Realistic 3D Reconstruction from Two Uncalibrated Views

Shungang Hua and Ting Liu

Key Laboratory for Precision and Non-traditional Machining Technology of Ministry of Education, Dalian University of Technology, Dalian, 116024, P.R.China

Summary
An approach for 3D surface reconstruction from two uncalibrated views is proposed in this paper. The SUSAN corner detector is used for corners from points in edges, which are detected by Sobel edge operator. After corners matching between two images, the fundamental matrix is estimated to acquire 3D structure from matched corners by an improved weighted linear algorithm, which is based on the epipolar geometry and the absolute conic theory. The triangular textures are acquired by dividing 2D image. Then a realistic 3D surface model is built by mapping the triangular textures to the 3D structure surface. Experiments show that the reconstructed 3D surface models are satisfactory and photo-realistic.

Key words:
Image-based modeling, camera calibration, triangulation, fundamental matrix, corner detection

1. Introduction

In recent years, image-based rendering and modeling have been one of the most active subjects in virtual reality and computer graphics domain. With the technique of image-based modeling (IBM), we can acquire 3D information and models of scene or objects from images taken from digital equipment, such as hand-held digital camera and digital video. As compared with conventional geometry-based modeling, the image-based modeling technique can be used to extract actual texture and illumination from images directly for visual 3D modeling, without many complex processes, such as geometry modeling, shade and ray-tracing computing.

The technique of image-based modeling is related to virtual reality, pattern recognition, computer graphics, computer vision, photogrammetry and so on. Various algorithms were proposed to perform 3D reconstruction of architecture, natural scene or face etc [1-15], which is based on calibrated or uncalibrated views. An early approach was proposed by Tomasi et al [1]. They extracted 3D model from perspective image sequences with the method of affine transform factorization. Pollefeys [2] presented a method which reconstruct object surface model more accurately with a hand-held camera. He mapped image textures onto the surface of 3D model to achieve perfect visual effect after matching every point between images. For architecture scene images, Debevec et al[3] gave a 3D modeling method to reconstruct simple geometry objects, which only needed several images and some known geometric parameters. In order to achieve photorealistic 3D reconstruction from handheld cameras, Rodriguez et al [4] presented an integrated approach to generate 3D mesh from sparse data and deal with partial occlusions.

3D surface model can be reconstructed from one, two or several uncalibrated images. For one uncalibrated image, the 3D information is acquired through geometric attribute, such as coplanarity, orthogonality and parallelism. This method only needs one image, but there is too much restriction in content of image to widely use. For two images, 3D reconstruction is based on matched points. This method has no restriction in content of image, and could be the basis of 3D reconstruction from several images. These images are captured in the same scene with different viewpoints.

This paper proposes an approach to achieve 3D surface reconstruction from two uncalibrated images. The steps of our approach are: (1) detection and matching of feature points based on two uncalibrated images; (2) fundamental matrix computation based on matched points, camera self-calibration for intrinsic parameters and exterior parameters; (3) 3D coordinates computation of matched points for surface reconstruction; (4) mapping textures from images onto 3D surface for visual geometric model. We adopt approaches synthetically, such as the smallest univalue segment assimilating nucleus (SUSAN) corner detector[16], the weighted linear algorithm[17], the Delaunay Triangulation algorithm[18]etc, and also improve them for more accurate and faster 3D modeling.

2. Detection and matching of feature points

The accuracy of matched feature points would affect estimation of fundamental matrix and computation of 3D points intensively. This paper detects and matches corners of images as feature points. Corners are points, which have biggest curvature on partial outline, embody the significant information. The smallest univalve segment assimilating nucleus (SUSAN) algorithm [16] is an efficient approach

Manuscript received June 5, 2007
Manuscript revised June 20, 2007

to low level image processing, and can be used for edge and corner detection. But it needs detecting each point of image. In this paper, we detect edges of image by Sobel operator firstly, and then extract corners from the detected edges by SUSAN.

The feature points (corners) matching between two images could be achieved automatically or interactively. Automatic matching strategy could be adopted between two images with close viewpoints. The matching point is determined by similarity measures, which is defined as the weighted sum of the gradient magnitude, gradient orientation and intensity cross-correlation. For two corners in two images, the more similarity measures, the more possibility to be matched. Matched corners similarity measures should greater than a threshold. For two images from different viewpoints obviously, we can build the corners correspondence interactively. The example is shown in Fig.1, the symbol "+" indicates the detected corners, "□" is matched corners between two images.

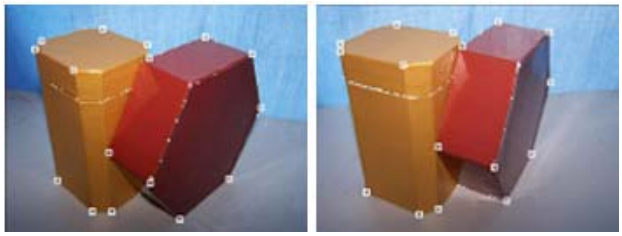


Fig. 1 Corners initial matching.

3. Fundament matrix estimation and refinement

Based on matched corners, fundamental matrix can be estimated and refined. Fundamental matrix is a matrix with rank-2, which implements corresponding geometry relationship. It does not only include intrinsic parameters and external parameters, but also can restrict the search range of matching points to a pair of corresponding epipolar lines. The weighted normalization linear algorithm [17] is steady for estimation of fundamental matrix. These matched points were weighted and normalized before fundamental matrix estimation to acquire robustness and improve accuracy. On the basis of the weighted normalization linear algorithm, our algorithm removes the outliers iteratively to refine fundamental matrix by following equation:

$$d_i = \left(\frac{1}{\sqrt{(\mathbf{F}\mathbf{m}_i)_1^2 + (\mathbf{F}\mathbf{m}_i)_2^2}} + \frac{1}{\sqrt{(\mathbf{F}^T\mathbf{m}'_i)_1^2 + (\mathbf{F}^T\mathbf{m}'_i)_2^2}} \right)$$

$$\left| \mathbf{m}'^T_i \mathbf{F} \mathbf{m}_i \right| = d(\mathbf{m}'_i, \mathbf{F}\mathbf{m}_i) + d(\mathbf{m}_i, \mathbf{F}^T\mathbf{m}'_i) \quad (1)$$

where F represents the fundamental matrix, $\mathbf{m}_i$ and $\mathbf{m}'_i$ denotes a pair of matched points of two images, $(\mathbf{F}\mathbf{m}_i)_j^2$ is the j component $(j = 1,2)$ of epipolar vectors $\mathbf{F}\mathbf{m}_i$. The $d(\mathbf{m}_i, \mathbf{F}^T\mathbf{m}'_i)$ is the distance from $\mathbf{m}_i$ to $\mathbf{F}^T\mathbf{m}'_i$. The $d(\mathbf{m}'_i, \mathbf{F}\mathbf{m}_i)$ is the distance from $\mathbf{m}'_i$ to $\mathbf{F}\mathbf{m}_i$. The d_i is the sum of the two epipolar distances.

If the error of matched points is greater than a threshold by Eq.(1), we consider the matched points as outliers and remove them. The epipolar distance obeys Gauss distribution, so the threshold is set to 3σ (σ is the variance). After the outliers removing, we refine the fundamental matrix with residual matched points iteratively until the disappearance of outliers. The residual matched points and epipolar lines after refining fundamental matrix are shown in Fig.2.

Fig. 2 Corresponding corners and epipolars.

4. Camera self-calibration

Camera self-calibration is key technology to acquire 3D information from images. It includes the computation of intrinsic parameters and exterior parameters. The intrinsic parameters are focus length, principal point, etc. The exterior parameters are the information of camera movement, such as rotation and translation.

4.1 Computation of intrinsic parameters

The intrinsic parameters can be expressed as the calibration matrix:

$$K = \begin{bmatrix} -fk_u & fk_u \cot\theta & u \\ 0 & -fk_v / \sin\theta & v \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where f is the focus length, k_u and k_v are proportion factors of pixel on the direction of u and v respectively, they implicate the scale of pixel, θ is the included angle of

x axis and y axis in image coordinate system; (u,v) is the principal point. Peter Sturm [19] presented a method to compute the focus length with fundamental matrix. This method is based on the simplified camera model. In general, the plane of image is rectangle ($\theta = \pi/2$), the principal point is the centre of image plane, $u = w/2, v = h/2$ (w and h are the width and height in image respectively); pixel is square ($k_u = k_v = -1$). After simplification, the calibration matrix likes this:

$$K = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

There is only one unknown parameter f (focus length). The semi-calibrated fundamental matrix G is used to compute focus length. G is an intermediate between the fundamental matrix and the essential matrix, can be expressed by:

$$G \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w/2 & h/2 & 1 \end{bmatrix} F \begin{bmatrix} 1 & 0 & w/2 \\ 0 & 1 & h/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

the symbol "~" denotes equality of matrices or vectors up to scale (the same with the following text). We can get G' by normalizing G to improve the stability of computation:

$$G' = \frac{G}{\sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3} g_{ij}^2}} \tag{5}$$

$g_{ij}(i,j = 1,2,3)$ is element of G. Let the singular value decomposition of G be given by:

$$G' = U\Lambda V^T \tag{6}$$

with $\Lambda = diag(a,b,0)$ the diagonal matrix, U and V are both orthogonal matrices: $U = [u_1 \quad u_2 \quad u_3]$, $V = [v_1 \quad v_2 \quad v_3]$, $u_i = [u_{i1} \quad u_{i2} \quad u_{i3}]^T$, $v_i = [v_{i1} \quad v_{i2} \quad v_{i3}]^T$, $i = 1,2,3$. Based on literature [20], the Kruppa equations can be written as:

$$\begin{aligned} & U\Lambda V^T diag(f^2, f^2, 1) V\Lambda U^T \\ & \sim [u_3]_\times diag(f^2, f^2, 1)[u_3]_\times \end{aligned} \tag{7}$$

$[u_3]_\times$ is the skew symmetric matrix of vector u_3:

$$[u_3]_\times = \begin{bmatrix} 0 & -u_{33} & u_{32} \\ u_{33} & 0 & -u_{31} \\ -u_{32} & u_{31} & 0 \end{bmatrix} \tag{8}$$

Multiplying the Kruppa equation (7) by U^T from the left and U from the right, due to the orthogonality of U:

$$\begin{aligned} & \Lambda V^T diag(f^2, f^2, 1) V\Lambda \\ & \sim U^T [u_3]_\times diag(f^2, f^2, 1)[u_3]_\times U \\ & \sim \begin{bmatrix} u_2^T \\ -u_1^T \\ 0^T \end{bmatrix} diag(f^2, f^2, 1) \begin{bmatrix} u_2 & -u_1 & 0 \end{bmatrix} \end{aligned} \tag{9}$$

The last row and the last column of this matrix equation are zero vectors, so we concentrate on the upper left 2×2 part of the equation:

$$\begin{aligned} & \begin{bmatrix} av_1^T \\ bv_2^T \end{bmatrix} diag(f^2, f^2, 1) \begin{bmatrix} av_1 & bv_2 \end{bmatrix} \\ & \sim \begin{bmatrix} u_2^T \\ -u_1^T \end{bmatrix} diag(f^2, f^2, 1) \begin{bmatrix} u_2 & -u_1 \end{bmatrix} \end{aligned} \tag{10}$$

Expanding the above equation:

$$\begin{aligned} & \begin{bmatrix} a^2(v_{13}^2(1-f^2)+f^2) & ab(v_{13}v_{23}(1-f^2)) \\ ab(v_{13}v_{23}(1-f^2)) & b^2(v_{23}^2(1-f^2)+f^2) \end{bmatrix} \\ & \sim \begin{bmatrix} u_{23}^2(1-f^2)+f^2 & -u_{13}u_{23}(1-f^2) \\ -u_{13}u_{23}(1-f^2) & u_{13}^2(1-f^2)+f^2 \end{bmatrix} \end{aligned} \tag{11}$$

We obtain two linear equations and a quadratic one to calculate f:

$$\begin{aligned} & f^2(au_{13}u_{23}(1-v_{13}^2)+bv_{13}v_{23}(1-u_{23}^2)) \\ & +u_{23}v_{13}(au_{13}v_{13}+bu_{23}v_{23}) = 0 \end{aligned} \tag{12}$$

$$\begin{aligned} & f^2(av_{13}v_{23}(1-u_{13}^2)+bu_{13}u_{23}(1-v_{23}^2)) \\ & +u_{13}v_{23}(au_{13}v_{13}+bu_{23}v_{23}) = 0 \end{aligned} \tag{13}$$

$$\begin{aligned} & f^4(a^2(1-u_{13}^2)(1-v_{13}^2)-b^2(1-u_{23}^2)(1-v_{23}^2)) \\ & + f^2(a^2(u_{13}^2+v_{13}^2-2u_{13}^2v_{13}^2)-b^2(u_{23}^2+v_{23}^2 \\ & -2u_{23}^2v_{23}^2))+(a^2u_{13}^2v_{13}^2-b^2u_{23}^2v_{23}^2)=0 \end{aligned} \quad (14)$$

There are two focus lengths f_1 and f_2 computed by Eq.(14). The two focus lengths are checked with Eq.(12) and Eq.(13). The focus length, which makes Eq.(12) and Eq.(13) both closer to zero, will be chosen. Then the calibration matrix K also will be determined.

4.2 Computation of exterior parameters

Camera exterior parameters includes rotation matrix R and translation vector t. Rotation matrix $R=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, it is an orthogonal matrix. Translation vector $t=[t_x \quad t_y \quad t_z]^T$.

We compute camera exterior parameters based on essential matrix E [21]. The essential matrix implicates information of camera movement (rotation matrix R and translation vector t). As known from epipolar equation, the essential matrix E can be expressed by fundamental matrix F and intrinsic matrix K:

$$E = K^T F K \quad (15)$$

rotation matrix R and translation vector t can be obtained by the singular value decomposition of essential matrix E:

$$E = U \Lambda V^T \quad (16)$$

$$R = U \begin{bmatrix} 0 & -k & 0 \\ k & 0 & 0 \\ 0 & 0 & s \end{bmatrix} V^T \quad (17)$$

$$t \sim u_3 \quad (18)$$

where $k = \pm 1$, $s = \det(U) \cdot \det(V)$. So the R will have two values, R_1 (when $k=1$) and R_2 (when $k=-1$). u_3 is the third column of matrix U. By above equations, the exterior parameter can be computed.

5. 3D coordinates computation of feature points and mapping texture

After determination of intrinsic parameter, 3D coordination of the feature points can be computed by triangulation[21]. Suppose that the 3D homogeneous coordinate of space point M is $[x \quad y \quad z \quad w]^T$, and its 2D homogeneous coordinate in image is $m=[u \quad v \quad 1]^T$. They are related by projection matrix:

$$[u \quad v \quad 1]^T = P \cdot [x \quad y \quad z \quad w]^T \quad (19)$$

where P is a 3×4 projection matrix. It can be expressed with intrinsic matrix and exterior matrix:

$$P = K \cdot [R \mid t] \quad (20)$$

We suppose that: $P = K \cdot [I \mid 0]$ is the projection matrix of one image. Based on exterior parameter, there are four choices for projection matrix of the other image: $P_1' = K \cdot [R_1 \mid t]$, $P_2' = K \cdot [R_1 \mid -t]$, $P_3' = K \cdot [R_2 \mid t]$ or $P_4' = K \cdot [R_2 \mid -t]$. If $m=[u \quad v \quad 1]^T$ and $m'=[u' \quad v' \quad 1]^T$ are points in two images respectively corresponding to 3D point $M=[x \quad y \quad z \quad w]^T$, we can write the projection equation as follows:

$$\begin{cases} m = PM \\ m' = P'M \end{cases} \quad (21)$$

Where

$$P=\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \end{bmatrix}, P'=\begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} \\ \beta_{31} & \beta_{32} & \beta_{33} & \beta_{34} \end{bmatrix}.$$

Expanding above equation, the projection equation can be written as:

$$\begin{bmatrix} u\alpha_{31}-\alpha_{11} & u\alpha_{32}-\alpha_{12} & u\alpha_{33}-\alpha_{13} & u\alpha_{34}-\alpha_{14} \\ v\alpha_{31}-\alpha_{21} & v\alpha_{32}-\alpha_{22} & v\alpha_{33}-\alpha_{23} & v\alpha_{34}-\alpha_{24} \\ u'\beta_{31}-\beta_{11} & u'\beta_{32}-\beta_{12} & u'\beta_{33}-\beta_{13} & u'\beta_{34}-\beta_{14} \\ v'\beta_{31}-\beta_{21} & v'\beta_{32}-\beta_{22} & v'\beta_{33}-\beta_{23} & v'\beta_{34}-\beta_{24} \end{bmatrix} \cdot M = 0 \quad (22)$$

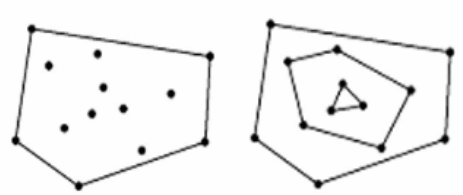
With known image matched points m and m', 3D homogeneous coordinate of M can be computed using the singular value decomposition through P_1', P_2', P_3' and P_4' respectively. One of P_1', P_2', P_3' and P_4', by which the most 3D points in front of camera can be obtained, is selected as the final projection matrix P'. Once the

projection matrix P' is determined, we can pick some matched feature points between two images, compute the 3D coordinates, and build 3D structure (As shown in Fig. 3).

Fig. 3 Matched feature points and 3D coordinates.

After building 3D structure, we can map the 2D texture from images to the surface of the 3D structure to improve the visual effect. Through the texture mapping, we obtain a "photo-realistic" 3D surface model. In this paper, we divide 2D image into triangular meshes by feature points, then map the triangular textures onto the corresponding 3D triangular meshes one by one.

Triangulation is an important step before texture mapping. Firstly we need to link the feature points to form the polygons, and then triangulate them. There are several sorts of polygon [18], such as self-intersection polygon, non self-intersection polygon with overlap and non self-intersection polygon without overlap. In this paper, Convex Hull Algorithm [22] is used to link the feature points become no-overlap polygon without self-intersection in 2D images. Among some discrete points, convex hull is the smallest convex polygon that contains these points. As shown in Fig.4 (a). After iterative computing, we obtain several convex hulls as shown in Fig.4 (b). Then we divide these convex hulls with Delaunay triangulation algorithm [18]. The principal idea of Delaunay triangulation algorithm is "min max angle criterion", namely the sum of inner angles is the biggest. And these angles are the smallest in each triangle. This idea is also beneficial to mapping texture. In real scene, the result of triangulation is not perfect generally, so we need to modify it interactively to obtain the correct triangulation. Fig.5 is the triangulation according to feature points that are shown in Fig.3.



(a) Convex hull in out layer (b) Convex hull in each layer

Fig. 4 Convex hulls from discrete points.

With the application of programming interface OpenGL, we can map triangular textures to the 3D structure one by one. The triangular texture images can be fused from two images or extracted directly from one image.

Once the 3D surface model is reconstructed, we can look around it from any viewpoint. Fig.6 is two views of the reconstructed 3D model from two different viewpoints. Fig.7 is the additional examples.

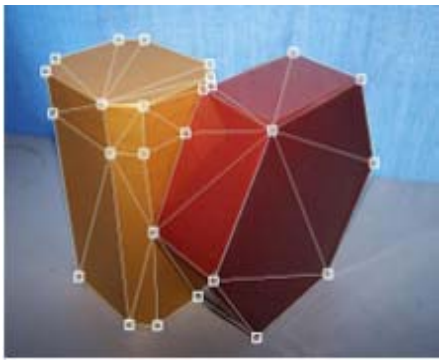


Fig. 5 Triangulation.

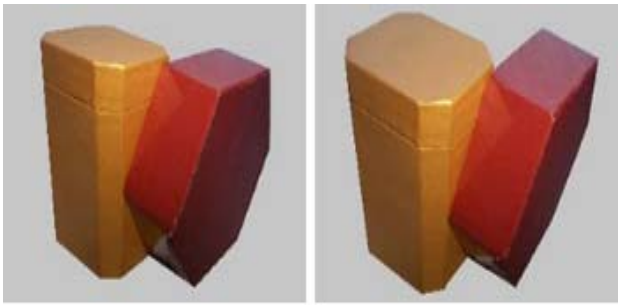


Fig. 6 Two views of 3D surface model.

(a)Two original images of a building and a view of the reconstructed 3D surface

(b)Two original images of a stele and a view of the reconstructed 3D surface

Fig. 7 Examples of 3D surface reconstruction

6. Conclusion

In this paper, 3D modeling from two uncalibrated images is investigated and realized. We combine the sobel edge detection and SUSAN algorithm to enhance computation speed in the process of corner detection, and improve the weighted normalization linear algorithm to compute more accurate fundamental matrix. Based on the structure of scene, we choose some feature points to reconstruct the 3D surface structure, divide the 2D image into triangular meshes with these points, and then map triangular textures onto the surface of 3D structure to reconstruct 3D surface model. Experimental results prove that our approach is accurate and efficient.

References

[1] Tomasi. C, "Pictures and trails. A new framework for the computation of shape and motion from perspective imagesequences[A]," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition[C], pp.913-918, 1994 (in Seattle, WA).

[2] Pollefeys. M, "Obtaining 3D models with a hand-held camera[A]," Proceedings of Vision Modeling and Visualization[C], pp.195-202,2001 (in Stuttgart, Germany).

[3] Debevec. P, Taylor. C, Malik. J, "Modeling and rendering architecture from photographs: A hybrid geometry and image-based approach[A]," In: Computer Graphics Proceedings, Annual, Conference Series[C], ACM SIGGRAPH, pp.11-22, 1996 (in Louisiana, New Orleans).

[4] Rodriguez. T, Sturm. P, Gargallo. P, "Photorealistic 3D Reconstruction from Handheld Cameras[J]," Machine Vision and Applications IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16(4), pp.246-257, 2005.

[5] Pollefeys. M, "Visual Modeling with a Hand-Held Camera[J]," International Journal of Computer Vision vol.59(3), pp.207-232, 2004.

[6] Pollefeys. M, Leuven. K. U, "3D modeling from images[J]," Lecture Notes, July, 2000.

[7] Colombo. C, Del Bimbo. A, Pernici. F, "Metric 3D reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view[J]," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27(1), pp.99-114, 2005.

[8] Lhuillier. M, Quan. L, "A Quasi-Dense Approach to Surface Reconstruction from Uncalibrated Images[J]," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27(3), pp.1-16, 2005.

[9] Park. I.K, Zhang. H, Vezhnevets. V, "Image-based 3D face modelling system," Eurasip Journal on Applied Signal Processing, vol.13, pp.2072-2090, Aug, 2005.

[10] Mueller. K. A, "3D Reconstruction of Natural Scenes with View-Adaptive Multi-Texturing[A]," 2nd International Symposium on 3D Data Processing, Visualization and Transmission[C], pp.116-123, 2004.

[11] Park. J. S, "Interactive 3D reconstruction from multiple images: A primitive-based approach[J]," Pattern Recognition Letters, vol.26(16), pp.2558-2571, 2005.

[12] Yu. Y, Ferencz. A, Malik. J, "Extracting objects from range and radiance image[J]," IEEE Transactions on Visualization and Computer Graphics, vol.7(4), pp.351-364, 2001.

[13] Faugeras. O, "Stratification of 3-D vision: Projective, affine, and metric representations[J]," Journal of the Optical Society of America A, vol.12(3), pp.465-484, 1995.

[14] Matusik. W, Pfister. H, Ngan. A, "Image-based 3D photography using opacity hulls[A]," In: Computer graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pp.427-437, 2002 (in San Antonio, Texas).

[15] Liu. G, Peng. Q.S, Bao. H. J, "Review and Prospect of Image-Based Modeling Techniques[J]," Journal of Computer Aided Design . Computer Graphics, vol.17(1), pp.18-27, 2005.

[16] Smith. S. M, Brady. J. M, "SUSAN-a new approach to low level image processing[J]," International Journal of Computer Vision, vol.23(1), pp.45-78, 1997.

[17] Chen. Z. Z, Wu. C. K, Liu. Y, "A Weighted Normalization Algorithm for Estimation of Fundamental Matrix[J]," Journal of Software, vol.12(3), pp.420-427, 2001.

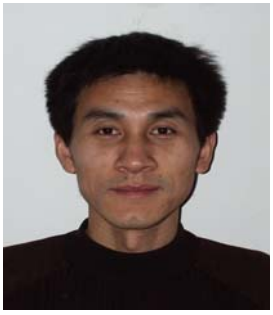
[18] Ding. Y. X, Xia. J. C, Wang. Y, "Delaunay Triangulation of Arbitrary Polygons[J],"Chinese Journal in Computers, vol.17(4), pp.270-275,1994.

[19] Sturm. P, "On focal length calibration from two views[A]," CVPR-INT. CONF, On Computer Vision and Pattern Recognition[C] , pp.145-150, 2001.

[20] Zeller. C, Faugeras. O, "Camera Self-Calibration from Video Sequences: the Kruppa Equations Revisited," Research Report 2793, 1996 (in INRIA, France).

[21] Torr. P. H. S, "A structure and motion toolkit in matlab "Interactive Adventures in S and M," Microsoft Research, Technical Report MSR-TR-2002-56, 2002.

[22] Joseph. O'Rourke, "Computational Geometry in C [M]," Cambridge University Press, 1998.



Shungang Hua received the B.E. and M. E. degrees from Bejing institute of technology in 1986 and 1989, respectively. He received the Dr. degree from Dalian univ. of technology in 2004. He is currently an associate professor in the School of Mechanical Engineering, Dalian univ. of technology. His research interests include computer graphics, image processing, virtual reality.

Ting Liu received the B.E. degree in Mechanical Engineering from Dalian Univ. of Technology in 2006. She is a graduate student in the School of Mechanical Engineering, Dalian Univ. of Technology. Her research interests include computer graphics, image processing.