Linear camera calibration and pose estimation from vanishing points

Jun Chu (储 珺)[1*], Li Wang (王 丽)[2], Ruina Feng (冯瑞娜)[3], and Guimei Zhang (张桂梅)[2]

[1]*Software College of Nanchang Hangkong University, Nanchang 330063, China*

[2]*Aviation Manufacturing Engineering College of Nanchang Hangkong University, Nanchang 330063, China*

[3]*Aircraft Engineering College of Nanchang Hangkong University, Nanchang, 330063, China*

[*]*Corresponding author: chujun99602@163.com*

Received November 7, 2011; accepted January 6, 2012; posted online April 27, 2012

In order to broaden the scope of application and ensure the calibration precision, a new method of linear calibration by making fully use of vanishing point attributes is proposed. The method dose not need any rigorous restrictions, and solves the self-calibration problem with only five vanishing points in two digital images, which are arbitrarily taken by a handheld digital camera. Furthermore, another approach for camera's pose estimation is also put forward without any strict controlled motions. The experimental results of both computer simulation and real images show that the calibration algorithm is effective, feasible, and robust.

OCIS codes: 100.2000, 100.2960, 100.6890.

doi: 10.3788/COL201210.S11007.

Camera calibration is one of the main research contents in computer vision, because it plays an important role in two-dimensional (2D) image reconstruction, and camera calibration is the basic premise of following three-dimensional (3D) modeling work.

Caprile *et al.* firstly put forward the idea of using vanishing point for self-calibration[1]. They proved three attributes of the vanishing point for the first time, and pointed out the relationship between the vanishing points and the camera intrinsic parameters under unit aspect ratio assumption. However, this assumption may not be applicable to some digital cameras because the aspect ratio is affected by camera fabrication and installation[2]. In order to further increase the accuracy of calibration, an algorithm was proposed by utilizing six main vanishing points in two images with unknown aspect ratios[3]. This algorithm requires that every image should provide at least three main vanishing points. However, according to the clairvoyant principle of vanishing points[4], vanishing point's number is connected with the shooting angle that three vanishing points could only be generated just when every axis of Manhattan directions[5] intersects with the projection plane. On the other hand, partially occluded images or detection fault may also cause a loss of vanishing point's number. So, the application of this method is limited.

Self-calibration using vanishing points can also be realized by means of other methods. The intrinsic parameters could be determined by two vanishing points under the hypothesis that the image center was the camera projection center, which was defined as principal point as well[6]. However, that principal point is actually separate from the image center due to the camera fabrication process. Another self-calibration method required that some line segments in the image should be symmetrical and equal. Then a square is certainly generated. And the orthogonality of the square's diagonal lines is useful for calibration[7]. Obviously, this method is only satisfied in special application background. He *et al.*[8] proposed a self-calibration algorithm that uses vanishing points and image edges in each image, but this approach required to make a pure controlled translation motion between the viewpoints.

According to restriction equations of the absolute conic (IAC)[9] and the dual image of the absolute conic (DIAC)[10], the camera intrinsic parameters can also be determined by using vanishing point. In fact, the two methods are similar. For lack of making fully use of vanishing point properties, those solving process is a bit complicated. And, the approach establishing IAC or DIAC constraint equations about homography matrix should need at least 3 images of different viewpoints[11,12]. Furthermore, there is a need to strictly control the camera movements, which include at least one translation motion and two rigid motions[12].

Besides those methods by using vanishing point, there are some other calibration methods. For example, the camera intrinsic parameters can be determined by solving kruppa equations[2,13]. This approach requires three image views and needs to know fundamental matrix between each pair of views. Hence it brings a problem that the equations are always nonlinear. And, Essence matrix is also found useful for inter parameters' computation[14]. However, this method should firstly solve the fundamental matrix, which is easily turned into local optimum.

This letter presents a new method of linear calibration by making fully use of vanishing point properties. The main contributions of this paper are following three points:

(1) Reduce restrictions of camera self-calibration, such as unit aspect ratio assumption or assumed position of principal point, etc. And get the inter parameters with high precision by a simple method.

(2) To further broaden the scope of calibration, we solve camera self-calibration problem with only five vanishing points in two images, which are arbitrarily taken by a handheld digital camera.

(3) Give a simple method to estimate the camera's exterior orientation without any controlled motions between the viewpoints. The parameter matrixes are just

obtained according to vanishing points and the acquired inter parameters above.

In the perspective projection model (or pinhole model), arbitrary point M in space corresponds to one point m in the image, the relational expression is presented as follows:

$$\lambda m = PM = K\left[R,T\right]M,$$

where λ is a nonzero scale factor, R and T are the exterior orientation parameters of the camera and represent 3×3 rotation matrix and 3×1 translation matrix, respectively. K is the inter parameters, its matrix expression is

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

And $f_x = f/d_x$, $f_y = f/d_y$, f is the focal length, d_x and d_y are the physical size in u and v axes of camera coordinates, d_x/d_y is the camera pixel aspect ratio. s refers to the skew factor, which usually equals 0. (u_0, v_0) is the principal point.

A series of parallel lines in space can intersect at one point in image plane after perspective projection, and the point is just called vanishing point. Parallel lines projection in different directions can form different vanishing points. Notations V_1, V_2, V_3 are called the three main vanishing points corresponding to three mutually orthogonal directions of X, Y, Z axes in space. Consequently, there are at most three main vanishing points of a cube (as shown in Fig. 1). A main vanishing point (hereinafter referred to as the vanishing point) will thereupon decrease as any axis of the three X, Y, Z axes parallels to the image plane. Vanishing points have some special properties, and we list a few related properties, which have been detailed demonstrated in Refs. [1,15].

Property 1 In the camera system, the unit vectors of the three vanishing points generated by the three groups of lines in space are pairwise orthogonal[1].

Assuming sets L_1, L_2, L_3 are three groups of orthogonal lines in 3D space, the coordinates of each vanishing point is $V_i \equiv (u_{xi}, v_{yi}, f)$, $(i = 1, 2, 3)$ in the camera coordinate system. And l_1, l_2, l_3 correspond to the unit vectors of the three vanishing points, so there is

$$\begin{cases} l_1 \cdot l_2 = 0 \\ l_1 \cdot l_3 = 0 \\ l_2 \cdot l_3 = 0 \end{cases} \Rightarrow \begin{cases} u_{x1}u_{x2} + v_{y1}v_{y2} + f^2 = 0 \\ u_{x1}u_{x3} + v_{y1}v_{y3} + f^2 = 0 \\ u_{x2}u_{x3} + v_{y2}v_{y3} + f^2 = 0 \end{cases}.$$

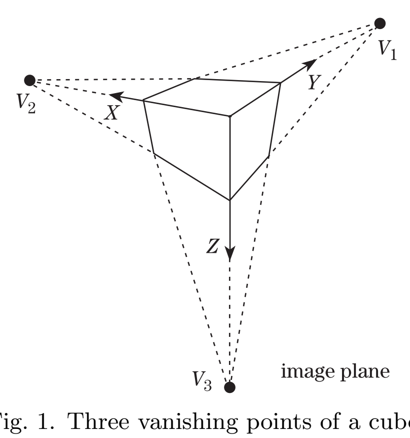


Fig. 1. Three vanishing points of a cube.

Property 2 Transforming from the world system to camera coordinate system, the coordinate transformation of vanishing point is only concerned with the rotation matrix R, and has nothing to do with the translation matrix T[1].

Let V' be the vector of vanishing point in the world system, and V be its related vector in the camera coordinate system, then we have $V = R \cdot V'$.

Property 3 The lines generated the vanishing point parallel to the line connected between the vanishing point and the optical center of camera[15].

Suppose (u_i, v_i) to be the coordinate of vanishing point $V_i\,(i = 1, 2, 3, 4, 5)$ in the image plane, then $[(u_i - u_0)\,d_x, (v_i - v_0)\,d_y, f]$ is the related coordinate in camera system. Assume that we can just get five vanishing points in the two digital images. And, there are three vanishing points in the first image, while two vanishing points in the second image, then the following constraint equations can be established by property 1:

$$(u_1 - u_0)(u_2 - u_0)\,d_x^2 + (v_1 - v_0)(v_2 - v_0)\,d_y^2 + f^2 = 0, \tag{1}$$

$$(u_1 - u_0)(u_3 - u_0)\,d_x^2 + (v_1 - v_0)(v_3 - v_0)\,d_y^2 + f^2 = 0, \tag{2}$$

$$(u_2 - u_0)(u_3 - u_0)\,d_x^2 + (v_2 - v_0)(v_3 - v_0)\,d_y^2 + f^2 = 0, \tag{3}$$

$$(u_4 - u_0)(u_5 - u_0)\,d_x^2 + (v_4 - v_0)(v_5 - v_0)\,d_y^2 + f^2 = 0. \tag{4}$$

These can be further rewritten as

$$(u_1 - u_0)(u_2 - u_0) + (v_1 - v_0)(v_2 - v_0)(d_y/d_x)^2 + (f/d_x)^2 = 0, \tag{5}$$

$$(u_1 - u_0)(u_3 - u_0) + (v_1 - v_0)(v_3 - v_0)(d_y/d_x)^2 + (f/d_x)^2 = 0, \tag{6}$$

$$(u_2 - u_0)(u_3 - u_0) + (v_2 - v_0)(v_3 - v_0)(d_y/d_x)^2 + (f/d_x)^2 = 0, \tag{7}$$

$$(u_4 - u_0)(u_5 - u_0) + (v_4 - v_0)(v_5 - v_0)(d_y/d_x)^2 + (f/d_x)^2 = 0. \tag{8}$$

Let $t_1 = (d_y/d_x)^2$ and $t_2 = (f/d_x)^2$, unfold the Eq. (5), and then make it to subtract Eqs. (6–8), respectively. After rearranging, it follows that

$$(u_2 - u_3)u_1 - (u_2 - u_3)\,u_0 + (v_2 - v_3)\,v_1 t_1 - (v_2 - v_3)\,v_0 t_1 = 0, \tag{9}$$

$$(u_1 - u_3)u_2 - (u_1 - u_3)\,u_0 + (v_1 - v_3)\,v_2 t_1 - (v_1 - v_3)\,v_0 t_1 = 0, \tag{10}$$

$$u_1 u_2 - u_4 u_5 - (u_1 + u_2 - u_4 - u_5)\,u_0 + (v_1 v_2 - v_4 v_5)\,t_1 - (v_1 + v_2 - v_4 - v_5)\,v_0 t_1 = 0. \tag{11}$$

If let $v_0 t_1$ be unity, then the three unknown number u_0, v_0, and t_1 can be solved linearly by the three independent equation from Eqs. (9) to (11). With the obtained parameters, then substituting them back into the Eq. (5), the unknown t_2 can also be calculated. And immediately it is available that

$$f_x = f/d_x = \sqrt{t_2}, f_y = f/d_y = \sqrt{t_2}/\sqrt{t_1}.$$

Hence, all of the inter parameters of K are already determined, and it is achieved just only by vanishing point's property with mathematical derivation method.

From property 2, we can get $V = R \cdot V'$, and V is a 3×3 unit matrix formed by $V_i\,(i = 1, 2, 3)$ with the column vectors V_1, V_2, and V_3 in the camera system. In the same way, V' is the related unit vector formed by $V_i'\,(i = 1, 2, 3)$ in the world system. The directions of the main vanishing points are X, Y, and Z axes respectively in the world system, and there are $V_1'\,(1, 0, 0)$, $V_2'\,(0, 1, 0)$, $V_3'\,(0, 0, 1)$, so we have

$$R = V = [V_1\ V_2\ V_3] = [R_1\ R_2\ R_3]$$
$$= \begin{vmatrix} \frac{u_{x1}}{\sqrt{u_{x1}^2+v_{y1}^2+f^2}} & \frac{u_{x2}}{\sqrt{u_{x2}^2+v_{y2}^2+f^2}} & \frac{u_{x3}}{\sqrt{u_{x3}^2+v_{y3}^2+f^2}} \\ \frac{v_{y1}}{\sqrt{u_{x1}^2+v_{y1}^2+f^2}} & \frac{v_{y2}}{\sqrt{u_{x2}^2+v_{y2}^2+f^2}} & \frac{v_{y3}}{\sqrt{u_{x3}^2+v_{y3}^2+f^2}} \\ \frac{f}{\sqrt{u_{x1}^2+v_{y1}^2+f^2}} & \frac{f}{\sqrt{u_{x2}^2+v_{y2}^2+f^2}} & \frac{f}{\sqrt{u_{x3}^2+v_{y3}^2+f^2}} \end{vmatrix},$$

where $u_{xi} = (u_i - u_0)\,d_x$, $v_{yi} = (v_i - v_0)\,d_y$, $(i = 1, 2, 3)$. Although the inter parameters have been determined in the previous sections, we cannot directly calculate the matrix yet, so we have to do some mathematics transformation, for example

$$\frac{u_{x1}}{\sqrt{u_{x1}^2 + v_{y1}^2 + f^2}} = \frac{(u_1 - u_0)d_x}{\sqrt{(u_1 - u_0)^2 d_x^2 + (v_1 - v_0)^2 d_y^2 + f^2}}$$
$$= \frac{(u_1 - u_0)}{\sqrt{(u_1 - u_0)^2 + (v_1 - v_0)^2 t_1 + t_2}}. \tag{12}$$

Similarly, other matrix items should also be solved in the same way. After all of that, the whole rotation matrix R can be computed.

While for the image with two vanishing points detected, the computation of R can be realized according to the orthogonality of rotation matrix. If R_1 and R_2 are available, while R_3 is unavailable, so we can deal with: $R_3 = R_1 \times R_2$.

In order to simply recover the translation matrix appropriating for conventional vision system, a geometrical derivation method is adopted to solve the matrix T. As shown in Fig. 2, sign o is the camera optical center, AB is the known segment in 3D space, and we assume that it has unit length; sign ab is the image projection of AB, and point v is the vanishing point formed by line AB. It is known from property 3 that $AB//ov$, then we draw auxiliary line $ab'//AB$. Assuming that the point A is the original point in space, then the vector $\overrightarrow{oA}$ is the translation vector between the world coordinate system and the camera coordinate system, therefore we have: $T = \overrightarrow{oA}$.

Due to similar triangles $\Delta bab' \sim \Delta bvo$ and $\Delta oab' \sim \Delta oAB$, it can be derived as $\frac{|oA|}{|oa|} = \frac{|AB|\cdot|vb|}{|ab|\cdot|ov|} = \frac{|vb|}{|ab|\cdot|ov|}$. For $\frac{|oA|}{|oa|} = \frac{\gamma}{d_x}$, in which γ is a computable constant got from the continued equality above. Then, the translation matrix T can be recovered from the already obtained inter parameters in the previous sections, and there is

$$T = \overrightarrow{oA} = (\gamma/d_x) \cdot \overrightarrow{oa} = \gamma \left[(u_a - u_0), (v_a - v_0)\sqrt{t_1}, \sqrt{t_2}\right]^{\mathrm{T}}.$$

This experiment uses a group of simulated 3D spatial points to verify the efficiency of our algorithm, and compares the result in Ref. [3]. In order to ensure comparability of the experimental results, this letter adopts the same simulation data in Ref. [3]. The simulation data is from by a small house and two blocks as shown in Fig. 3. The inter parameters of the simulative camera is set as u_0=50, v_0=50, f_x=400, f_y=600. We get three simulative images from every different viewpoint. The exterior orientation of the first viewpoint is set as T_1= $(30, 40, 60)^{\mathrm{T}}$, R_1= $(\pi/15, \pi/10, \pi/15)$, and $\pi/15$, $\pi/10$, $\pi/15$ are the rotation angle that camera around the X, Y, Z axes, respectively. The second viewpoint is assumed as: T_2= $(60, 40, 80)^{\mathrm{T}}$, R_2= $(\pi/15, -\pi/5, \pi/15)$; and then we set the third viewpoint as: T_1= $(60, 15, 80)^{\mathrm{T}}$, R_3= $(\pi/10, -\pi/15, \pi/10)$.

With every two images from the three viewpoints, the inter parameters can be regained by the algorithm presented in this letter. Table 1 gives the coordinates of the vanishing points corresponding to each viewpoint[3]. Table 2 shows the calibration results from the two algorithms; the former result corresponds to our presented algorithm, and the result in the bracket belongs to Ref. [3]. From Table 2, it indicates that the algorithm by this article is effective, and partial result has a higher precision.

In order to further validate the robustness of this algorithm, we adds some noise to the vanishing points in 2&3 viewpoints above, since the algorithm relies heavily on vanishing point's extraction. During every experiment, each group of datum is added with white Gaussian noise of 0 mean values and 0—2.0 variance respectively, and 100 times of independent random experiments are tested at each noise level. The results at different levels are presented in Table 3.

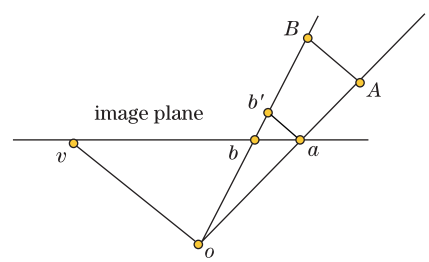


Fig. 2. Analysis graphics of translation vector.

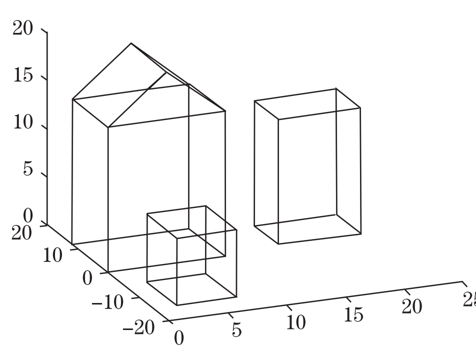


Fig. 3. Simulated 3D scene.

Table 1. Coordinates of Vanishing Points for Every Viewpoint

Viewpoint	Coordinates of Every Viewpoint		
1	(23 861.18,52 622.3)	(889.19,−805.58)	(−82.78,177.53)
2	(−281.15,−2 264.96)	(−2 408.04,966.65)	(136.92,177.53)
3	(−290.58,−1445.21)	(−3 969.92,2 351.03)	(139.40,244.95)

Table 2. Results of Linear Calibration

Parameters	This Algorithm (in the Parentheses for Ref. [3])		
(standard)	1&2	1&3	2&3
u_0(50)	49.82(50.12)	49.71(48.77)	49.99(52.46)
v_0(50)	50.20(49.38)	50.31(51.23)	49.98(51.66)
f_x(400)	400.12(396.13)	399.98(397.50)	400.01(395.48)
f_y(600)	597.94(594.43)	597.72(569.45)	600.02(593.39)

Table 3. Noise Simulation Experimental Results (Mean Value of 100 Times Experiment)

Noise Coefficient	u_0	v_0	f_x	f_y
0.0(standard)	50	50	400	600
0.2	50.0690	50.1047	399.5707	599.3624
0.4	50.4508	50.7312	398.7869	598.0820
0.6	49.8322	49.6296	400.5477	600.8622
0.8	50.2903	50.5797	398.9970	598.5899
1.0	49.8450	49.6961	400.2645	600.3766
1.2	49.9145	49.8422	399.9956	600.1326
1.4	49.5298	48.9053	400.8990	601.6818
1.6	50.0080	50.2064	399.4639	599.2259
1.8	49.9720	49.5630	400.1339	600.3296

Figure 4 is taken by a handheld digital camera of SAMSUNG ES60/VLUU with 4 000×3 000 resolutions. The labeled red dots in Fig. 4 are manually selected, and they are used as contrasts for the subsequent afresh projection[16].

Extract the edges of the original image, and then find out all of the parallel lines corresponding to the three Manhattan directions[5], the optimal intersection point of each group of parallel lines is the vanishing point. The vanishing points of the two images are just the input data which is presented in Table 4. And all of the parameter matrixes can be obtained by our presented algorithms:

$$K_A = K_B = \begin{bmatrix} 4\,049 & 0 & 2\,101.4 \\ 0 & 4\,149.1 & 1\,909.6 \\ 0 & 0 & 1 \end{bmatrix},$$

$$R_A = \begin{vmatrix} -0.5924 & 0.8153 & 0.0334 \\ -0.4608 & -0.3689 & 0.8312 \\ 0.6900 & 0.4536 & 0.5838 \end{vmatrix},$$

$$R_B = \begin{vmatrix} -0.8784 & 0.4774 & 0.0241 \\ -0.3537 & -0.6850 & 0.6751 \\ 0.3306 & 0.5703 & 0.7519 \end{vmatrix},$$

$$T_A = (-51.7846,\ 34.2174,\ 940.8193)^{\mathrm{T}},$$

$$T_B = (44.7,\ 238.4,\ 1247.5)^{\mathrm{T}}.$$

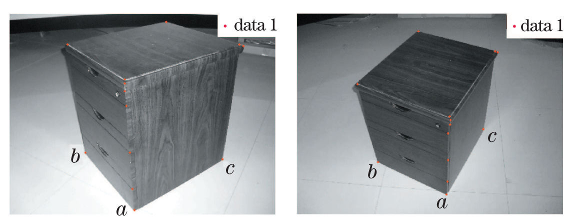
Take an afresh projection[16] with the parameter matrixes, and the deviations from the selected red dots are visible in Fig. 5 after afresh projection. As seen from the figure, afresh projections of green dots are almost coincident with the original marked red dots, just with a slight deviation. After calculation, the mean value of deviation is 4.03 pixels, and the standard deviation is 1.52. Further, the length of segment ab reconstructed is 0.386 m, and its actual length measured is 0.395 m, so the absolute error is 0.009, the relative error is 2.27%; while the length of segment ac reconstructed is 0.448 m, and its actual length is 0.460 m, so the absolute error is 0.012, the relative error is 2.61%.

From Fig. 5 and the results calculated above, the self-calibration parameters obtained by our algorithms is also effective for real images in practical application. In addition, the relative error can be further reduced if the points' coordinates can be extracted accurately, while the points in this experiment are dependent on manual extraction which may produce manual errors.

In conclusion, a new linear calibration approach from two images is proposed, which are arbitrarily taken by a handheld digital camera. This method broadens the scope of calibration that does not need to assume any rigorous restrictions. In addition, we determine both inter parameters and exterior orientation of the camera with only 5 vanishing points in two images. The experimental results show that our algorithm is effective, robust, and has a fine practical application. As the algorithm is dependent on vanishing points, so our further work would continue to study the accurate extraction of the vanishing points in each image.

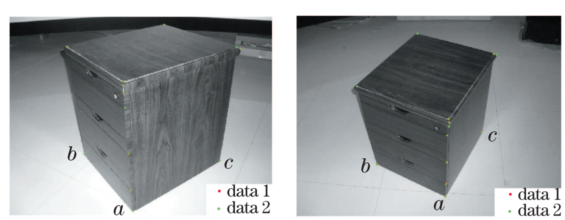
Table 4. Vanishing Points of Real Images

Viewpoint	Coordinates of Vanishing Points		
A	(9379.1,−1383.9)		(2333.1,7674.4)
B	(−8654.7,−2422.3)	(5490.6,−2953.2)	(2231.3,5544.9)



(a) (b)

Fig. 4. Real images of a cabinet and the marked red dots. (a) Viewpoint A; (b) Viewpoint B.



(a) (b)

Fig. 5. Marked red dots and the green dots from afresh projection. (a) Viewpoint A; (b) Viewpoint B.

This work was in part supported by the National Basic Research Program of China (No. 2009CB320902), the Aeronautic Science Foundation of China (No. 2010ZC56004), and the National Natural Science Foundation of China (No. 60954002).

References

1. B. Caprile and V. Torre, Int. J. Comput. Vision. **4,** 127 (1990).
2. Z. Jiang, W. Wu, and M. Wu, in *Proceedings of International Conference on Computer Science and Software Engineering, IEEE Computer Society, Wuhan, China* 1032 (2008).
3. Z. Chen and C. Wu, Journal of Image and Graphics (in Chinese) **8,** 341(2003).
4. Y. He, Journal of Computer-aided Design and Computer Graphics (in Chinese) **17,** 734 (2005).
5. J. M. Coughlan and A. L. Yuille, Neural Comput. **15,** 1063 (2003).
6. E. Guillou, D. Meneveaux, E. Maisel, and K. Bouatouch, Vis. Comput. **16,** 396 (2000).
7. G. Wang, H. Tsui, Z. Hu, and F. Wu, Image Vis. Comput. **23,** 311 (2005).
8. B. W. He and Y. F. Li, Opt. Laser Tech. **40,** 555 (2008).
9. C. K. Fong and W. K. Cham, in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hong Kong, China* 53 (2005).
10. R. Cipolla, T. Drummond, and D. Robertson, in *Proceedings of the British Machine Vision Conference (BMVC), UK* 382 (1999).
11. F. Wu, *Mathematical Method in Computer Vision* (Science Publishing Company, Beijing, 2008).
12. F. Wu and Z. Hu, Acta Automatica Sinica (in Chinese) **28,** 488 (2002).
13. C. Lei, F. Wu, Z. Hu, and H. T. Tsui, in *Proceedings of the 16*th *International Conference on Pattern Recognition* 308 (2002).
14. Z. Jiang and W. Wu, Journal of Image and Graphics (in Chinese) **15,** 565 (2010).
15. Y. Liu, Y. Wu, and M. Wu, in *Proceedings of the International Conference on Image and Graphics, Hong Kong, China* 460 (2004).
16. Z. Chen, Y. Liu, and C. Wu, Progress in Natural Science (in Chinese) **12,** 975 (2002).