Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **5**, September **2018**

Self-Calibration Based View-Invariant Hand Gesture Trajectory Analysis

Mo-yi Zhang, Qiu-yu Zhang, Hong-xiang Duan and Hai-yan Chen

School of Computer and Communication
Lanzhou University of Technology
Lanzhou, 730050, P. R. China
Corresponding Author: zhangqylz@163.com

Received December, 2017; revised May, 2018

Abstract. *In the process of gesture trajectory analysis on monocular vision, if the camera poses are different, the same gesture trajectory of a moving object can project onto different paths, which will affect the recognition and application of the gesture trajectory. To solve this problem, using gesture's motion points abstracted, we constructed a camera self-calibration model to compute the pose of the camera. Then, the trajectory was re-projected to obtain the orthographic projected trajectory which is view-invariant. At last, the re-projected trajectory was used to recognize. Unlike the previous trajectory recognition studies, in a large range of viewpoints, this method can be used to recognize the gesture trajectory and can achieve a high average recognition rate. In our experiments, the correctness, and effectiveness of the method were analyzed. In the gesture drawing and gesture recognition experiment, the model increased the trajectory recognition accuracy rate by 22%. The proposed model can be used widely in camera pose calibration and view-invariant trajectory analysis.*
Keywords: Camera Pose Self-Calibration, Gesture Trajectory, Human-Computer Interaction, Planar Rectification, View-Invariant

1. **Introduction.** Hand gesture recognition is an important part of the behavior recognition, and it improves the efficiency and the naturalist of human-computer interaction. Gesture recognition consists of gesture segmentation [1], tracking, feature extraction and gesture recognition. Traditional gesture trajectory recognition methods include the application of Clustering [2], Neural Networks [3], Hidden Markov Models (HMM) [4], and related improved methods. However, some extrinsic factors are barriers for the development of behavior recognition, such as the requirement that the human action planes are perpendicular to the camera optical axis, which sometimes is difficult to implement. Consequently, a view-invariant recognition model based on the analysis of projected trajectory data is a key problem that urgently needs to be solved.

Research on view-invariant trajectory recognition can be summarized into the following three categories. One approach is to extract 3D trajectory features. Pitsikalis et al. [5] used a Kinect depth camera to extract hand gestures, voice, and human body gestures in order to realize multi-model human-computer interaction. Ghaleb et al. [6] also used stereo vision in their work, combining a Conditional Random Field Algorithm (CRF) and Support Vector Machine (SVM) for gesture location and recognition. Using 3D vision, we can find the 3D motion of an object easily, but this approach requires a depth camera, multi-camera, or stereo camera. This equipment limits the method's range of application.

A second approach is to extract view-invariant features in motion, including the cross ratio feature [7] and 3D facets histogram of projection transformation [8]. However, the results are heavily dependent upon the contour extraction in image preprocessing.

A third approach uses the corresponding relation of feature in a series of images to recover the spatial information of the image. This method is flexible and easy to apply. When 3D information is recovered from 2D features, the imaging principle of the camera combined with the knowledge of projective geometry is needed to build the complex model. Li et al. [9] proposed a relaxation-based objective function, which utilized both smoothness and geometric constraints, posing articulated trajectory reconstruction as a non-linear optimization problem. However, this method aimed at special trajectories. Park et al. [10] of Carnegie Mellon University presented a linear solution for reconstructing the 3D trajectory of a moving point from its correspondence in a collection of 2D perspective images. In this solution, though, the camera's pose should be known first. When the trajectory of the target is arbitrary, it's necessary to know the pose of the camera in order to recover the 3D information of the trajectory from the trajectory's images. But the previous calibration methods were based on static calibration images [11,12] or based on some of the constraints. They cannot apply to the specific conditions of the moving target in trajectory analysis [13,14].

In view of the above challenges, for this research, we proposed a view-invariant gesture trajectory recognition method. First, we analyzed the motion constraints of feature points of an object. Second, we built the camera self-calibration model to estimate the camera pose, and the estimated results were used for planar rectification. Finally, we realized view-invariant gesture trajectory feature extraction and recognition.

Compared with current research in the field, our study makes three significant contributions. (1) The proposed method facilitates view-invariant gesture trajectory analysis. (2) A camera pose self-calibration model is built using a series of motion images, whereas previous methods use static calibration images. (3) The proposed method improves the average recognition accuracy in large extent.

The rest of this paper is organized as follows. Section 2 presents the camera pose self-calibration model, including the motion constraints of feature points, the building of the self-calibration model, the theory solution of the model, the description of the model related properties, and the model parameters estimation. Section 3 describes the image planar rectification. Section 4 describes the view-invariant gesture trajectory recognition method. Section 5 gives the experimental results and performance analysis as compared with other related methods. Finally, we present our conclusions in Section 6.

2. **Camera Pose Self-Calibration Model.**

2.1. **Camera imaging model.**

The diagram of the camera imaging model is shown in Fig. 1. As shown in Fig. 1, the image physical coordinate system is composed of o_c, x_c, y_c, the image pixel coordinate system is composed of of x_f, y_f, and the camera coordinate system is composed of O, X_c, Y_c, Z_c . According to the camera pinhole imaging theory, in the camera coordinate system, the target point $Q(X, Y, Z)$, image point $q(x, y, f)$, and the optical center of the camera O are in a line, which can be expressed as follows:

$$x = f\left(\frac{X}{Z}\right), y = f\left(\frac{Y}{Z}\right)$$

In practice, the principal point of the image may not be at the center of the projected image. Therefore, the deviation of the optical axes C_x and C_y is introduced in camera

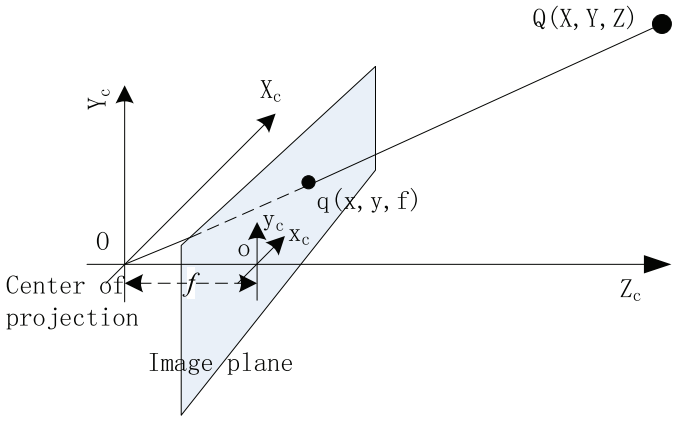

FIGURE 1. Camera imaging model

imaging model. Similarly, since a single pixel in the low-cost imaging instrument is rectangular rather than square, two focal length parameters f_x and f_y are introduced, and the focal length is measured in pixels. The camera model follows the following function:

$$x = f_x\left(\frac{X}{Z}\right) + C_x, y = f_y\left(\frac{Y}{Z}\right) + C_y$$

We can translate the function into matrix form:

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1}$$

2.2. **Motion constraints of feature point.** The diagram of the camera pose self-calibration model is shown in Fig. 2.

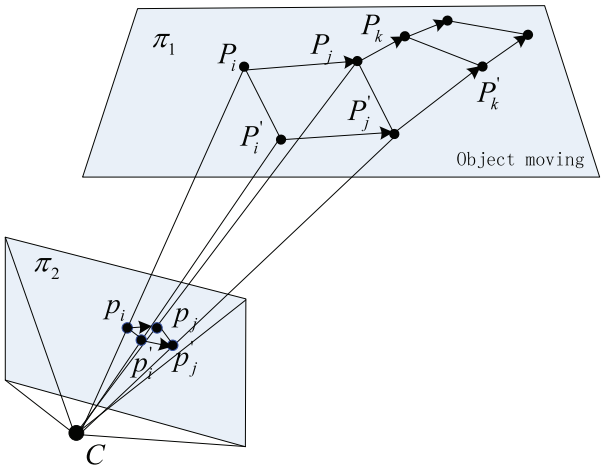

FIGURE 2. Camera pose self-calibration model

As shown in Fig. 2, the target motion plane is π_1, and the image plane is π_2. There are two moving feature points p_i, p_i' in image plane π_2, in which i indicates that the moving object is in i time. The coordinates of p_i, p_i' in the image physical coordinate

system are $p_i(x_i, y_i)$, $p_i'(x_i', y_i')$. The corresponding points on target in motion plane π_1 are $P(X_i, Y_i, Z_i)$,$P'(X_i', Y_i', Z_i')$ in camera coordinate system. Moving feature points can satisfy the camera imaging model of Eq. (1).

Since the target is moving on the plane, the target feature points meet the planar equation in Eq. (2).

$$AX_i + BY_i + CZ_i + D = 0 \tag{2}$$

In the movement of the rigid body, the distance between the feature points remains the same, which can be expressed as following:

$$|P_iP_i'| = |P_jP_j'| \tag{3}$$

The mathematical model is composed of Eq. (1), Eq. (2), and Eq. (3). The unknown parameters in the model are A, B, C, D, X_i, Y_i, Z_i.

2.3. **Model theoretic solution.** $p_i(x_i, y_i)$ is a feature point in the image physical coordinate system. We put $p_i(x_i, y_i)$ into Eq. (1), and it can be transformed as following:

$$X_i = \frac{(x_i - C_x)Z_i}{f_x}, Y_i = \frac{(y_i - C_y)Z_i}{f_y} \tag{4}$$

When we put Eq. (4) into Eq. (2), and Eq. (2) can be transformed into the following equation.

$$Z_i = -D/\left(\frac{A(x_i - C_x)}{f_x} + \frac{B(y_i - C_y)}{f_y} + C\right) \tag{5}$$

Set

$$\Delta_1 = \frac{(x_i - C_x)}{f_x}, \Delta_2 = \frac{(y_i - C_y)}{f_y} \tag{6}$$

Put Eq. (6) into Eq. (4), (5), and we have:

$$\begin{aligned} X_i &= -D/(A\Delta_1 + B\Delta_2 + C) \\ Y_i &= -D\Delta_1/(A\Delta_1 + B\Delta_2 + C) \\ Z_i &= -D\Delta_2/(A\Delta_1 + B\Delta_2 + C) \end{aligned} \tag{7}$$

The three-dimensional coordinates of two feature points at two different times can be obtained by using Eq. (7). A, B, C, D are used to express its corresponding three-dimensional coordinates P_i, P_i', P_j, P_j'. Put these coordinates into Eq. (3), and we get a nonlinear quadratic equation of four variables. Select $n \geq 4$ matched point pairs, and n groups of quadratic equation of four variables can be formed. From these equations, a feasible solution exists in theory.

Because the model has the property that the distance between the camera and the target plane is independent(Chapter 2.4), $n \geq 3$ point pairs $p_i(x_i, y_i)$, $p_i'(x_i', y_i')$, $i = 1, ...n$ are sufficient to estimate the model.

2.4. **Model properties description.** (1)The distance between the camera and the target plane is independent.

The target motion plane, which meets the requirement of the model, is constructed in parallel with the target motion plane of ground truth. In other words, the parameters of the model are independent in terms of the distance between the camera and the target plane. The proof is given as follows:

There is a target plane which is the result of the model, and it satisfies Eq. (2):
$AX_i + BY_i + CZ_i + D = 0$

Assume there is another target plane $A\overline{X_i} + B\overline{Y_i} + C\overline{Z_i} + \overline{D} = 0$, and it is parallel to the plane above.

From Eq. (7), we have
$\overline{X_i} = -\overline{D}/(A\Delta_1 + B\Delta_2 + C) = \alpha X_i$
$\overline{Y_i} = -\overline{D}\Delta_1/(A\Delta_1 + B\Delta_2 + C) = \alpha Y_i$
$\overline{Z_i} = -\overline{D}\Delta_2/(A\Delta_1 + B\Delta_2 + C) = \alpha Z_i$
$\alpha = \overline{D}/D$.

Then,

$$|\overline{P_iP_i'}| = \sqrt{\left(\overline{X_i} - \overline{X_i'}\right)^2 + \left(\overline{Y_i} - \overline{Y_i'}\right)^2 + \left(\overline{Z_i} - \overline{Z_i'}\right)^2} =$$
$$\alpha\sqrt{(X_i - X_i')^2 + (Y_i - Y_i')^2 + (Z_i - Z_i')^2} = \alpha|P_iP_i'|$$

Similarly,$|\overline{P_jP_j'}| = \alpha|P_jP_j'|$

If P_i, P_i', P_j, P_j' meets the Eq. (3) i.e.,$|P_iP_i'| = |P_jP_j'|$, we have $|\overline{P_iP_i'}| = |\overline{P_jP_j'}|$, which means $\overline{P_i}$, $\overline{P_i'}$, $\overline{P_j}$, $\overline{P_j'}$ also meets Eq. (3). This independent property has been proved.

From the model, the distance between the target plane and the camera is unavailable. Consequently, in this paper, we set the normal vector (A,B,C) of the target plane in the camera coordinate system as the relative pose between the camera and target.

(2)Selection of n matched point pairs

In this model, in order to get the correct solution, it is necessary to ensure that in the selected n groups of matched point pairs, there are at least 3 groups in which the vectors between the two points are in different directions.

Of course, the larger the n is, the more robustness the algorithm is, and the more accurate the solution is.

2.5. **Model parameters estimation.** The model is solved by Algorithm 1, which is explained in the following.

Algorithm 1: Model parameters estimation

Input: the abstracted projected points p_i, p_i' on image series, $i = 1, 2, , m$.
Output: the normal vector $(A,B,C)$of planar π_1.
1: Construct the parameter $D = 10$.
2: **while** $i \leqslant m$
4: Describe the (X,Y,Z)coordinates of the p_i, p_i' by the parameters A, B, C using Eq.(7).
5: $i = i + 1$
6: **end while**
7: Describe the fitness function $min(d)$ in Eq. (8) by the parameters A, B, C.
8: Use a genetic optimization algorithm to solve the optimal solution of the model, the optimized parameters are A, B, C.
9: **return** the normal vector (A, B, C) as the relative pose between camera and plane.

$$d = \frac{var(|p_ip_i'|))}{mean(|p_ip_i'|}, i = 1, ..., m \tag{8}$$

In the algorithm, the input comprises the projected points p_i, p_i' of the image series, with m frames, and the output is the normal vector (A, B, C) of the plane π_1.

In step 1: According to property (1), D is independent, so we construct the parameter $D = 10$. Thus, there are now only three parameters, and in theory $n \geqslant 3$ points matches (p_i, p_i', p_j, p_j') are sufficient to solve the model.

In steps 2 to 6: the 3D coordinates of the points in m frames are described by the parameters A, B, and C.

From steps 7 to 8: the parameters are solved by a genetic optimization algorithm and the optimized parameters are A, B, and C. In the camera's coordinates system, the distances between the feature points p_i, p_i' remain the same at different times. Thus, we design a special fitness function as $min(d)$: (See Eq.8)

In step 9: The normal vector (A, B, C) of the target plane in the camera's coordinates system is the relative pose between the camera and motion plane.

In addition, in order to ensure that the motion plane obtained from the model is not trapped into a local minimum near the origin, we add a constraint:

$$s = |\frac{D}{\sqrt{A^2 + B^2 + C^2}}| > \Phi,$$

where Φ is a constant that is not very small, e.g., Φ can be set to 1. This constraint means that the distance from the camera's optic center to the motion plane is larger than a constant.

3. **The Image Planar Rectification.** Based on steps 2 to 6 in algorithm 1, the 3D coordinates of the points on an object can be described by (A,B,C). Using the value of the calibrated normal vector (A,B,C), the 3D coordinates of the feature points in the camera's coordinates system are obtained. The coordinates system must be transformed in order to obtain the orthographic projected 2D trajectory.

We construct a target 3D coordinates system (u_x', u_y', u_z'), as shown in Fig. 3.

In Fig. 3, the camera's coordinates system has moving feature points $P_i(X_i, Y_i, Z_i)$, $i = 1, 2, \ldots$, and i is the time order. To construct this system, the original point is $P_1(X_1, Y_1, Z_1)$ and the coordinate axes are $u_x = \overline{P_1 P_1'}, u_y = (A, B, C), u_z = u_x \times u_y$, where "$\times$" represents the vector cross product. Then, the coordinate axis vector is normalized to (u_x', u_y', u_z').

During planar rectification, we transform the trajectory coordinates into this new coordinates system, and the components (u_x', u_z') become the orthographic projected 2D trajectory coordinates.

4. **View-Invariant Gesture Recognition.** Fig. 4 is the flow chart of view-invariant hand gesture recognition. The following steps are taken:

Step 1. Key frame extraction.

Take samples of M frames in gesture sequence.

Step 2. Skin color-based gesture segmentation and gesture trajectory extraction.

First, it is necessary to translate the image from the RGB color space to YCbCr color space, and then to translate the color format to YCb'Cr'space in which a threshold is set for gesture segmentation [4]. Denoising is based on the segmented gesture. We remove the noise from the image using the corrosion and expansion method to obtain the gesture's region.

In the gesture's region, we use the classic Harris feature point extraction method to obtain the gesture's feature points and use the classic Harris points matching algorithm to match the points of the adjacent frames[15]. The result is the matched feature point set $\{(p(t), p(t+1))_i\}_{i=1t=1}^{m_t N-1}$, in which $p(t)$ represents the feature point at time t, m_t is the number of matched feature points between t and $t+1$ time $(m_t \geqslant 2), N$ is the number of key frames in the video, and $(p(t), p(t+1))_i$ represents the matched feature points between t and $t+1$ time.

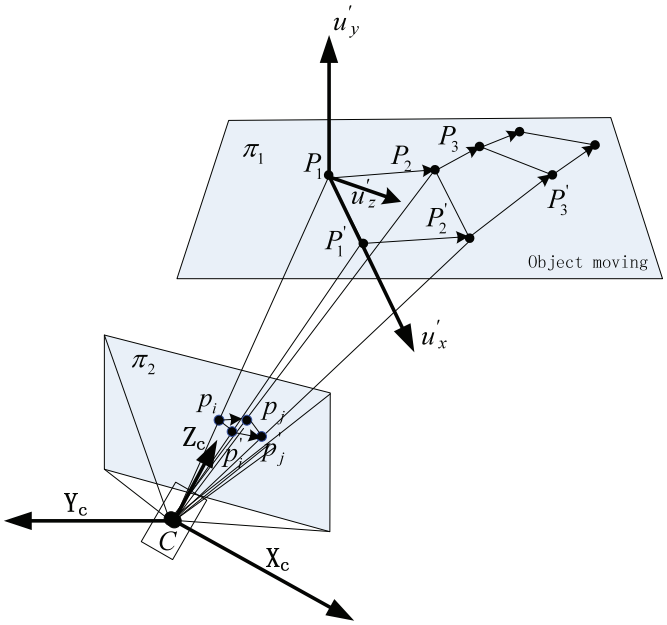

FIGURE 3. The constructed target 3D coordinates system (u'_x, u'_y, u'_z)

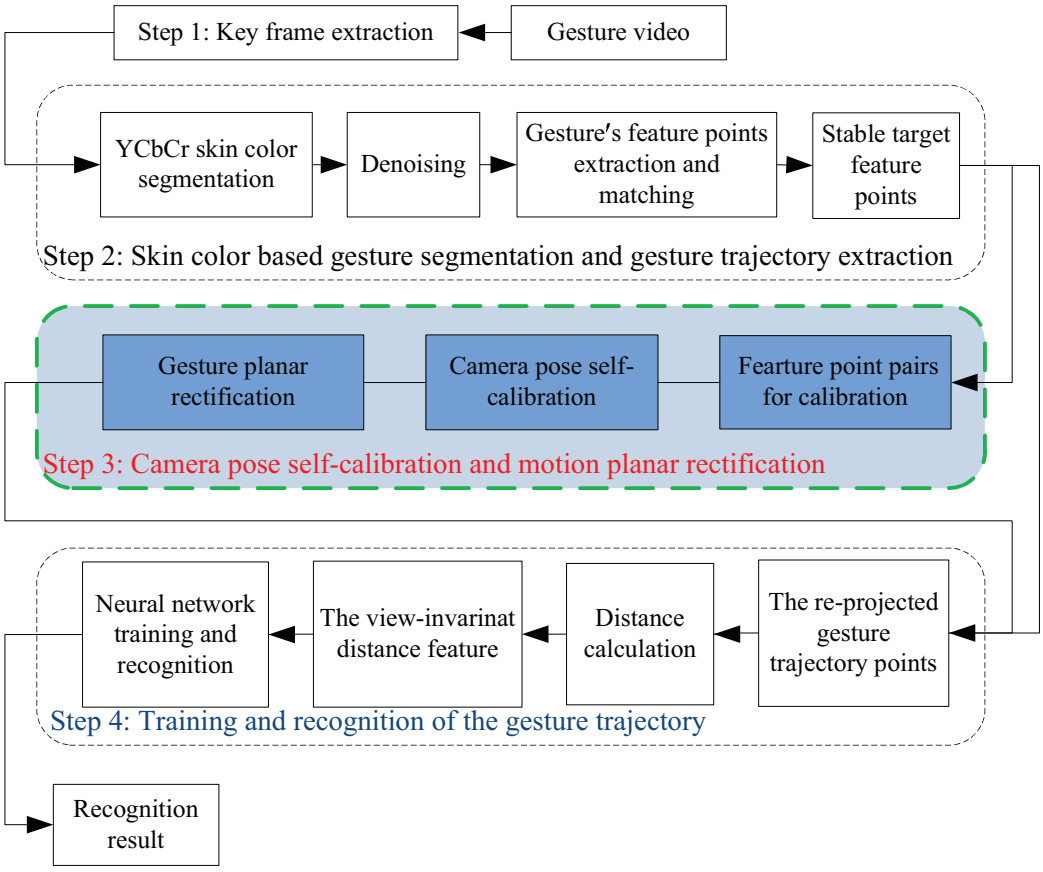

FIGURE 4. The view-invariant hand gesture trajectory recognition flow

Regarding stable target feature points, if there has special points that could be matched in every frame, these points are the stable target feature points, and they belong to the stable target feature points set $\{p(t)_i\}_{i=1t=1}^{m\ N}$, in which m is the number of the points in the set.

Step 3. Camera pose self-calibration model and motion planar calibration.

We need to establish the feature point pairs for calibration. In the stable target feature points set, the Euclidean distances between the feature points of the first frame are calculated to form the distance matrix. From the distance matrix, we select the maximum distance value as shown: $d(k', s') = max\{d(k, s)\}_{k=1s=1}^{m\ m}$.

Then k, s are the selected initial camera pose calibration feature points, in which m denotes the number of feature points in stable target feature points set, and $d(k, s)$ denotes the distance between the number k point and number s point in the first frame.

In initial feature point pairs for calibration, we calculate the vectors of the feature points for every moment. From the vectors, we select $n \geqslant 3$ vectors of different directions. (Different direction meant that the angles between vectors were $\geqslant 2^\circ$). We set $s(i)$ as the selected sequence number of the frame, $i = 1 \sim n$. Then its feature point pairs $p_{s(i)}(x_{s(i)}, y_{s(i)})$, $p'_{s(i)}(x'_{s(i)}, y'_{s(i)})$ are the camera pose calibration point pairs finally selected.

To self-calibration the camera pose, we use the proposed self-calibration model so that the relative pose (A, B, C) between the camera and the motion plane is calibrated. It is shown in Section 2.2 $\sim$ 2.5. Using the obtained relative pose (A, B, C), the gesture plane is rectified to get the orthographic projected 2D points, as shown in Section 3.

Step 4. Training and recognition of the gesture trajectory.

Connecting the re-projected gesture points in time order, the trajectories are obtained. We choose one re-projected trajectory to represent the trajectory of the gesture. We calculate the Euclidean distances between the trajectory points and the trajectory's central point on discrete time, and the result is the distance sequence $G = (r_1, r_2, \ldots, r_M)$. The distance sequence is normalized by $l_s = mod(r_s, 5) + 1$, to transform the distance to $1 \sim 5$ integer, and the result is the normalized distance feature $D = (l_1, l_2, \ldots, l_M)$. Using the normalized distance feature as the input, and its class as the output, we use the neural network for training and recognition. In training and recognition, the two layers of the neural networks are employed.

5. Experimental Results and Analysis.

For our experimental platform, the hardware environment was Inter(R) Core(TM) i3-2120, 4G, 3.30GHz. The software environment was the MATLAB R2013a in Windows 7.

5.1. The gesture drawing rectification experiment.

For the gesture rectification experiment, we made a video of a hand drawing a star, as shown in Fig. 5. The video was composed of 656 frames. During the process of drawing, the thumb fingertip and the index fingertip were used as the extracted camera calibration feature points [16]. The absolute distance between the two fingers remained the same throughout the drawing process. In Fig. 5, we display twenty images of the video. In Fig. 6, the extracted calibration feature points are marked in red circles.

FIGURE 5. The original star image

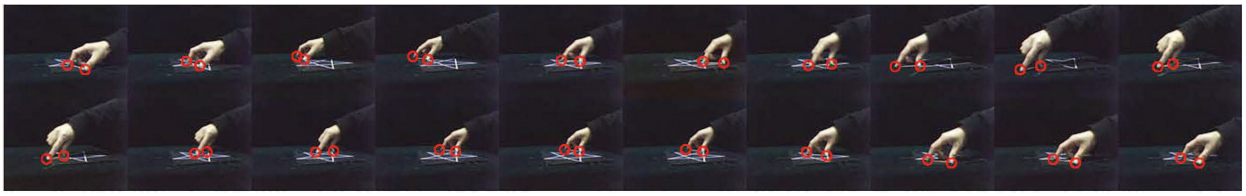

FIGURE 6. The extracted calibration feature points

In Fig. 7, the fingertip feature points were extracted to form two trajectories, marked in blue and brown respectively. To calibrate the camera pose, we selected the feature points of the frame sequence numbers 251, 51, 158, 51, 358, 501, 502, and 2, as shown in Fig. 7. The coordinate system in Fig. 7 is the image pixel coordinate system.

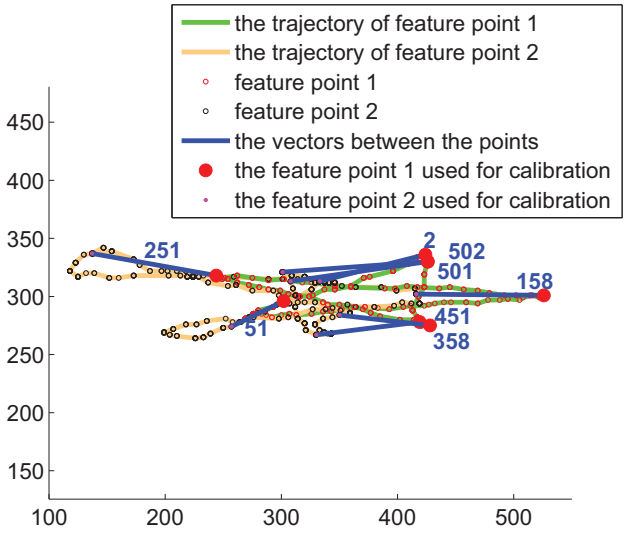

FIGURE 7. The calibration feature points and the trajectories

To translate the eight vectors to the same starting position, the result is in Fig. 8. Obviously the directions of the vectors are different, so it is easy to get the solution of the model.

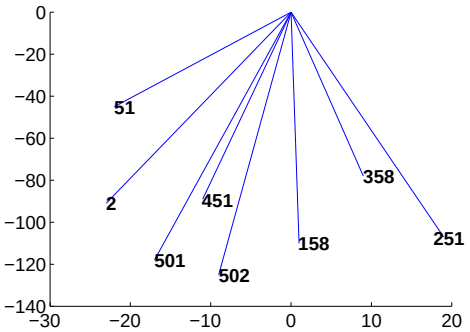

FIGURE 8. The vectors between the calibration feature points with the same starting point

The proposed model was used for camera pose calibration and trajectory planar rectification. The rectified trajectory is displayed in Fig. 9. The marked number is the frame

sequence number of the calibration feature points. Obviously, after the planar rectification, the trajectory of feature point 1 (thumb fingertip) is similar to the original star image. We re-projected the original background image, and the result is in Fig. 10.

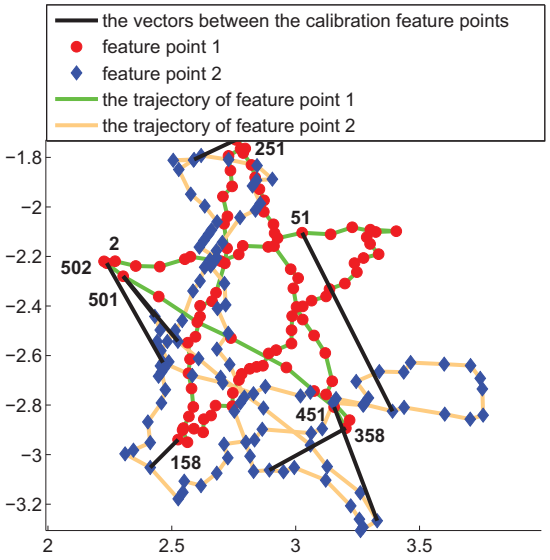

FIGURE 9. The result of planar rectification

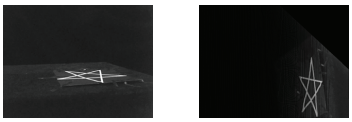

FIGURE 10. The original image and the rectified background image

Time efficiency analysis: The time efficiency analysis for the gesture drawing experiment, shown in Table 1, reveals that the feature point extraction time is the longest. To satisfy the gesture drawing experiment, every frame's feature points were extracted from the video in order to save the details of the drawing. The video has 656 frames in total, so it costs more time. In the camera calibration's application, it is necessary to select and match the feature points from only $n \geqslant 3$ key frames, so the time cost of the feature extraction can be reduced to less than 10s, and the overall runtime can be controlled in $30 \sim 40$ s. At the same time, it is necessary to improve the algorithm from the aspect of feature extraction.

TABLE 1. Time efficiency analysis

Frames	656
Time of feature points extraction(s)	1021.1
Time of feature points matching(s)	10.06628
Time of stable feature points extraction(s)	< 1
Time needed for model solving and planar Rectification(s)	17.707101
Overall runtime (s)	1049

Error analysis: In the gesture drawing experiment, the calibration of the camera pose was based on the fingertip feature point extraction. However, the accuracy of the fingertip

feature point extraction was not high enough to meet the high accuracy requirements for camera pose calibration. Furthermore, because the hand gesture did not involve a rigid object, it was not natural for the distance between the two fingertips to remain unchanged, which may cause error. In addition, we used only 8 feature point pairs to calibrate. More feature point pairs from more frames should be extracted to increase the robustness and accuracy of the model.

5.2. **Gesture recognition experiment.** In the experiment concerning gesture recognition, the camera's pose was the same as described in Section 5.1 of this paper. The dynamic gesture data set in the experiment consisted of the dynamic gestures of 0, 1, 2, 3, 7. The experimenter repeated 11 times for each gesture, and 55 hand gestures were obtained. Each gesture lasted about 2∼10 s (frame rate $\delta = 20fps$). In order to simulate the gesture recognition applications in near distance, the experimenter was 40∼80 cm away from the camera. The size of the image is 320×240 with 24bit true color. The average sampling number of the key frame is $M = 17$. The trajectory of the frontal shot was used for network training.

(1) The gesture trajectory

The center point of the hand gesture [4] was used as the feature point of the gesture trajectory. We obtained five kinds of trajectories, as shown in Fig. 11. Obviously, the trajectories are affected by perspective.

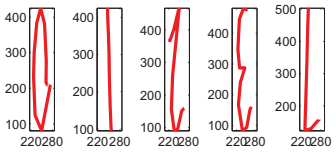

FIGURE 11. The samples of 5 projected trajectories

Using the proposed model, the gesture trajectories were rectified to get the orthographic projected gesture trajectories. Fig. 12 displays the rectified trajectories that correspond with the trajectories in Fig. 11. The rectified trajectories reflect the true trajectories more closely, but the coordinate system has changed.

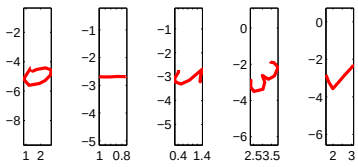

FIGURE 12. The samples of 5 rectified trajectories

(2) The gesture trajectory's distance feature

Fig. 13 represents the gesture trajectory's distance features in the experiment. From left to right, the graphs show the distance features of 0, 1, 2, 3, and 7 respectively. Based on Fig. 13, we can see that the curves for the rectified trajectory's distance features most closely resemble those of the frontal shot.

(3) The performance analysis of gesture recognition

For our research, Table 2 presents the rectified gesture trajectory recognition efficiency and its comparisons. The average recognition rate reaches 98%. Compared with the recognition rate of the original trajectory, the recognition rate of the rectified trajectory increases by 22%.

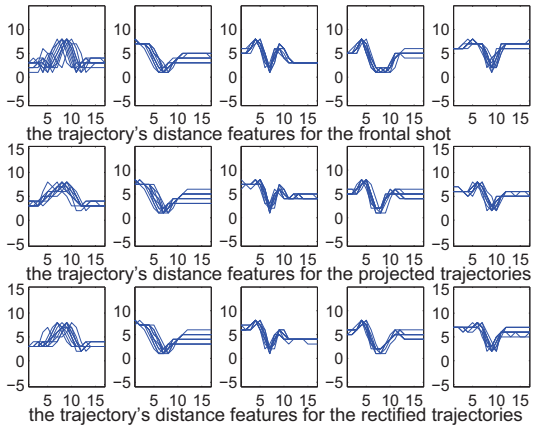

FIGURE 13. The gesture trajectory's distance features

TABLE 2. Recognition efficiency and its comparisons

Method	Index	0	1	2	3	7	Total
Use neural network to recognize the original data	Recognized number	9	11	11	8	3	42
	Recognition rate	0.82	1	1	0.73	0.27	0.76
Use neural network to recognize the rectified data	Recognized number	11	10	11	11	11	54
	Recognition rate	1	0.91	1	1	1	0.98
Ref. [3]	Recognition rate	0.9	1	0.83	0.83	1	0.91
Ref. [4]	Recognition rate	0.94	0.92	0.96	0.9	0.98	0.94
Ref. [6]	Recognition rate	0.96	0.96	0.93	0.93	0.9	0.94

Table 2 also presents the recognition results of [3], [4], and [6] for the same kind of gestures. In Table 2, after the planar rectification, the recognition rate of our proposed method is higher than the rate achieved by the others. For the methods that are compared, the camera's frontal shot angle was operated manually, which would produce some errors, so the factors of perspective may have had an effect on the extracted trajectory features.

In their approach to utilizing monocular vision, Zhang et al. [3] used frontal shot images and a Hidden Markov Model (HMM) to achieve gesture recognition. For the research presented in this paper, we applied their feature extraction and processing methods. Dixit and Agrawal [4] extracted fingertip feature, and also used an HMM for recognition. Using Kinect depth camera, Ghaleb et al. [6] extracted a 3D trajectory to achieve gesture recognition.

In order to assess the performance of our proposed method, we compared its results with the performance of the methods used in Ref. [3], [4], [6], [17], [18], [19], and [20]. The outcomes are shown in Table 3.

In their approach to utilizing monocular vision, Bao et al. [17] extracted the Speeded Up Robust Features (SURF), and used data stream clustering based on the correlation analysis to recognize the gesture. Ren et al. [18] used dynamic space-time warping for gesture recognition. Kılıboz and *Güdükbay* [19] used a six-degrees-of-freedom magnetic 3D position tracker to extract an ordered sequence of directional movements in 2D. In yet another approach, Chen et al. [20] employed an infrared tracker and Wii Remote Plus (Wiimote) to construct the 6-DOF sequence for free degree decision tree recognition. In that way, the trajectory's sequence feature was accurate and view-invariant.

Running time comparison: According to Table 3, using an ordinary camera with a high frame rate and high image resolution, our proposed method has a running time of

TABLE 3. Performance of the proposed method and its comparisons

Method	This paper	Ref.[3]	Ref.[4]	Ref.[6]	Ref.[17]	Ref.[18]	Ref.[19]	Ref.[20]
Computer frequency	3.3 GHz	3.3GHz	-	-	2.2GHz	600MHz	2.2GHz	-
Frame rate(f/s)	20	25	-	15	$8 \sim 16$	10	-	-
Recognition rate(%)	98	87.67	91	92.5	84.6	91.7	73	98.1
Average sequence length (s)	3.95	5	-	-	3×5	1.2	-	1.95
Average sequence length (frame)	79.02	120×130	-	-	18×38	12	-	-
Image resolution	320×240	160×120	-	320×240	176×144	160×120	-	-
Running time (s)	0.89	0.46	-	-	1∼3	≥2.07	-	-
Number of videos	55	300	160	600	1040	120	1998	5720
Number of classes	5	10	16	10	26	12	11	26
Sampling rate (Hz)	-	-	-	-	-	-	60	60

0.89 s/sequence (84.99 framess). Comparing the running times alone, the time achieved by Ref. [3] is the fastest. However, the image resolution achieved by Ref. [3] is low, i.e., only one quarter of the resolution achieved by the method used in this paper. To achieve a better overall comparison, if we were to multiply the running time result of Ref. [3] by 4, the result would be 1.84, which is larger than this paper's 0.89. With this assessment, we can see that the proposed method runs fast, and it also meets the real-time requirement fully.

Recognition rate comparison: According to Table 3, the gesture recognition rate of out method is 98%. Comparing the recognition rates, the recognition rate achieved by Ref. [20] is slightly higher than the rate results of this paper. However, Ref. [20] did not use ordinary visual acquisition, but rather used an infrared tracker and the Wii Remote Plus (Wiimote) to construct the 6-DOF sequence. In this way, in Ref. [20], the trajectory sequence's abstracted feature was accurate, and it was not affected by the visual angle and feature extraction.

6. **Conclusions.** In this paper, we offered a model of camera pose self-calibration, and we applied the model to view-invariant gesture trajectory recognition. Using this approach, we solved the problem of visual angle influence in trajectory recognition. In our experiments, we analyzed the effectiveness of the view-invariant gesture analysis method.

From the experiment and analysis, our study makes three significant contributions. (1) The method has high average recognition accuracy rate. In the field of gesture recognition, our method increased the trajectory recognition accuracy rate by 22%. (2) The method is fast. The average running time is 1.60s for one video, which can realize the real time requirement. (3) The method is extensible to other area. Our proposed method can be used universally in the field of trajectory analysis, and it can provide ideas for the pose calibration of industrial robots. This method is most suitable for application to a moving target that is a rigid body.

In order to increase the robustness and the accuracy of camera pose self-calibration model, in the process of calibration, more feature point pairs of multi-frames should be extracted.

Further research is planned to consider the view-invariant hand gesture analysis in the more complex conditions, like natural human computer gesture interaction.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61363078), the National Natural Science Foundation of China (No.

61362034), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 201700005).The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] T. H. Tsai, C. Y. Lin, Visual hand gesture segmentation using three-phase model tracking technique for real-time gesture interpretation system, *Journal of Information Hiding & Multimedia Signal Processing*, pp.122–134, 2012.

[2] L. Qin, Q. Hu, Q. Huang, and Q. Tian. Action recognition using trajectories of spatio-tempral feature points, *Chinese Journal of Computers*, vol. 37, no. 6, pp. 1281–1288, 2015.

[3] X. Zhang, H. Wang, Y. Tian, L. Peyrodie, and X. Wang. Model-free based neural network control with time-delay estimation for lower extremity exoskeleton, *Neurocomputing*, vol. 47, no. 10, pp. 3172–3183, 2017.

[4] Q.Y. Zhang, L. Lv, and M.Y. Zhang. A Dynamic Gesture Trajectory Recognition Based on Key Frame Extraction and HMM, *International Journal of Signal Processing Image Processing & Pattern Recognition*, vol. 8, no. 6, pp. 91–106, 2015.

[5] V. Pitsikalis, A. Katsamanis, S. Theodorakis, and P. Maragos. Multimodal gesture recognition via multiple hypotheses rescoring, *Journal of Machine Learning Research*, vol.16, pp.255–284, 2015.

[6] F.F.M. Ghaleb, E.A. Youness, and M. Elmezain. Vision-Based Hand Gesture Spotting and Recognition Using CRF and SVM, *Journal of Software Engineering and Applications*, vol. 8, pp. 313–323, 2015.

[7] K. Huang, Y. Zhang, and T. Tan. A discriminative model of motion and cross ratio for view-invariant action recognition, *IEEE Transactions on Image Processing*, pp. 2187–2197, 2012.

[8] C. Zhang and Y. Tian. Histogram of 3D Facets: A depth descriptor for human action and hand gesture recognition, *Computer Vision and Image Understanding*, vol. 139, pp. 29–39, 2015.

[9] B. Li, Y. Dai, and M. He. A relaxation method to articulated trajectory reconstruction from monocular image sequence, in *IEEE China Summit & International Conference on Signal and Information Processing(ChinaSIP)*, pp. 389–393, 2014.

[10] H.S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D Reconstruction of a Moving Point from a Series of 2D Projections, in *European Conference on Computer Vision Conference on Computer Vision*, pp. 158–171, 2010.

[11] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, *IEEE Journal on Robotics & Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[12] Z. Zhang. A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.

[13] S. Thompson, D. Stoyanov, and C. Schneider. HandCeye calibration for rigid laparoscopes using an invariant point, *International Journal of Computer Assisted Radiology & Surgery*, vol. 11, no. 6, pp. 1071–1080, 2016.

[14] M.Y. Zhang, Q.Y. Zhang, H.X. Duan, and H.Y. Wei. View-invariant hand gesture planar trajectory recognition on monocular vision, *Journal of Information Hiding & Multimedia Signal Processing*, vol. 8, no. 1, pp. 76–85, 2017.

[15] S.Tang, Y. Miao. An image matching algorithm based on Harris corner, *Microcomputer & Its Applications*, vol. 32, no. 2, pp. 41-43, 2013.

[16] R. Hartanto, A. Susanto, and P. I. Santosa. Real time hand gesture movements tracking and recognizing system, *Electrical Power, Electronics, Communications, Controls, & Informatics International Seminar*, vol. 6, pp. 137–141, 2014.

[17] J.T. Bao, A.G. Song, and Y. Guo. Dynamic Hand Gesture Recognition Based on SURF Tracking, *ROBOT*, vol. 33. pp. 482–489, 2011.

[18] R. Hai. Spatio-Temporal Appearance Modeling and Recognition of Continuous Dynamic Hand Gestures, *Chinese Journal of Computers*, vol. 23, no. 8, pp. 824–828, 2000.

[19] N. Kılıboz and U. Güdükbay. A hand gesture recognition technique for humanCcomputer interaction, *Journal of Visual Communication & Image Representation*, vol. 28, pp. 97–104, 2015.

[20] M. Chen, G. Alregib, and B. Juang. Air-Writing RecognitionPart I: Modeling and Recognition of Characters, Words, and Connecting Motions, *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 403–413, 2016.