Rapid Texture Mapping from Image Sequences for Building Geometry Model

ZHANG Zuxun  WU Jun  ZHANG Jianqing

1 Introduction

Texture Mapping is an important step for reconstructing three-dimensional (3D) city landscape, for it simplifies the modeling process by only using some image patch to represent some geometry detail as well as producing a photo-realistic scene. At present, because of the powerful capacity in measuring and capturing a larger range scene, mapping both the geometric and physical information for building from aerial images is considered as one of the economical and practical solutions in city modeling. When actually using aerial images to provide geometry and texture information, two issues have to be considered: (1) the vertical face for building is usually invisible in such a horizontal aerial image; (2) until now, even semi-automatic approach for building geometry modeling from aerial images is still far away from real practice. So, it becomes clear that any rapid and economical solution for building texture mapping will play important role in reconstructing 3D city landscape.

In this paper, a deep research on rapidly texture mapping from three image sequences for building geometry model is introduced. Here, three image sequences are captured by a digital video camera on a helicopter. The initial building geometry model is produced by integrating the 2D ground vector map with LIDAR (light detecting and ranging). On the basis of the strict digital photogrammetric theory and redundant image information, By this approach we can semi-automatically obtain the correct image patch (texture) for each face of building by setting up the correct correspondence between the
space edge of building and its line feature in image sequences. The proposed approach is tested in producing 3D data for car navigation and good results are acquired.

2 Principle for building texture mapping

Generally speaking, the texture mapping for each space planar face (we suppose each face of the building geometry model is planar) requires finding exact image pixel position for each corner vertex in the planar face. With the assumption of perfect projection, e.g., with a pin-hole camera, it is well known that we can directly get the correct image pixel position for each space corner vertex, once image pose parameters are determined and sequentially the resampled image patch as texture can be automatic output. Usually, there are always ample line features in the man-made scene. Instead of directly searching image point features, the line feature should be used to determine the corner vertex of the building geometry model, by tracing the adjacent two image line features which exactly meet at that corner, since the image line feature has more stable geometry characters (length, direction and grayscale) and would be very helpful in overcoming such difficult situations as image discontinuity, shadow, etc. The above strategy is mainly used to match each corner vertex of building’s roof to its exact image position in vertical images as well as improving the whole building geometry model. On the basis of the improved building geometry model, we can easily set up the correspondence between the space edge of building’s wall and its image line feature in oblique image sequences only by adjustment the top/bottom height in the building geometry model. As a result, the texture mapping for the vertical face of building is also solved successfully.

2.1 Implementing “from coarse to fine” for line extraction

The line extraction is only applied to the neighborhood of the projected result of a roof. By considering different projected results caused by variable roof in the process of iteration, “from coarse to fine” is successfully implemented in extracting all possible line features on image for each spatial line of the roof.

2.1.1 The “coarse” line extraction

At the beginning, one rectangle image patch with width $d$, centering at the projected position, is used as the target region for searching possible image edges. At this stage, $d$ ($d = 30-50$ pixels) is required to be wide enough to contain the corresponding image line feature, and gaining the long line feature is the main purpose. Common
steps of line extraction are edge detecting, edge tracing and edge fitting on the basis of image pyramid. Although the image line features extracted in this stage are not so fine in geometry precision, they are still good enough for line intersection and space forward intersection to improve the roof geometry model.

2.1.2 The “fine” line extraction

Since the roof geometry model is improved a lot with the adjustment of space forward intersection, and its projection in vertical image sequences is becoming similar to its real image line feature in such geometry characters as length, direction and distance, we can apply the “fine” line extraction algorithm based on least squares template matching (LSTM) to a narrow rectangle image patch ( d = 5-10 pixels) to gain the final ILFC. The steps are of this operation:

1. Determine the theory line template (TLT). Usually, the ideal step edge in image space is in the form of some curve shape like S and its grayscale distribution can be supposed as Fig. 1 shows. Then, for any projected line of roof space edge , we can obtain its TLT as Fig. 2 shows, according to its length.

2. Rotate the rectangle image patch which centers at to be horizontal and calculate the cross-coefficients by vertically moving the TLT along the horizontal image patch. As a result, we can obtain a cross-coefficients distributing curve as Fig. 3 shows, taking the cross-coefficient values as the vertical axis and the displacement away from as the horizontal axis.

3. Find the position for any wave-apex or wave-valle in the cross-coefficient distributing curve and inversely rotate them to obtain the ILFC. Certainly, at this time, the image lines in ILFC is still not accurate in geometric position, and we can correct them by LSTM.

2.2 Image line evaluation

With the topological relation provided by the initial building geometry model, we can easily determine the image corner point by intersecting two adjacent image line features, whose correspondent space edge passes through that space corner vertex. However, since each space edge in roof owns an ILFC with different numbers of image line features in each image, it is impossible to get the intersected image corner point simply by enumerating all the combination of possible image line features. Here, we score every image line feature in ILFC under some reasonable criterion and select the image line feature with the highest score as the best candidate for further intersection.

Let the straight edge denote the projection of the space line in one image, denote ILFC of . denote the length of , denote the angle between and , denote the distance from geometry center point of the roof projection to (Fig. 4). Then the following function
\[ F(t_i) = p_1 \frac{\text{LEN}_{t_i}}{\max(\text{LEN}_{t_i})} + p_2(1 - \frac{\text{DIR}_{t_i}}{\max(\text{DIR}_{t_i})}) + p_3(1 - \frac{\text{DIST}_{t_i}}{\max(\text{DIST}_{t_i})}) \] (1)

Fig. 4 Estimation for each line in ILFC

where \( \max(*) \) is the maximum value for * among all the possible image edges; \( P_j (j = 1, 2, 3) \) are the constants reflecting the different effect caused by different geometry character on the selected function. At the “coarse” line extraction stage, for the projected result is far away from its image line feature and the length is mainly as criterion, we have: \( p_1 = 0.8, p_2 = p_3 = 0.1 \). At the “fine” line extraction stage, for the projected result is approximate to its image line feature and all length, direction and distance are used as criterion, we have: \( p_1 = 0.4, p_2 = p_3 = 0.3 \).

2.3 Space forward intersection by indirect least squares adjustment with constraint

An indirect least squares adjustment with the flat roof constraint is used in space forward intersection to restore the roof vertex, since we can get several homo-points in different images for each roof vertex by intersecting the two adjacent candidate line features. The equation and the conditional equation for estimating the roof vertex are, according to the collinear equation\(^{[6,7]} \):

\[
\begin{align*}
x_i - u_0 &= -f \left[ a_1 (X_i - X_s) + b_1 (Y_i - Y_s) + c_1 (Z_i - Z_s) \right] = -f \frac{u}{w} \\
y_i - v_0 &= -f \left[ a_2 (X_i - X_s) + b_2 (Y_i - Y_s) + c_2 (Z_i - Z_s) \right] = -f \frac{v}{w} \\
Z_i &= Z_0
\end{align*}
\]

And the conditional equations are:

\[
H_{11} \Delta Z_i + H_{12} Z_i - L_{1i} = 0,
\]
where \( H_{11} = 1, H_{12} = -1 \)

2.4 Homo-line matching based on three-view geometry constraint

As described above, the most projections of roof can be matched to its corresponding image edge. However, for some difficult situations such as dense buildings and building with double edges, it is possible to produce some misidentification, and some necessary human interaction is required to correct it. In order to reduce the human interaction and achieve as high effect as possible, a homo-line matching is applied for automatically getting the corresponding image line features in all target images.

Supposing the space line \( L \) appears in three neighbor images, \( L_m \) is the corresponding image...
age line feature in the middle image. \( L_i = \{ l_i, j \in [0, m] \} \) is the ILFC in left image. \( L_i = \{ l_i, j \in [0, n] \} \) is the ILFC in the right image. As we know, the interpretation plane \( S' \) (formed by \( l_i \) and the left camera center) and the interpretation plane \( S'' \) (formed by \( l_n \) and the right camera center) will intersect at some space edge \( L' \) provided that the two interpretation planes are not parallel. Let line \( L_{l_i} \) be the projection of \( L_i \) in the middle image, and \( e_i \) be the error between \( L_i \) and \( L_{l_i} \). Then we can set up the error matrix \( E_L = \{ e_i \} \), \( i \in \{ 0, n \} \). Obviously, only if both \( l_i \) and \( l_n \) are the homo-line of \( L_i \), then \( L_i \) and \( L_{l_i} \) are the same space line and we can get the minimum \( e_i \). Thus we can obtain the following items.

2.4.1 Definition for line error (Fig. 5)
Under the 2D image coordinate system (taking the left-down corner of middle image as the origin point and its horizontal direction as the axis \( X \) ) and let \( l \) be the length of \( L_i \). \( L_i, L_n \) be the predicted line, \( X_1, X_2 \) be two end points of \( L_i \), and \( S_1, S_2 \) be the corresponding perpendicular points in \( L_{l_i} \), then we can define the line error \( e_i \) as:

\[
\text{Area}(X_i, S_1, S_2, X_2)
\]

![Fig. 5 Error definition for predicted line](image)

Furthermore, with the point-to-line formula, we have:

\[
h_1 = \frac{m_x x_2 + m_y y_2 + m_z}{\sqrt{m_x^2 + m_y^2}};
\]

\[
h_2 = \frac{m_x x_3 + m_y y_3 + m_z}{\sqrt{m_x^2 + m_y^2}};
\]

\[
h(s) = h_1 + \frac{h_2 - h_1}{l} \cdot s
\]

where \( h_1, h_2, h(s) \) are the distances from points \( X_1, X_2, P(s) \) in \( L_i \) to \( L_{l_i} \). Thus we can obtain the following items.

2.4.2 Line prediction
Let \( P', P \) and \( P'' \) be the projection matrixes for the left image, middle image and right image, respectively. By applying the same perspective transformation to \( P', P \) and \( P'' \), we can obtain the result for some one of three matrixes as \([ I | 0 \] \). Then, let the transformed projection matrixes be in the following forms:

\[
P' = [ A | a ], P = [ I | 0 ], P'' = [ B | b ]
\]

where \( A \) and \( B \) are \( 3 \times 3 \) rotate matrices, \( a \) and \( b \) are \( 3 \times 1 \) translation vectors. Let \( a_i \) and \( b_i \) be the \( i \)th (\( i = 2, 3, 4 \)) column for the projection matrixes \( P' \) and \( P'' \), respectively. Let \( L', L \) and \( L'' \) be the corresponding image line features for space line \( L \) in left, middle and right image, respectively, \( S', S \) and \( S'' \) be the interpretation plane for \( L', L \) and \( L'' \) respectively, then we have:

\[
S' = P'^T L' \quad S = P^T L \quad S'' = P''^T L''
\]

Since the three interpretation planes intersect on one space line \( L \), any space point \( O(x, y, z) \) in \( L \) will be certainly in any one of the three interpretation planes and satisfy:

\[
S' \circ O = S' \circ O = S'' \circ O = 0.
\]

Let the matrix \( M = \begin{bmatrix} S' & S & S'' \end{bmatrix} \), then we have:

\[
M = \begin{bmatrix} P'^T L' & P^T L & P''^T L'' \end{bmatrix} = \begin{bmatrix} a_i^T L' & 1 & b_i^T L'' \end{bmatrix},
\]

\[
M^T O = 0.
\]

Obviously, the rank of \( M \) is 2. \( S, S', S'' \) are not independent with each other. Let \( S = K_1 S' + K_2 S'' \), and by using the zero element in \( M \) we can solve for the \( K_1 \) and \( K_2 \) as

\[
K_1 = k(b_i^T L''), \quad K_2 = -k(a_i^T L')
\]

\[
(k \text{ is a constant and } k \neq 0).
\]

Furthermore we can have:

\[
l_i = P''^T (a_i b_i^T) L' - P'\tilde{X} (a_i b_i^T) L''
\]

\[
= P''^T (a_i b_i^T) L' - P'\tilde{X} (a_i b_i^T) L''
\]

where \( i = 1, 2, 3 \), and \( l_i \) is the \( i \)th part of \( l \). Actually, if let the matrix \( T = [ T_1, T_2, T_3 ] \), \( T_1 = a_i b_i^T - a_i b_i^T \), then we will get the 3D tensor \( T \),
which clearly reflects the relation among homo-lines in three images and is determined only by their interior orientation parameters and relative movement\(^4\).

Now, by Eq. (3) we can easily use two homo-lines located in the left and right images to predict corresponding image lines in the middle image in the form of line equation. Moreover, because any unparallel two interpretation planes always determine one space edge, we can certainly get the predicted image line feature in the middle image from any two image line features located in the left image and the right image.

From the above analysis we can infer that, on the basis of the three-view geometry constraint, all the corresponding image line features for any space edge can be automatically determined, once its corresponding image line feature in the middle image is determined and ILFC is set up for other images. Usually, we are able to correct misidentification in one image manually. Because the relative position between the projection of space edge and its corresponding image line feature is almost similar in different vertical images, it is easy to get the initial position for each homo-line in other images and each ILFC by further line extraction.

3 Implementing rapid texture mapping

3.1 Setting up initial building geometry model

The initial building geometry model is determined by digital map and LIDAR data.

3.2 Registering target images

The target images, in which the building should be fully included, are automatically registered by comparing the building location with every image range on ground, and this comparison is based on image pose parameters.

3.3 Matching space edge to its corresponding image line feature

- Matching of space edge for the building roof
- Matching of space edge for the vertical building face

After the correspondence between the roof space edge and its image line feature is determined, the building geometry model is also improved. When the building geometry model is projected onto oblique images, we can easily set up the correspondence between the space edge in the vertical building face and its image line feature simply by adjusting the height of the building bottom or top.

3.4 Mapping texture data

- The texture for the roof is mapped from the vertical images. Usually, the texture is taken from the middle image among all the target images.
- The texture for the wall is mapped from the oblique images. When a wall appears in multiple images, the image with maximum projected area will provide the corresponding texture.

4 Experimental results

A software, 3Dmapcreate, has been developed to test the above approach and rapidly map the texture for the remarkable building along several streets for the purpose of 3D car navigation application. The resource data include:

1. three image sequences captured by a digital video camera on a helicopter which flies along the road at the height of about 400 meters. Two strips (each 89 images) are the oblique photographs for the texture of walls, and one (349 images) is the vertical photographs for the texture of roof. The images with the size 1080X1920 overlap each other of 98%.
2. the digital map describing the building shape and its 2D location.
3. LIDAR data providing height information.

Fig. 6 shows the digital map and LIDAR data together (the grid points are for LIDAR data). Fig. 7 and Fig. 8 show the three image sequences (only one image for each strip). Fig. 9-Fig. 14 demon-
strate the whole texture mapping process for the cross-marked building in Fig. 3. Among those figures, Fig. 9 is the projected result with initial building geometry model in two vertical images and Fig. 10 and Fig. 11 are the line extracting result (including "coarse" and "fine") and Fig. 12 and Fig. 13 are the projected result with improved building geometry model in three strips (two for vertical strip and two for oblique strips). Fig. 14 is final building model with the corresponding texture rendering after some height adjustment.

![Fig. 6 Urban digital map and LIDAR data](image6)

![Fig. 7 Vertical image sequences](image7)

![Fig. 8 Oblique image sequences](image8)

![Fig. 9 Projection with initial building geometry model in vertical image sequences](image9)

![Fig. 10 "Coarse" line extraction](image10)

![Fig. 11 Line extraction result](image11)

![Fig. 12 Line extraction result](image12)

![Fig. 13 Improved building model](image13)

![Fig. 14 Final building model](image14)
5 Conclusions

As we know, the texture mapping is actually the reverse process from 2D image space to 3D Euclidean space. Usually, intensive human interaction can greatly reduce the complexity of this reversing process as well as lead to expensive cost. For example, a city with about 6,000 buildings usually requires ten persons to work three months to complete the texture mapping project and their average speed for one person is less than 10 buildings per workday. In our experiment, however, it can be found that the practical work time for one person to handle one building is less than two minutes, and so, even we consider some additional human operation such as photo capturing, data format converting etc., it is possible for one person to deal with 100 buildings in one day. The semi-automatic approach proposed above is based on the correspondence between the space edge of building geometry model and its image line feature by the strict digital photogrammetric theory and the redundant image information.

References

1 Gruen A, Nevatia R (1998), Automatic building extraction from aerial images. Computer Vision and Image Understanding, 73(2): 1-2
2 Zhang Z X, Zhang J Q (1997) Digital photogrammetry. Wuhan: Press of Wuhan Technical University of Surveying and Mapping. (in Chinese)
3 Förstner W (1982) On the geometric precision of digital correlation. IAPRS Comm. III, 24, 176-189

(Continued on Page 31)
tics of city is another research subject.

References

1 Haala N, Brenner C (1999) Virtual city models from laser altimeter and 2D map data. *Photogrammetric Engineering & Remote Sensing*, 65(7):787-796
2 Ackerman F (1999) Airborne laser scanning—present status and future expectations. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3): 64-67
3 Weidner U, Forstner W (1995) Towards automation building extraction from high-resolution digital elevation models. *ISPRS Journal of Photogrammetry & Remote Sensing*, 50(4):38-49
4 Baillard C, Maitre H (1999) 3D reconstruction of urban scenes from aerial stereo imagery: a focus strategy. *Computer Vision and Image Understanding*, 76(3), 244-258
5 Baillard C, Dissard O, Jamet O, et al. (1998) Extraction and textural characterization of above ground areas from aerial stereo pairs: a quality assessment. *ISPRS Journal of Photogrammetry & Remote Sensing*, 53(3):130-141
6 Maas H G (1999) The potential of height texture measurement for the segmentation of airborne laser scanner data. The 4th International Airborne Remote Sensing Conference and Exhibition, Ottawa,
7 Hiroshi M, Hiroyuki H, Izumi K (1999) Extraction of building shapes from laser scanner data using region segmentation method. International Workshop on Vision-based Techniques in Visualization and Animation, Onuma, Japan.
8 Haala N, Brenner C (1999) Extraction of buildings and trees in urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):130-137
9 Baistsavias E P (1999) Airborne laser scanning: existing systems and firms and other resources. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3): 164-198
10 Petzold B, Reiss P, Stossel W (1999) Laser scanning—surveying and mapping agencies are using a new technology for the derivation of digital terrain models. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3): 95-104

(Continue from Page 15)
4 Gruen A (1985) Adaptive least square coorelation; a powerful image matching technique. *Journal of Photogrammetry and Remote Sensing*, 14(3):175-187
5 Zhang Y J (2002) Three dimentional reconstruction and visual inspection of industrial sheetmetal parts with image sequences: [Ph. D dissertation]. Wuhan, Wuhan University, (in Chinese)
6 Wang Z Z (1984) Principles of photogrammetry. Beijing, Publishing House of Surveying and Mapping, (in Chinese)
7 Jin W X, Yang X H (1994) Photogrammetry. Beijing, Publishing House of Surveying and Mapping, (in Chinese)
8 YU Z T, Lu L C (1982) Principles of survey measurement. Beijing, Publishing House of Surveying and Mapping, (in Chinese)