carried out based on the nearly vertical lines by taking the advantages of discontinuity on the nearly horizontal lines. In this method, a red pixel, which is adjacent to another red one, was searched along the lines. This process continued until no red pixels were found adjacent to each other, indicating that the end of the line has been reached. The beginning and ending points of all the nearly vertical lines were determined by using the same procedure.

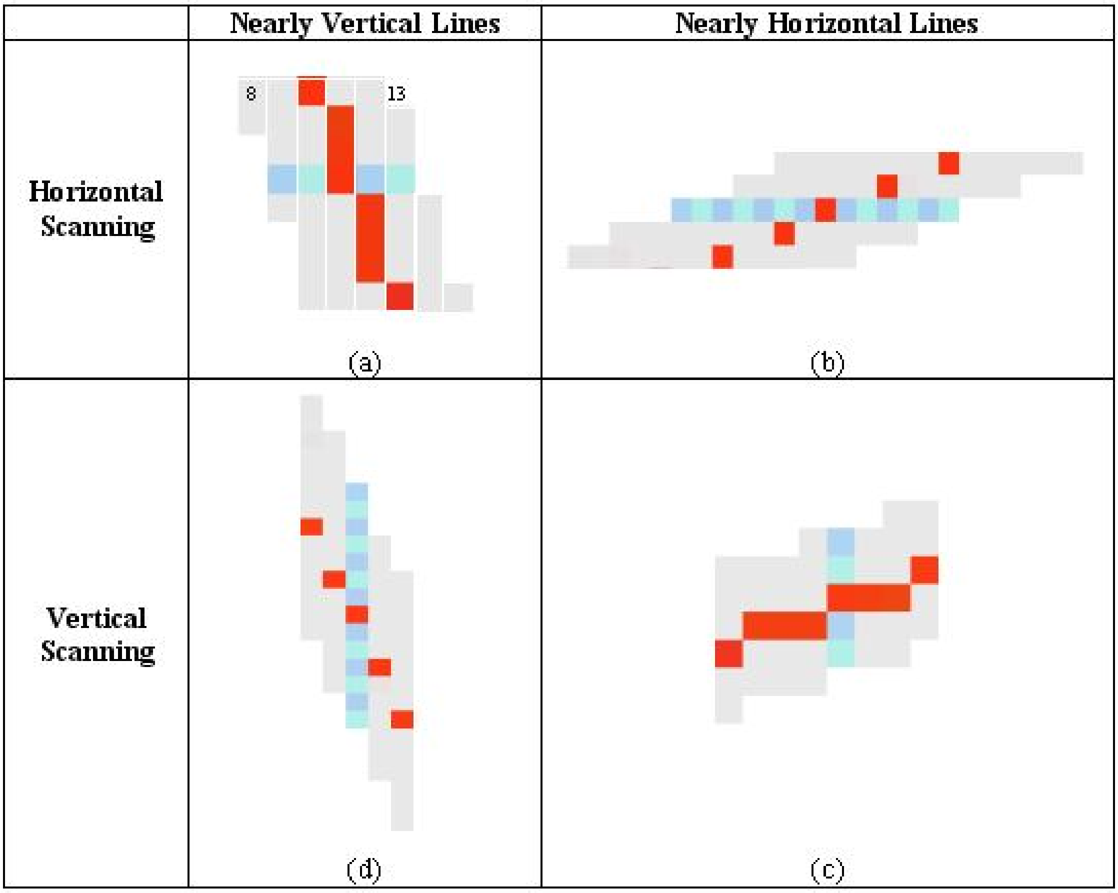


Figure 2. Determining red pixels by using horizontal and vertical scanning process.

At the second step, the binary image was scanned vertically, and then, the same process described above was carried out for all columns. Unlike horizontal scanning, the red pixels have continuity for nearly horizontal lines (Fig-2c); however they have discontinuity for nearly vertical lines (Fig-2d). Therefore, the neighborhood analysis was carried out based on the nearly horizontal lines and the beginning and the ending points of all the plenary horizontal lines were determined. After completing the horizontal (Fig-3a) and vertical (Fig-3b) scanning of the binary image, the final vectorized data (Fig-3c) was generated by vectorizing the nearly vertical and the horizontal lines.

2.3. Detecting Wrongly Vectorized Lines

In a case where two or more consecutive lines are nearly horizontal or nearly vertical, raster data becomes unmanageable and the process described in the previous stages generates wrongly vectorized lines. For example, initially, three consecutive nearly horizontal lines (AB, BC, and CD) were horizontally scanned as displayed in Fig-4a. Due to discontinuity of the red pixels between intersection points A, B, C, and D, the neighborhood analysis can not be performed and vectorized data can not be generated. When the raster image was vertically scanned during the second step, the neighborhood analysis yielded wrong vectorization results because of continuity of the red pixels. The algorithm recognizes point A as the beginning point of the line, skips point B and point C, and ends the line at point D. Therefore, the process generates a wrongly vectorized line between point A and D as indicated in Fig-4b.

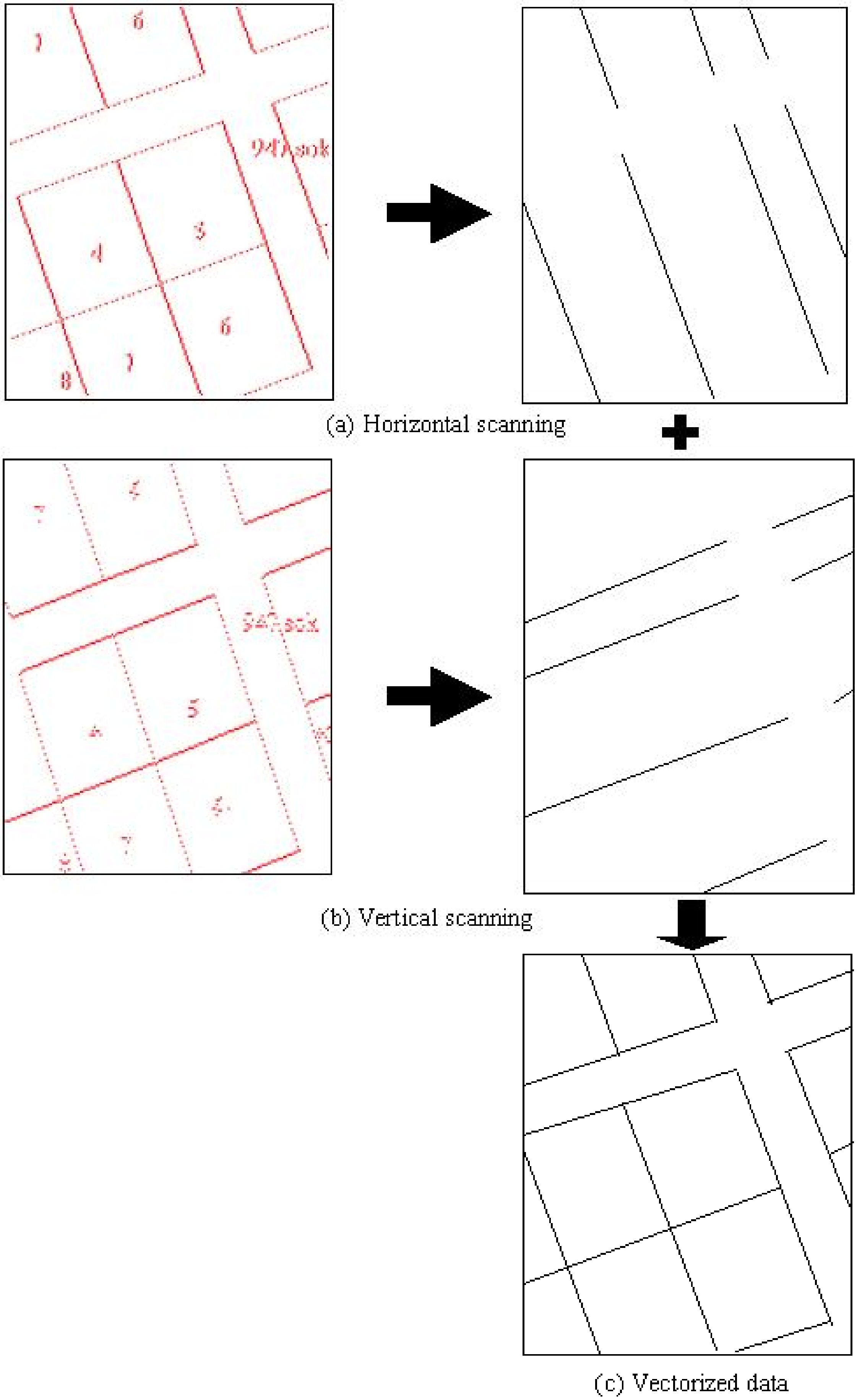


Figure 3. Horizontal and vertical scanning in vectorization process.

The detection of wrongly vectorized data is performed by comparing the middle axis of the lines (red pixels) with the vectorized lines. The middle axis and the vectorized line have to be based on the same linear equation. For example, if a sample vectorized line (AB line) is a line with the beginning point of A(x_a,y_a) and the ending point of B(x_b,y_b), then, the linear equation for this vectorized line can be formed as follows:

$$(Y- y_a) / (y_a - y_b) = (X- x_a) / (x_a - x_b) \tag{2}$$

$$Y = ((y_a - y_b) / (x_a - x_b))\ X + ((y_b\ x_a - x_b\ y_a) / (\ x_a - x_b)) \tag{3}$$

When X coordinate of a red pixel is inserted into the linear Equation 3, and if the difference between the Y value derived from this equation and the Y coordinate of this pixel is greater than a user defined maximum deviation, the model defines this line as a wrongly vectorized line. After this process, the red pixels within the acceptable deviation range were eliminated from the image by converting them into the white pixel values. The wrongly vectorized lines with red pixels were remained unchanged within the image.

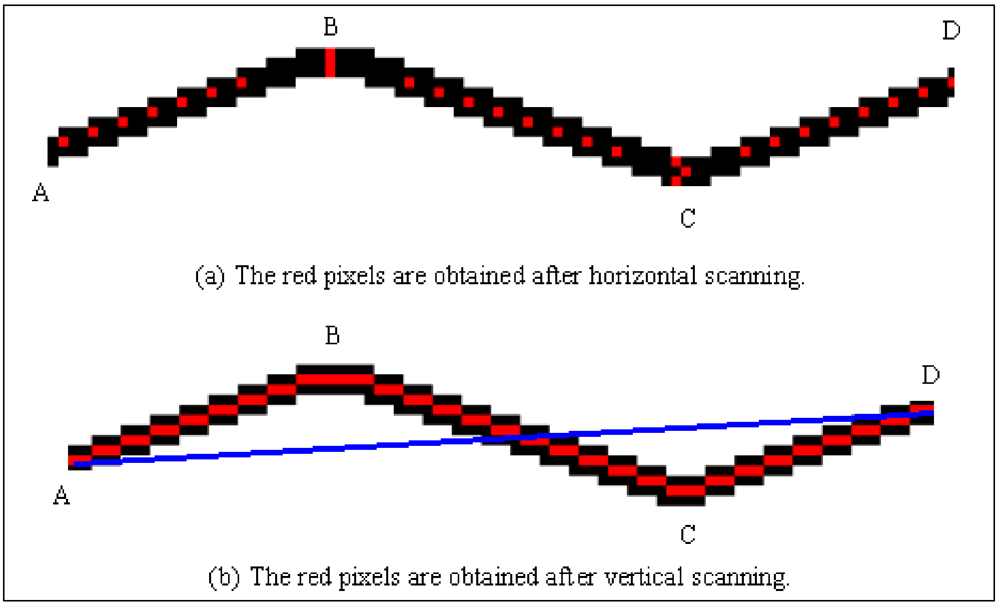


Figure 4. Detecting wrong vectorization after vertical and horizontal scanning.

2.4. Correction of wrongly vectorized lines by using diagonal scanning

The image having the wrongly vectorized lines (Fig-5a) was diagonally (under 45^o angle) scanned; first, from left to right, and then, from right to left (Fig-5b). In diagonal scanning process, if there were two consecutive red pixels along the direction of scanning, the second red pixel is eliminated. Thus, vectorized line took a discontinuous form as shown in Fig-5c. After applying the neighborhood analysis, the lines failed to have the acceptable number of pixels were not vectorized. The continuous pixels, determined by implementing diagonal scanning from both directions, were vectorized as indicated in Fig-5d. Then, corrected vector data was generated by combining both of the vectorized lines together (Fig-5e).

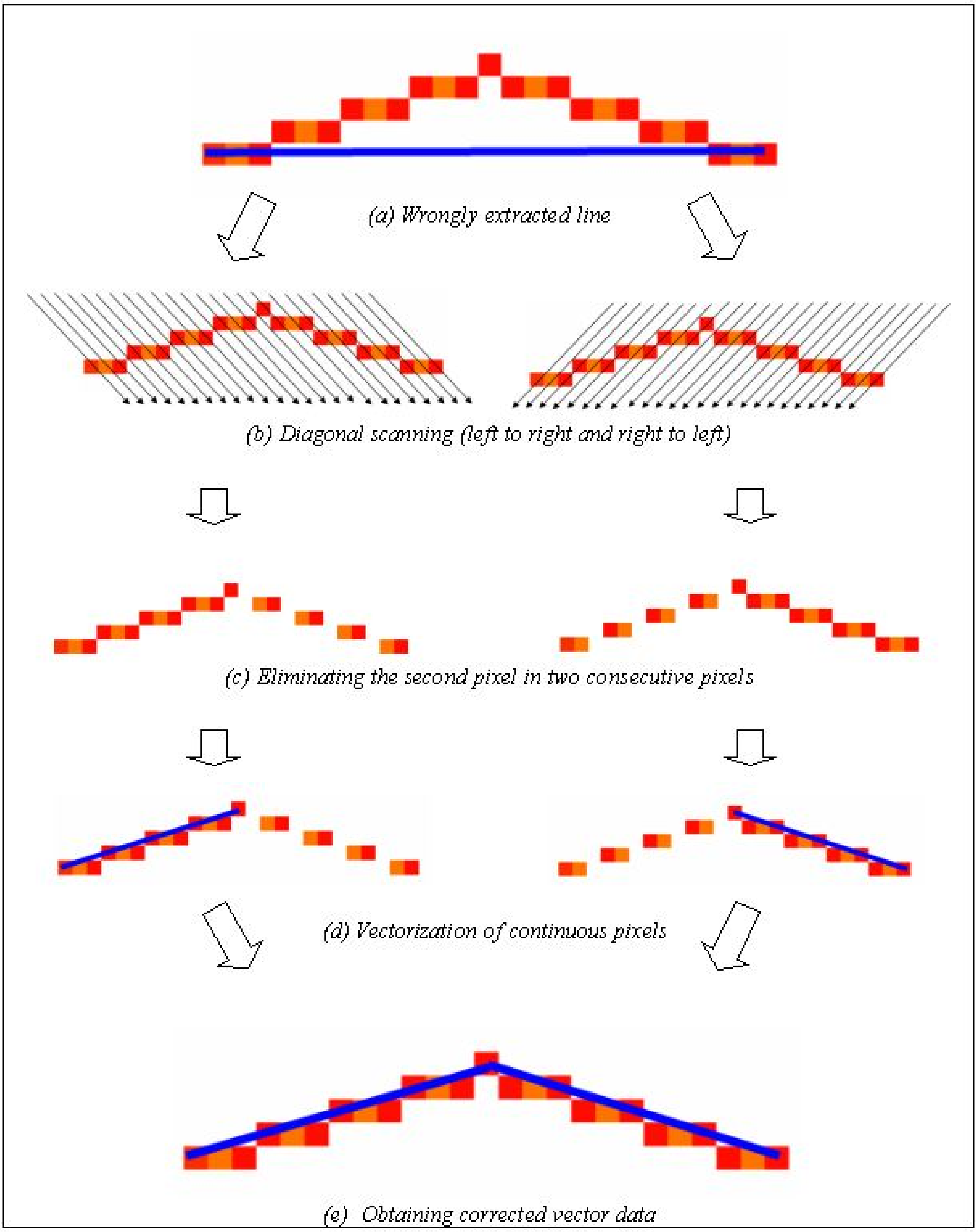
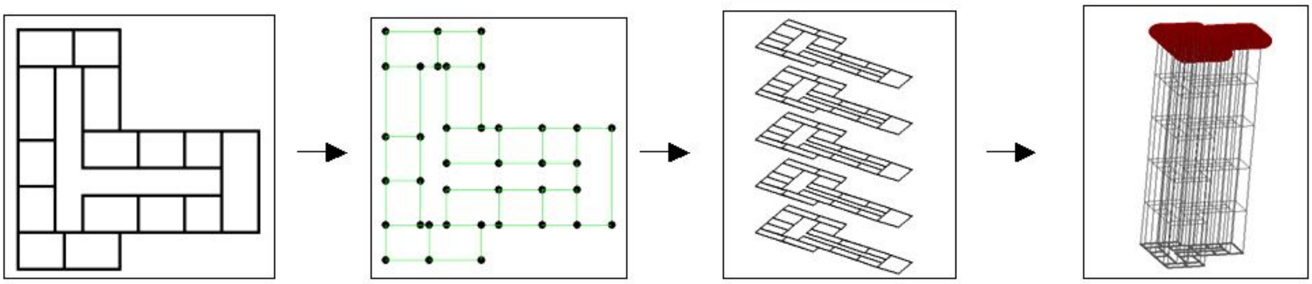


Figure 5. Correction of wrong extracted lines by using diagonal scanning procedure.

3. Generation of 3D Building Models

By using the MUSCLE Model (Multidirectional Scanning for Line Extraction) as described in the preceding sections, the 3D model of buildings can be generated automatically from raster floor plans and thus a 3D Geo-Database. We believe that, the MUSCLE Model described in this paper can be used to generate 3D building model automatically as shown in Fig-6. Detail discussions and descriptions of this new technique of generating 3D model could be found in Karas et al, 2007. As shown in Fig-6, first, lines are determined. Then lines and points, both generate the floor plan, are obtained by determination of intersection points using optimization process (Fig-6b). Based on user defined data such as number of floors and heights, 3D building model can be generated after designing each floor automatically by assigning different elevation values to floor plan (Fig-6c and Fig-6d). Then, features of the building are automatically stored in the created 3D Geo-Database.



(a) (b) (c) (d)

Figure 6. Generation of 3D Building Models.

4. Conclusions

In this study, MUSCLE Model (Multidirectional Scanning for Line Extraction) was used to automatically extract 3D Building Model from architectural plan of a building. The algorithm of the model generates the line thinning and the simple neighborhood techniques for vectorization processes. Unlike traditional vectorization process, this model generates straight lines based on a line thinning algorithm, without performing line following-chain coding and vector reduction stages. The results indicate that the model successfully generate the models of the buildings and creates a 3D geo-database. The outcome of this research could be utilized for future 3D GIS applications.

References

Belkasim, S., Ghazal, A., Basir, O.A. (2003). "Phase-based optimal image thresholding." Digital Signal Processing, 13, 636–655.

Climer, S., Bhatia, S. K. (2003). "Local Lines: A linear time line detector." Pattern Recognition Letters, 24, 2291–2300.

Dori, D., Wenyin, L. (1999). "Automated CAD Conversion with the Machine Drawing Understanding System: Concepts, Algorithms, and Performance." IEEE Transactions On Systems, Man, And Cybernetics - Part A: Systems And Humans, 29(4), 411-416.

Hori, O., Tanigawa, S. (1993). "Raster-to-vector Conversion by Line Fitting Based on Contours and Skeletons." *Proc., Int. Conf. Document Analysis and Recognition*, Tsukuba (Japan), 353-358.

Jennings, C. (1993). "Computer Vision for Line Drawings." MSc Thesis, University of Calgary.

Karas, I.R., Batuk, F., Emem O. (2007), "A 3D Mobile Mapping System: Data Generation, Network Analysis, Simulation And Navigation", 5th International Symposium on Mobile Mapping Technology (MMT'07), Padua, Italy.

Lagunovsky, D., Ablameyko, S. (1999). "Straight-line-based primitive extraction in grey-scale object Recognition." Pattern Recognition Letters, 20(10), 1005-1014.

Liao, P.S., Chen, T.S., Chung, P.C. (2001). “A Fast Algorithm for Multilevel Thresholding.” Journal of Information Science and Engineering, 17, 713-727.

Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. (2001). “GIS Data Collection”, *Geographic Information Systems and Science*, Hoboken, NJ: John Wiley & Sons, 203-224.

Madhvanath, S., Kim, G., Govindaraju, V. (1999). “Chaincode Contour Processing for Handwritten Word Recognition.” IEEE Transactions On Pattern Analysis And Machine Intelligence, 21(9), 928-932.

Meijers M., Zlatanova, S. and Preifer, N., 2005. 3D geoinformation indoors: structuring for evacuation. in: Proceedings of Next generation 3D city models, 21-22 June, Bonn, Germany, 6 p.

Miao, L., Liu, X., Peng, Q., Bao, H. (2002). “BRDC: binary representation of displacement code for line.” Computers & Graphics, 26(3), 401–408.

Nieuwenhuizen, P.R., Kiewiet, O., Bronsvoort, W.F. (1994). “An Integrated Line Tracking and Vectorization Algorithm.” Eurographics'94, 3(3), 349-359

Pu, S. and Zlatanova, S., 2005. Evacuation route calculation of inner buildings, in: PJM van Oosterom, S Zlatanova & EM Fendel (Eds.), Geo-information for disaster management, Springer Verlag, Heidelberg, pp. 1143-1161

Shpilman, R., Brailovsky, V. (1999). “Fast and robust techniques for detecting straight line segments using local models.” Pattern Recognition Letters, 20(9), 865-877.

Treash, K., Amaratunga, K. (2000). “Automatic Road Detection in Greyscale Aerial Images”, J. Comp. in Civ. Engrg., 14(1), 60-69.

Wenyin, L., Dori, D. (1999). "From Raster to Vectors: Extracting Visual Information from Line Drawings." Pattern Analysis and Applications, 2(2), 10-21.

Zhong, D. X. (2002). “Extraction of embedded and/or line-touching character-like objects Pattern Recognition”, 35(11), 2453 – 2466.

Paper Reference Number:
Title of the paper : AUTOMATED 3D BUILDING MODELS GENERATION USING MULTIDIRECTIONAL SCANNING MODEL
Name of the Presenter : Dr. Ismail Rakip Karas
Author(s) Affiliation : Karabuk University, Karabuk, Turkey.
Mailing Address : 3D GIS Lab, FKSG, Universiti Teknologi Malaysia, Johor, Malaysia
Email Address : irkaras@gmail.com
Telephone number(s) : +6010.7014051
Fax number (s) :

Authors' Brief Biography:

Dr. Ismail Rakip Karas
Dr. Ismail Rakip Karas is Asst. Professor in Computer Science Department at Karabuk University, Turkey. He is currently continuing his researches in 3D GIS Lab FKSG, UTM. His research is focused on 3D Geographic Information Systems in particular 3D Network analysis. He received BSc degree from Selcuk University, MSc degree from Gebze Institute of Technology, and PhD degree from Yildiz Technical University in Geodesy and Photogrammetry Engineering in Turkey.

Prof. Dr. Fatmagul Batuk
Prof. Fatmagul Batuk is currently a Professor in Geodesy and Photogrammetry Engineering department at Yildiz Technical University of Turkey. Her expertise is on Geographic Information Systems and Photogrammetry. She holds BSc, MSc, and PhD of Geodesy and Photogrammetry Engineering from Yildiz Technical University, Turkey.

Prof. Dr. Alias Abdul-Rahman
Prof. Alias Abdul-Rahman is academic member of Universiti Teknologi Malaysia (UTM), works on 3D GIS for his professional activities, and as Chair for ISPRS WG II/5 on MultiDimensional GIS and Mobile Data Model.