

STUDY ON 3D TEXTURED BUILDING MODEL BASED ON ADS40 IMAGE AND 3D MODEL

Liu Zhen ^{a,b,*}, Gong Peng ^c, Shi Peijun ^a, Sasagawa T ^d

^a College of Resource Science & Technology, Beijing Normal University, 100875, liuzhen@bnu.edu.cn

^b Center of Information & Network Technology, Beijing Normal University, 100875

^c Department of Environmental Science, Policy, and Management, California University at Berkeley, USA

^d Institute of GIS, Pasco Corp, Tokyo

KEY WORDS: 3D Textured model, ADS40 image, Feature extraction, Feature matching

ABSTRACT:

An automatic method of 3D textured building modelling with an airborne linear scanner sensor image and 3D model is proposed, which includes to extract buildings' feature from image, to match the extracted feature with 3D model and to map texture according to the correspondence between image and model. Feature extraction from image and feature matching with 3D model are two key steps to build 3D textured building model automatically. In the article, a new line extraction approach through multi-level feature filter is presented, which is consisted of Canny edge detector, edge phase filter, edge direction filter with fault tolerance, Hough transformation, neighbour line segment fusion. Based line segmentation, a parallelogram extraction algorithm applying perceptual organization is proposed, which is based on perceptual organization, employing uncertainty reasoning and a shape expression form based on "part" to obtain simplicity, low computational overhead, and noise free. Matching between 2D image and 3D model is in essence to find a transformation matrix to minimize the error. A lot of algorithms are presented to solve the match problem with several unknown parameters, but still no good solution to problems with too much unknown parameters. Projected image will first be outputted based on camera model after partial matching between parallelogram extracted and 3D building. Then Hausdorff distance is calculated between edge image and projected image. Based on which, feature matching is finished to extract texture and to map texture to buildings.

1. INTRODUCTION

Virtual Reality based on 3D building are successfully employed today for real-time visualization in such diverse fields as urban planning, city planning and architectural design. Visualisation of 3D building is a general task of nearly all science and engineering disciplines to easily assess complex object features, object grouping, environmental goodness-of-fits and many other applications. Texture mapping for 3D building becomes an issue when 3D building reconstruction is desired. It is out of question that 3D textured buildings can manually and semi-automatically reconstructed using photogrammetric stereo restitution processes. Nowadays, many software packages offer methods for rectification and texture mapping, but very often the handling is time-consuming and circumstantial^[1]. There are a few example of automation 3D textured building modelling. Previous research especially from University of South California and University of Stuttgart, has found that because of noise, occlusion and lack of camera information, extracting buildings from urban area monocular aerial images is not feasible^[2]. While matching 3D models with 2D images is also difficult without knowing anything about the camera parameters.

An automatic method of 3D textured building modelling with an airborne linear scanner sensor image and 3D model is proposed, which includes extracting buildings' features from image, to match extracted features with 3D model and to map texture according to the correspondence between image and model. Feature extraction from image and feature matching

with 3D model are two key steps to build 3D textured building model automatically. In the paper, a parallelogram extraction algorithm based on the theory of perceptual organization, employing uncertainty reasoning and a shape expression form based on "part" to obtain simplicity, low computational overhead, and noise free. The methods are illustrated with an airborne linear scanner sensor image over the downtown of Shinjuku in Tokyo city, Japan. The result of building texture mapping explains the method of building outline extraction and match between 2D feature and 3D model very well.

2. DATA USED

The multi-spectral data used were acquired by ADS40 SP1 with the resolution of 0.2m, which has 4 bands, namely NIR, R, G and B channel covered the downtown of Shinjuku of Tokyo. The other data used is 3D model CAD file (DXF) of this region.

3. APPROACH

3.1 Edge extraction

Edge Detector

The Canny edge detector has been shown to be optimal for images corrupted by Gaussian white noise^[3], and is used to

* Corresponding author.

detect edges in the current work. It is demonstrably more effective than the LoG operator, Sobel operator and others. Using Canny Edge detector, we obtain not only edge intensity information, but also edge phase, which is used in the latter step.

Phase filter

It reasonably assume that neighbour pixels in a straight contour of a man-made object would have similar phase. Thus we designed the algorithm as follows:

Label edge pixels with phase: For each pixel, label surrounding pixels 0, 1, 2...7, and assign each its phase. We needn't calculate an inverse trigonometric function for each edge pixel, but only apply addition and multiplication on (derivative in X axis) , (derivative in Y axis) and invariant $\tan(\pi/8)$, to reduce the computational overhead.

Track edge: For each pixel P, check its 8 neighbouring pixels. If a neighbour pixel Pn has a phase similar to P, then we go on tracking along Pn. If this tracking results in too few pixels decide by threshold value, we can reasonably judge it to be a noise edge and delete it.

Edge filter according direction

The above filter process has no effect on curve edges whose pixels phases don't change abruptly. Therefore, further filtering is necessary.

For a line-supported edge, connections between neighbour pixels are typically in the same direction. Figure 1 shows that neighbour pixels only have 8 connection directions at most and a reciprocal oscillation phenomenon would appear.



Figure 1-2 Direction oscillate.

As a line is tracked, a direction fault tolerance is applied as Table 1.

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | | 3 |
| 2 | 1 | 0 |

Table 1 Line track direction

I) For an edge pixel P, tracking in a given direction. For each neighbour pixel located on the D direction of P, if direction D has not been tracked, then go on tracking, set direction fault counter C to 0, and add the temporary track result to the final track result. Otherwise track edge pixels located in neighbour direction of D, increase direction fault counter C, and put this track result in a temporary buffer. Retain the direction code.

II) If counter C is greater than a threshold, then abandon the track result in buffer.

III) If the final track result has too few pixels, then the edge is considered to be noise, which is not line-supported.

This method is similar to some chain code based line detection technologies [4-7], but it is more efficient and has more powerful capabilities in reducing noise..

3.2 Line segment extraction

Line segment extraction will be finished by line combination after Hough transformation. For each filtered line-support edge discovered by the abovementioned tracking process, a Hough Transformation (HT) is applied, which results a line segment. HT is a stable and robust line detection method [8], but its primary disadvantage is its large computational overhead. In the paper HT is modified to reduce overhead:

I) Calculate the centre of each line-supported track result. Set a new frame with the centre as its origin point and express the track result in this new frame. Thus, the possible radius range is reduced to locate in a small window around the centre.

II) Estimate the possible angle range according the retained direction code.

III) Apply HT, select the parameter with max vote, then select the pixels who vote for it. Then we determine the two end points of the extracted line according the max and min coordinates among the selected pixels.

IV) To express the line in the original frame.

This scheme avoids several disadvantages of HT, such as: rapid performance reduction with larger image size, difficulty determining the vote threshold when there are multiple lines to be extracted, and difficulty determining the end points of an extracted line.

The generality of HT, however, is lost in this process, so further processing is necessary. For each extracted straight line, search the neighbour area of its two end points.

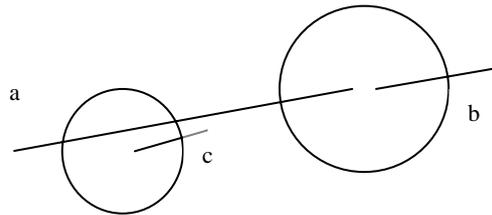


Figure 2 Line combination

If another line is found, and the two lines proceed in a similar direction, then combine the two lines. For each line in a line set, if another line is founded that can be combined with it in the same set; the line set can be combined. Through this process, short lines, such as line a and b in Figure 2, are combined into long lines, and many noise lines (such as line c in Figure 2) are removed.

3.3 Parallelogram extraction of building

To extract parallelograms the theory of perception organization and uncertainty reasoning are employed. Perception organization, first proposed by Lowe[9] in his SCERPO system, has been recognized by many researchers for its ability to derive global structure from local primitives. It is applied in numerous projects due to its low computational overhead and high anti-noise capabilities [10-12]. Based on the characteristics of parallelograms, the following algorithm, shown in Figure 3, is used to adapt perception organization to extract parallelogram. Uncertainty reasoning is also applied to solve the following problem. Although Bayesian reasoning and Dempster-Shafer's evidence theory are used by others [13-15], a simpler but effective method is used in the paper. The combination of multi evidence is described below:

1) $\text{Bel}(A1 \text{ and } A2 \text{ and } \dots \text{ and } An) = \min\{ \text{Bel}(A1), \text{Bel}(A2), \dots, \text{Bel}(An) \}$;

2) $Bel(A1 \text{ or } A2 \text{ or } \dots \text{ An}) = \max\{Bel(A1), Bel(A2), \dots, Bel(An)\}$;
 $Bel(X)$ denotes belief for X ;

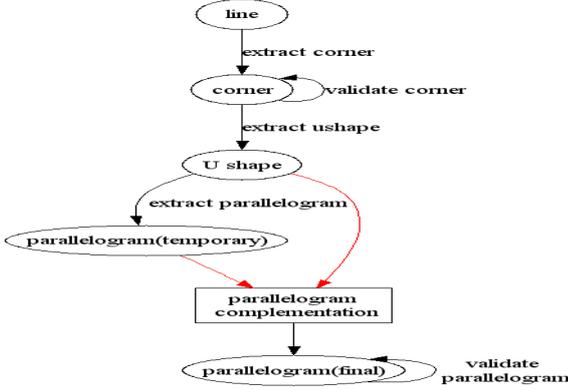


Figure 3 Parallelogram extraction

3.3.1 Shape representation: All shapes, including corners, U-shapes and parallelograms are represented by parts defined as follows.

Define: For a point set P in a 2D plane, the two end points of line segment a is $point1$ and $point2$, and the coordinates of an arbitrary point in line a is (x, y) . If $|point1.x - point2.x| < |point1.y - point2.y|$, then point set $\{(p, y) \mid p < x, (p, y) \in P\}$ is called the left side of line a , and two topple (a, pos) is called a *part*, where pos is a flag to denote left or right. In a similar way, we can define the case $|point1.x - point2.x| > |point1.y - point2.y|$.

Thus corners, U shapes and parallelograms can be expressed by 2 parts, 3 parts and 4 parts.

3.3.2 Corner extraction: For a possible corner $C1$ shown in Figure 4, the belief can be calculated as following.

$$Bel_{a1} = 1 - \text{dist}(P, P_{a1}) / \text{dist}(P, P_{a0}) = 1$$

$$Bel_{a2} = \begin{cases} \text{dist}(P, P_{a1}) > \text{lenTh}, & 1 \\ \text{else}, & 0 \end{cases}$$

$$Bel_{b1} = 1 - \text{dist}(P, P_{b1}) / \text{dist}(P, P_{b0})$$

$$Bel_{b2} = \begin{cases} \text{dist}(P, P_{b0}) > \text{lenTh}, & 1 \\ \text{else}, & 0 \end{cases}$$

Here $\text{dist}(p1, p2)$ denotes the distance between point $p1$ and $p2$, $Bel(X)$ is the belief of X and length is the minimum side length of an accepted corner. Then the belief of $C1$ can be expressed as:

$$Bel(C1) = \min\{Bel_{a0}, Bel_{a1}, Bel_{b0}, Bel_{b1}\}.$$

If a belief threshold is given, then any corner with belief greater than the threshold is accepted.

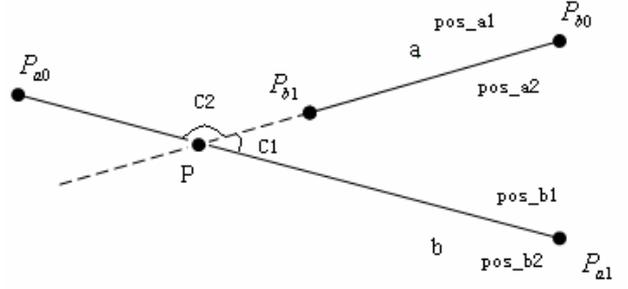


Figure 4 Corner extraction

3.3.3 U shape and parallelogram extraction: As Figure 5, a corner structure $C1$ consisting of $part1=(a, pos_a)$, $part2=(b, pos_b)$, if another corner structure $C2$ consisting of $part3=(c, pos_c)$ and $part2$ exists, then we can compute the belief of the parallel structure P consisting of $part3$ and $part1$ as follows:

If a is not located in pos_c side of c , or c is not located in pos_a side of a then $Bel(p)=0$;

Otherwise,

$$Bel(P) = \begin{cases} 1 - \text{angle}(a, c) / \text{Angle}, & \text{angle}(a, c) < \text{Angle} \\ 0, & \text{else} \end{cases}$$

, where $\text{angle}(a, c)$ denotes the inclination of a and c and Angle is the max inclination between two lines in an accepted parallelogram structure.

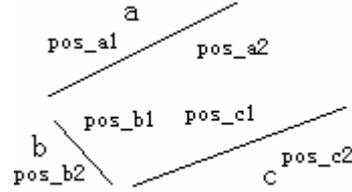


Figure 5 U shape and parallelogram extraction

3.3.4 Parallelogram complementation: For an extracted U-shaped U , if no parallelogram is extracted from it and $Bel(U)$ is greater than a given threshold, it could assume that in fact it is a contour of a parallelogram, but the fourth side has been lost because of line extraction. The parallelogram can be closed with a fourth side d such as Figure 6.

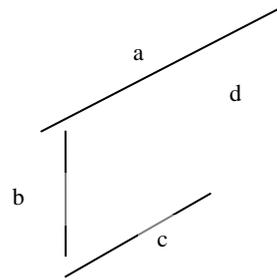


Figure 6 Parallelogram complementation

3.3.5 Solve conflict: For those extracted parallelograms, space conflicts could appear. The following rules are employed to solve this problem.

i) Block off rule: as shown in Figure 7(a), both a, b and a, c can form a corner structure, but because b blocks a and c , the corner formed by a and c should be deleted.

ii) Outer contour rule: as shown in Figure 7 (b), parallelogram $(P0, P1, P2, P5)$ is contained in parallelogram $(P0, P1, P3, P4)$, so the former is deleted.

- iii) Neighbour absorb rule: as shown in Figure 7 (c), side a is a line generated as a parallelogram complement, and it is near to b and has similar direction as b. Therefore, a should be absorbed by b; that is, we should delete a and enlarge b.
- iv) Higher belief rule: as shown in Figure 7 (d), the parallelogram with lower belief should be deleted.

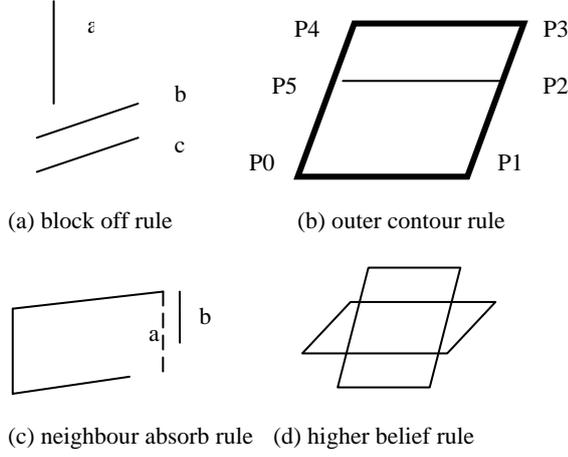


Figure 7 Conflict rules

3.4 Match between 2D image and 3D model

Matching between 2D images and 3D models is in essence the search for a transformation matrix that minimizes error, which is also a very important and difficult task to achieve texture mapping for building. In the paper, cursory match will be done based on camera parameters before fine matching. Cursory match can limit match in a small range to reduce calculation as well as error of matching.

The transformation matrix of perspective projection can be written explicitly as a function of its five intrinsic parameters (α, β, u_0 , and θ) and its six extrinsic parameters (the three angles defining rotation matrix R and the three coordinates defining translation vector t), namely,

$$M = \begin{pmatrix} \alpha r_1^T - \alpha \cot \theta r_2^T + u_0 r_3^T & \alpha t_x - \alpha \cot \theta t_y + u_0 t_z \\ \frac{\beta}{\sin \theta} r_2^T + u_0 r_3^T & \frac{\beta}{\sin \theta} t_y + u_0 t_z \\ r_3^T & t_z \end{pmatrix}$$

where r_1^T, r_2^T , and r_3^T denote the three rows of the matrix R and t_x, t_y , and t_z are the coordinates of the vector t [2]. There are 11 parameters to solve for. If information about these parameters is not available, the parameter space is very large and computational overhead would be unacceptably large. Various researchers have proposed algorithms to solve the match problem with several unknown parameters, but no solution exists to solve problems with many unknown parameters^[17-19]. So the algorithm will simplify the camera model according to the image and CAD data given. E.g., $\alpha = \beta$, $\theta = 0$, $R = I$ (I is a unit matrix), setting u_0, v_0 to the centre of the aerial image and so on. Thus parameters unknown are fewer, and the difficulty will come down, so to make the algorithm practical. Based on camera model parameters, for an arbitrary part of the aerial image, only u_0 and v_0 are unknown. It can move the aerial image along the

model-projecting image, for each position (u_0, v_0), calculate the Hausdorff distance, where the position which has the minimum Hausdorff distance is the optimal matching result. Hausdorff distance can be defined between point sets, line segment sets and parallelogram sets with lower computational overhead, yet the reliability of the process also drops. The following flowchart, shown in Figure 8, is employed to finish match between feature image and 3D model.

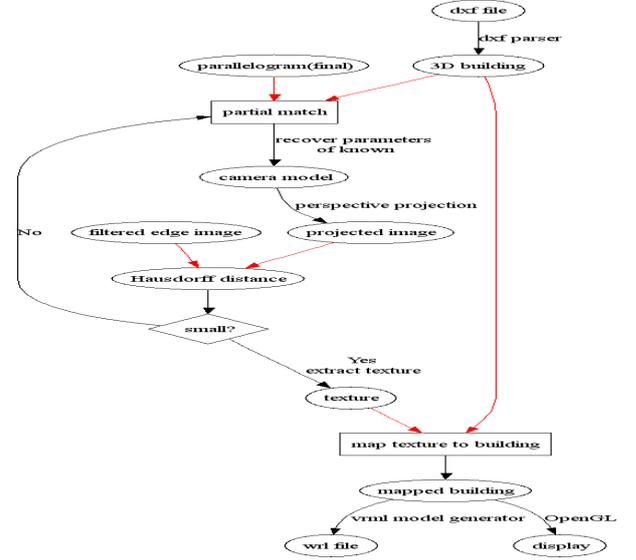


Figure 8 Match flowchart

3.5 Texture Mapping

Since the camera positions of the original photographs are recovered during the modelling phase, projecting the images onto the model is straight forward. The process of texture-mapping a single image onto the model can be thought of as replacing each camera with a slide projector that projects the original image onto the model. When the model is not convex, it is possible that some parts of the model will shadow others with respect to the camera. While such shadowed regions could be determined using an object-space visible surface algorithm, or an image-space ray casting algorithm, which is efficiently implemented using z-buffer hardware.

4. RESULT

The above algorithms are employed to finish 3D texture mapping for buildings in Shinjuku of Tokyo, Japan. The result is shown in Figure 9. From Figure 9(b), there are so many false edges after edge filtering and Hough transformation. But after line combination, most of residual edges are outlines of buildings. From Figure 9(d), some parallelograms of buildings' un-shadowed sides are extracted, while some of them are not resumed. Based some of parallelograms, partial match can be done.

After matching between 2D image and 3D model, 3D textured buildings can be achieved, which is shown in Figure 10. The experiment results suggest the algorithms of 3D textured building modelling based on matching between 2D feature image and 3D model is reliable and efficient. But there are still some errors in the result of texture mapping such as incorrect of texture mapping position. By analysis of result, errors are found

to be derived from the following factors: 1) error from CAD model given; 2) errors caused by ADS40 sensor in flying. 3) errors from calculation.

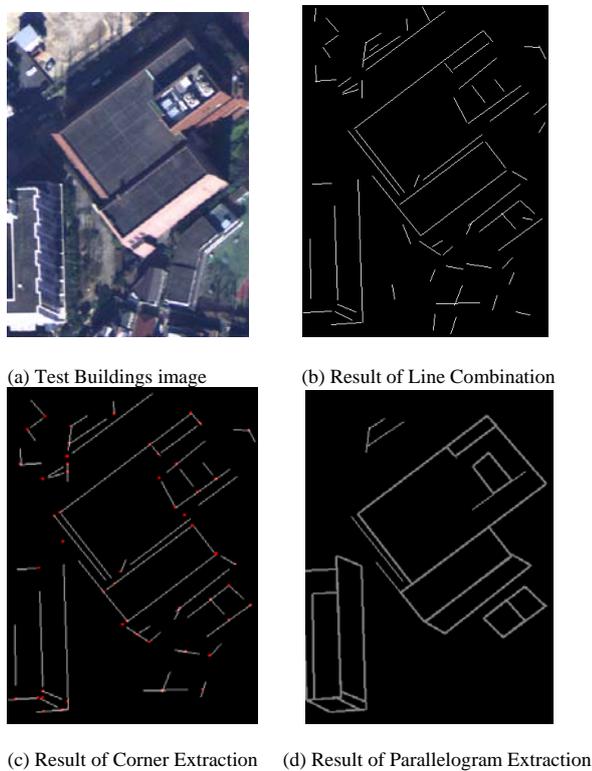


Figure 9 Building Features Extraction

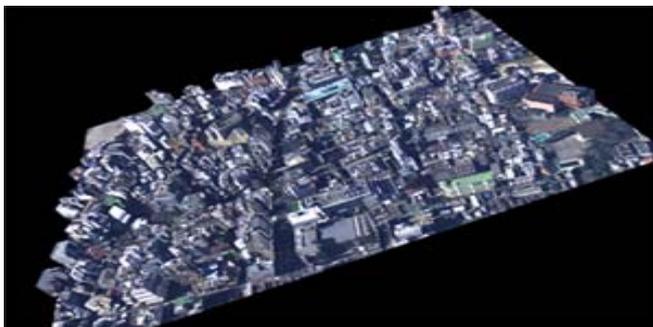


Figure 10 Result of 3D Textured Building

5. CONCLUSION

The algorithm of parallelogram extraction based on perceptual organization theory fits to resolve the problem of feature extracting for buildings using high spatial resolution imagery with more details and noises. Relation between coordinates of image and coordinates of 3D model can be built after matching between 2D image and 3D model. The experience suggests that automation 3D textured building modelling proposed in the paper is practical. Especially given camera model, matching result will be more exact. There are also many studies to continue, such as how to improve the algorithm of feature

extraction and matching to be more robust to noise by using artificial intelligence.

REFERENCE

A Gruen, Zhang L, Wang X. 3D City Modelling with TLS (Three-Line Scanner) Data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV-5/W10

F Dieter. 3D Building Visualisation – Outdoor and Indoor Applications, in Fritsch (Ed.). Photogrammetric Week '03. Wichmann Verlag, Heidelberg, pp. 281-290.

J F Canny. A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, 8 (6) : 679-698.

Zhu Xiao-yue. Extract Straight Line From Image Based on Computer Vision Task. Journal of Nanchang Institute of Aeronautical Technology (Natural Science).Sep. 2003.Vol.17,No.3:63-66

Bao su-su.Yang Lu. Algorithm for Detecting Vertical and Horizontal Lines in Real-time Image processing. Journal of Hefei University of Technology. Aug 2003. Vol.26, No.4:550-552.

Sun Ai-rong, Tan Yi-hua. Journal of Wuhan University of Technology (Transportation Science & Engineering).Dec.2003. Vol 27, No.6:807-809.

Shi Ce, Xu Sheng-rong, Jing Ren-jie etc. The Replace Algorithm of Hough Transform in Real-time Image Processing. Journal of Zhejiang University (Engineering Science). Sep,1999. Vol 33, No.5:482-486

Luc Baron, P. Eng. Genetic Algorithm for Line Extraction. Rapport technique EPM/RT-98/06, École Polytechnique de Montréal, 20 pages, août 1998.

David G. Lowe, Perceptual Organization and Visual Recognition, Kluwer Academic Publishers, Norwell, MA, 1985

Chung-An Lin. Perception of 3-D Objects From an Intensity Image Using Simple Geometric Models. PhD thesis, Faculty of the Graduate School, University of Southern California.1996:28.

L.R.Williams,A.R.Hanson.Perceptual completion of occluded surfaces.Computer Vision and Image Understanding,64(1):1-20,July 1996.

Christopher O.Jaynes, Frank Stolle, Robert T.Collins.Task Driven Perceptual Organization for Extraction of Rooftop Polygons. Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on 1994.

P.Vasseur,C.Pegard. Perceptual Organization Approach Based on Dempster-Shafer Theory. Pattern Recognition,Volume 32, Issue 8 , August 1999:1449-1462

Mei Xue-liang. Automatic 3D Modeling of Regular Houses from Aerial Imagery. Dr. thesis. Wuhan Technical University of Surveying and Mapping.1997.

Xi Xue-qiang. Research on Model Based 3-D Object recognition in remote sensing image. Dr. thesis. National University of Defense Technology,2000.

David A. Forsyth, Jean Ponce. Computer vision: A Modern Approach. Tsinghua University Press, Beijing, 2004:28-31.

David.Jacobs, Ronen Basri. 3-D to 2-D Recognition with Regions. 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)

Ronen Basri, Daphna Weinshall. Distance Metric between 3D Models and 2D Images for Recognition and Classification. IEEE Trans. On Pattern Analysis and Machine Intelligence, 18(4):465-470,1996.

Steven Gold, Anand Rangarajan, Chien-Ping Lu, etc. New Algorithms for 2D and 3D Matching: Pose Estimation and Correspondence. Pattern Recognition 31(8): 1019-1031 (1998)