

Camera Pose Self-calibration-based View-invariant Trajectory Analysis with Monocular Vision



Mo-Yi Zhang, Qiu-Yu Zhang*, Hong-Xiang Duan, Hui-Yi Wei

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China
{zhangqylz, zhangmoyi_1985}@163.com

Received 22 August 2018; Revised 22 December 2018; Accepted 6 March 2019

Abstract. Detecting view-invariant trajectory features is a fundamental task in many human-computer interactions. Existing approaches often rely on motion models or behavioral models, which have strict constraints, and the accuracy of detection cannot satisfy the requirements. Thus, we propose a new practical method for extracting the view-invariant features of the trajectory. The most innovative feature of our proposed approach is a camera pose self-calibration model. By extracting sparse feature points from some video frames and matching them across frames, our model can compute the camera pose related to the motion plane. In experiments, we analyzed our method in terms of the correctness, effectiveness, computational efficiency, robustness, and error obtained. In addition, the proposed method and new calibration model obtained greater accuracy at trajectory analysis compared with a previously proposed motion estimation method and a calibration method. The new method obtained satisfactory performance in a gesture drawing experiment. The proposed model can be applied widely in view-invariant trajectory analysis.

Keywords: camera pose self-calibration, gesture interaction, monocular vision, motion trajectory analysis, structure from motion, view-invariant

1 Introduction

Understanding trajectories plays an important role in various computer vision applications. In trajectory understanding, monocular vision has advantages in terms of simplicity and flexibility, and it is widely used in human-computer interaction (HCI) [1]. In general, we need to avoid the influence of perspective, so obtaining the view-invariant trajectory is a fundamental task in many HCI problems.

View-invariant trajectory analysis in monocular vision uses a projected image or series of images. Previous studies that considered this type of analysis are summarized in Table 1. In general, these studies employed a motion model-based method and a camera pose calibration-based method.

Table 1. Studies of view-invariant analysis using monocular vision

Motion model based	Special motion type	[2]	With a constant velocity constraint
		[3]	Using a discrete cosine transform basis
		[4]	Solving an articulated trajectory reconstruction problem
	Invariant feature extraction	[5]	CNN, multi-view learning
[6]		A multi-domain and multi-task learning	
[7]		Learning a view-invariant classifier	
Pose calibration based	Active based	[8]	Using a squared gesture feature for calibration
		[9]	A calibration method based on multi-rectangle
	Self-calibration	[10]	A new checkerboard with included angle 120°
		[11]	Based on planar feature points
		[12]	Based on circular features
	[13]	based on 1D Homography with 1D objects	

* Corresponding Author

In motion model-based method, special motion type is targeted specific situations and are inappropriate for random dynamic gesture trajectory analysis in HCI. Xu et al. [2] reconstructed trajectories based on the assumption that velocity is constant. Wang et al. introduced a discrete cosine transform basis to reconstruct a non-rigid structure [3]. Anton et al. [4] proposed a relaxation-based objective function that utilizes smoothness and geometric constraints, where they considered articulated trajectory reconstruction as a nonlinear optimization problem. Some other studies have imposed invariant feature extraction method and have used the multi-view to learn a classifier. In particular, Li et al. [5] proposed a view-invariant convolutional neural network (CNN) model for scene understanding in disaster scenarios. In [6], a multi-domain and multi-task learning (MDMTL) method were used to learn domain-invariant information. Zhang et al. [7] learned a view-invariance transfer dictionary and a view-invariant classifier.

Camera pose calibration-based methods is more flexible. Zhang et al. [8] proposed a camera pose calibration method based on the feature of a square trajectory. Lu et al. [9] put forward a calibration method based on multi-rectangle, which constructs several rectangles with the mark lines of the traffic road. Wu et al. [10] designed a new checkerboard with included angle 120° . The camera parameters and the rotational axis can be calibrated simultaneously. In these studies, static feature information in images was used for calibration, but not all of the interactions had static features. In self-pose calibration methods, the analyzed image sequence is used to calibrate the camera pose. In a different approach, Zhang et al. [11] introduced a self-calibration method for a monocular vision system based on planar points. Thompson et al. [12] proposed a simple and clinically feasible calibration method based on a single invariant point. Lv et al. [13] calibrated the camera pose based on 1D Homography with 1D objects, rotating around a fixed point and moving on a plane. These method are more flexible but restricted to special conditions.

From above, the existing approaches have strict constraints, or the accuracy of detection is not high, so they cannot satisfy the requirements in HCI application. In this study, in case of 2D motion, a camera pose self-calibration model is proposed to compute the camera pose in relative to the moving plane. Then, the trajectory can be re-projected into orthographic projected 2D trajectory which is view-invariant.

Compared with the previous research, the proposed study makes three significant contributions. (1) The proposed method facilitates view-invariant gesture trajectory analysis and this method can also be used to calibrate the camera's pose. (2) A camera pose self-calibration model is built using a series of motion images, whereas previous methods used static calibration images. (3) Our approach requires weaker constraints and it is more practical.

The remainder of this paper is organized as follows. In Section 2, we introduce the framework for view-invariant trajectory analysis as well as the feature extraction, the camera pose self-calibration model and the planar rectification. In Section 3, we present the results of experiments conducted using simulated data and real-world examples. Finally, we give our conclusions in Section 4.

2 Proposed Method

2.1 Overview

Fig. 1 shows the framework employed for gesture trajectory analysis. A three-level approach for view-invariant trajectory analysis is proposed in this paper.

Phase 1 is the feature extraction. In this phase, we extract the feature points from the projected image series. These feature points are used to compose the trajectories analyzed. Phase 2 is the camera pose self-calibration. In this phase, using the feature points extracted, the camera pose self-calibration model is built to calibrate the relative pose of the camera in the scene. Phase 3 is the planar rectification. In this phase, the trajectory can be rectified to get the view-invariant trajectory. In the approach, the most innovative point is the camera pose self-calibration model. The three phases are described in specify in following subsections.

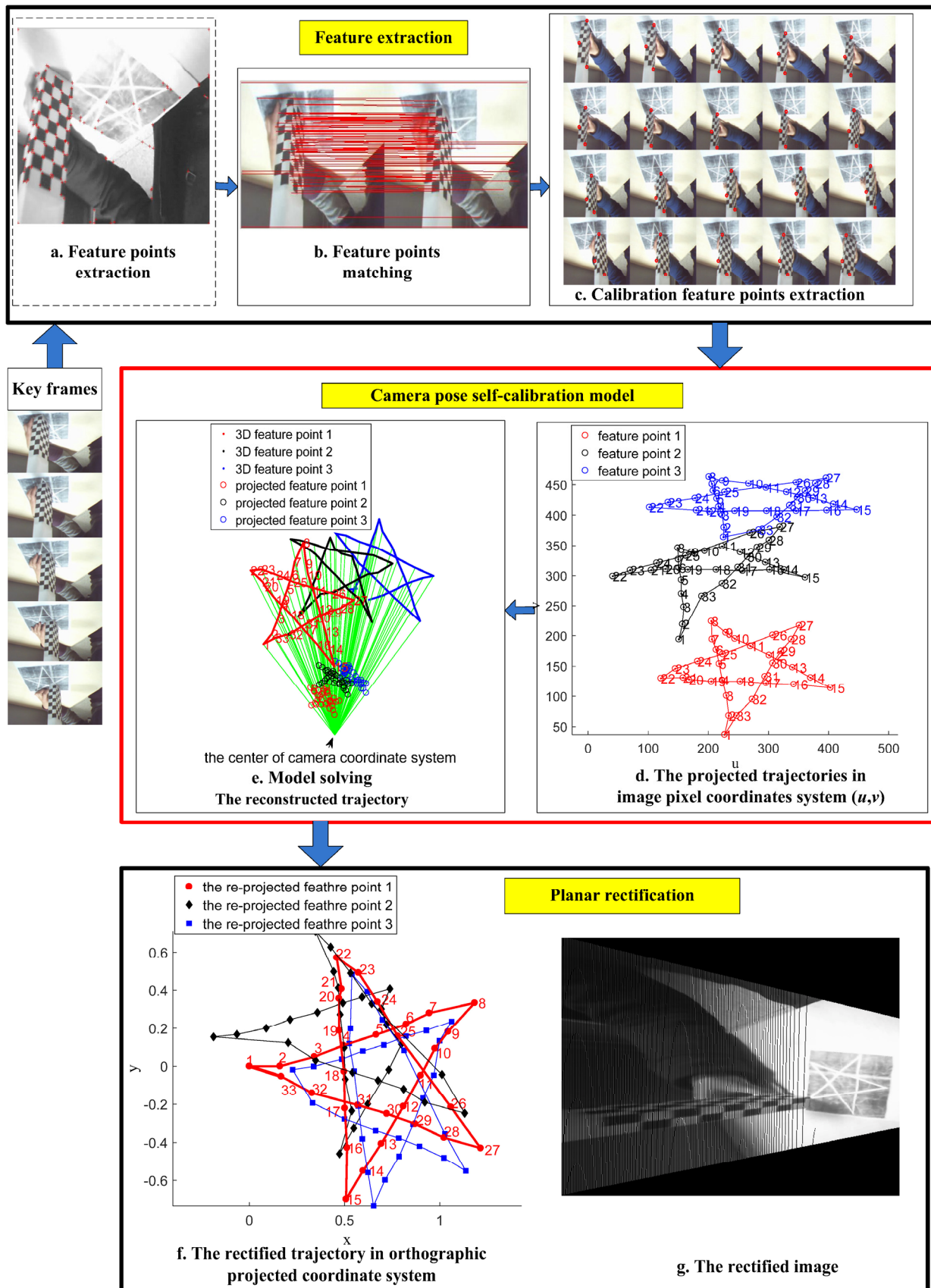


Fig. 1. Framework for view-invariant trajectory analysis

2.2 Feature extraction

In feature extraction phase, we extract the key frames as input, extract the feature points from the key frames, match the points across the key frames, and extract the trajectories for analyzing. The description is in the following:

Key frame extraction. take samples of M frames on an average in a gesture sequence.

Feature point extraction. classic Harris feature point extraction method [14] is used to obtain feature points.

Feature point matching. we need to match the feature points across the adjacent key frames. We used the classic Harris points matching algorithm [14] here.

Calibration feature point extraction. the motion region is determined based on the differences between adjacent frames. In motion region, the points that can be matched in every frame comprise the stable target feature points set $\{p(t)_i\}_{i=1}^m$, where m is the number of points in the set, and t is the serial number of key frame. In frame 1, the stable target feature point set is $\{p(t)_i\}_{i=1}^m$. From it, three points $p(1)_1, p(1)_2, p(1)_3$ are selected as the calibration feature points. We select the points with the maximum values based on the summed distances.

The projected trajectories. the connections of the calibration feature points in order constitute the trajectories extracted.

2.3 Camera Pose Self-calibration Model

2.3.1 Problem Description

We briefly describe the camera pose self-calibration problem shown in Fig. 2.

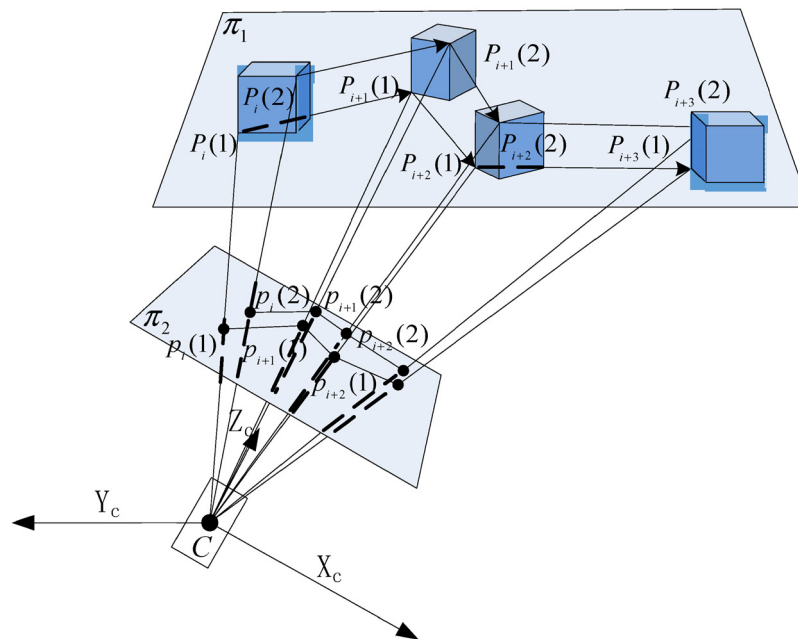


Fig. 2. Camera pose self-calibration model

In the camera's coordinates system, the target moves along the plane π_1 with the normal vector (A, B, C) . The problem involves computing (A, B, C) as the relative pose between the camera and motion plane based on the projected image series on plane π_2 .

Based on Fig. 2, we now explain the problem in detail. The camera's coordinates system comprises O, X_c, Y_c, Z_c . In the camera's coordinates system, the camera is set at the origin point with the optical axis as the z axis. The target motion plane is π_1 and the image plane is π_2 . The target is the rigid body. The target has moving feature points $P_i(s), s = 1, 2, \dots$. And the corresponding projected points $p_i(s), s = 1, 2, \dots$ are on the image plane π_2 , where i indicates that the moving object is at time i and s denotes the serial

numbers of the points. In this problem, based on the image series, the abstracted projected points $p_i(s)$ comprise the projected trajectory. Then, using the camera pose self-calibration model, we can calibrate the camera's pose spontaneously. Last, we can rectify the projected trajectory to the orthographic projected trajectory, which is view-invariant.

2.3.2 Camera Imaging Model

A diagram of the camera imaging model is shown in Fig. 3.

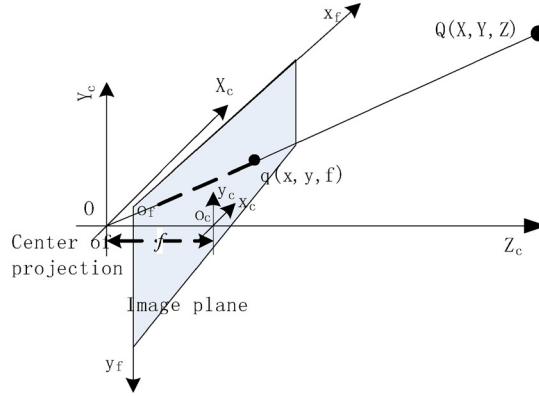


Fig. 3. Camera imaging model

As shown in Fig. 3, the image physical coordinates system comprises o_c, x_c, y_c , the image coordinates system comprises o_f, x_f, y_f , and the camera's coordinates system comprises O, X_c, Y_c, Z_c . According to the camera pinhole imaging theory, in the camera's coordinates system, the target point $Q(X, Y, Z)$, the image point $q(x, y, f)$, and the optical center of the camera O are in a line, which can be expressed as follows.

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

Actually, the principal point of the image might not be at the center of the projected image. Therefore, the deviations of the optical axes C_x and C_y are introduced in the camera imaging model. Similarly, a single pixel in the low-cost imaging instrument is rectangular rather than square, so two focal length parameters denoted as f_x and f_y are introduced, and the focal length is measured in pixels. The camera model employs the following function.

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

We can translate this function into matrix form, as follows.

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

2.3.3 Camera Pose self-Calibration Model.

The following explanation is based on Fig. 2.

The coordinates of $p_i(s)$ in the image physical coordinate system are $(x_i(s), y_i(s))$. The corresponding points on the target in the camera's coordinates system are $P_i(s) = (X_i(s), Y_i(s), Z_i(s))$.

(1) In the camera's coordinates system, the moving feature points satisfy the camera imaging model according to Eq. (1).

(2) The target moves along a plane, so the vector $\overline{P_i(s)P_j(s)}$ is vertical to the plane's normal vector (A, B, C) , which satisfies Eq. (2).

$$A(X_j(s) - X_i(s)) + B(Y_j(s) - Y_i(s)) + C(Z_j(s) - Z_i(s)) = 0. \tag{2}$$

(3) The target moves along a plane, so the angles between the vector $\overline{P_i(s)P_j(r)}$ and the normal vector (A, B, C) remain the same when i varies, which satisfies Eq. (3).

$$\cos \langle \overline{P_i(s)P_j(r)}, (A, B, C) \rangle = \cos \langle \overline{P_j(s)P_i(r)}, (A, B, C) \rangle. \tag{3}$$

(4) The distances between feature points $P_i(s), P_j(r)$ remain the same at different times, which satisfies Eq. (4).

$$|\overline{P_i(s)P_j(r)}| = |\overline{P_j(s)P_i(r)}|. \tag{4}$$

The mathematical model comprises Eq. (1), Eq. (2), Eq. (3), and Eq. (4).

2.3.4 Model Theoretic Solution.

The model is analyzed in the following steps.

(1) By substituting Eq. (1) into Eq. (2), we have

$$\Delta_{i/j}(s) = \frac{Z_i(s)}{Z_j(s)} = \frac{[A(x_j(s) - C_x) / f_x + B(y_j(s) - C_y) / f_y + C]}{[A(x_i(s) - C_x) / f_x + B(y_i(s) - C_y) / f_y + C]}. \tag{5}$$

(2) From Eq. (3), we have

$$\frac{\overline{P_i(s)P_j(r)} \cdot \overline{(A, B, C)}}{|\overline{P_i(s)P_j(r)}| \cdot \|\overline{(A, B, C)}\|} = \frac{\overline{P_j(s)P_i(r)} \cdot \overline{(A, B, C)}}{|\overline{P_j(s)P_i(r)}| \cdot \|\overline{(A, B, C)}\|}. \tag{6}$$

“ \cdot ” is the dot product, and “ $|\cdot|$ ” is the length of the vector.

(3) From Eq. (4), Eq. (6) can be transformed into

$$\overline{P_i(s)P_j(r)} \cdot \overline{(A, B, C)} = \overline{P_j(s)P_i(r)} \cdot \overline{(A, B, C)}. \tag{7}$$

(4) By substituting Eq. (1) and Eq. (5) into Eq. (7), we have

$$\begin{aligned} \Delta_j(s/r) = Z_j(s) / Z_j(r) = \\ C[1 + \Delta_{i/j}(s)] + \frac{A(x_j(s) - C_x + \Delta_{i/j}(s)x_i(s) - \Delta_{i/j}(s)C_x)}{f_x} + \frac{B(y_j(s) - C_y + \Delta_{i/j}(s)y_i(s) - \Delta_{i/j}(s)C_y)}{f_y} \\ C[1 + \Delta_{i/j}(r)] + \frac{A(x_j(r) - C_x + \Delta_{i/j}(r)x_i(r) - \Delta_{i/j}(r)C_x)}{f_x} + \frac{B(y_j(r) - C_y + \Delta_{i/j}(r)y_i(r) - \Delta_{i/j}(r)C_y)}{f_y} \end{aligned} \tag{8}$$

(5) From Eq. (5) and Eq. (8), we obtain proportional relationships between $Z_i(s), Z_i(r), Z_j(s)$, and $Z_j(r)$ as follows.

$$\begin{aligned} Z_i(s) &= Z_i(s) \\ Z_j(s) &= \Delta_{i/j}(s)Z_i(s) \\ Z_i(r) &= \frac{1}{\Delta_i(s/r)}Z_i(s) \\ Z_j(r) &= \frac{\Delta_{j/i}(r)}{\Delta_i(s/r)}Z_i(s) \end{aligned} \tag{9}$$

(6) By substituting Eq. (9) into Eq. (1), the exact coordinates of $P_i(s), P_j(s), P_i(r)$, and $P_j(r)$ can be denoted by the parameters A, B, C , and $Z_i(s)$, respectively. After entering the coordinates of $P_i(s), P_j(s), P_i(r)$, and $P_j(r)$ into Eq. (4), we then obtain an equation with four parameters.

(7) Select $n \geq 4$ point matches $(p_i(s), p_i(r), p_j(s), p_j(r))$ for the points $P(s)$ and $P(r)$. Set $Z_{i^*}(s)$ as one of the Z coordinates of $P(s)$. Then, n groups of quadratic equations for four variables $(A, B, C, Z_{i^*}(s))$ can be obtained. In theory, a feasible solution exists based on these equations. The 3D motion can be recovered from the solution and rectified to the orthographic projected trajectory.

2.3.5 Model Properties

If the parameters $(A, B, C, Z_{i^*}(s))$ are the result obtained by the model, then the other parameters $(A, B, C, Z'_{i^*}(s))$ also satisfy the model. Thus, the parameter $Z_{i^*}(s)$ is independent in the model.

The proof is given as follows.

For the parameters $(A, B, C, Z_{i^*}(s))$, $P_i(s) = (X_i(s), Y_i(s), Z_{i^*}(s))$.

For the parameters $(A, B, C, Z'_{i^*}(s))$, the parameters (A, B, C) remain unchanged so $\Delta_{i/j}(s), \Delta_i(s/r), \Delta_{j/i}(r)$ also remains unchanged. From Eq. (9), and Eq. (1), the recovered 3D point coordinates are as follows.

$$P'_i(s) = (X'_i(s), Y'_i(s), Z'_{i^*}(s)) = \frac{Z'_{i^*}(s)}{Z_{i^*}(s)} (X_i(s), Y_i(s), Z_i(s)) = \frac{Z'_{i^*}(s)}{Z_{i^*}(s)} P_i(s).$$

If $P_i(s)$ satisfies the constraints in Eq. (2), Eq. (3), and Eq. (4), it is easy to prove that $P'_i(s)$ also satisfies the constraints, which proves the property of independence.

Based on this property, we can only recover the 3D points up to a certain scale. Thus, in this study, we set the normal vector (A, B, C) of the target plane as the relative pose between the camera and motion plane.

2.3.6 Model Solving

From the property above, we need only to solve 3 parameters. Then, $n \geq 3$ point matches $(p_i(s), p_i(r), p_j(s), p_j(r))$ for the two points $P(s)$ and $P(r)$ are sufficient to estimate the model. Here, we abstract three feature points from the frames.

The model is solved by Algorithm 1, which is explained in the following. In algorithm, the input comprises the projected points $p_i(s), p_i(r)$, and $p_i(q)$ of image series, with m frames, and output is normal vector (A, B, C) .

Step 1: According to property (1), $Z_{i^*}(s)$ is independent, so we construct the parameter $Z_{i^*}(s) = Z_i(s) = 10$.

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

Steps 7 to 8. the parameters are solved by a genetic optimization algorithm and the optimized parameters are A, B , and C .

In camera's coordinates system, the distances between the feature points $P_i(s), P_i(r)$ remain same at different times. Thus, we design a special fitness function as $\min(d)$.

$$d = M(s, r) \times \text{var}(|P_i(s)P_i(r)|) + M(s, q) \times \text{var}(|P_i(s)P_i(q)|) + M(q, r) \times \text{var}(|P_i(q)P_i(r)|), \quad (10)$$

$$M(s, r) = \frac{\text{mean}(|P_i(s)P_i(r)|)}{\text{mean}(|P_i(s)P_{i+1}(s)|) \times \text{mean}(|P_i(r)P_{i+1}(r)|)},$$

where $|P_i(s)P_i(r)|$ denotes the sequence of distances between point $P_i(s)$ and point $P_i(r)$, var is the variance of the sequence, and mean is the mean of the sequence. The introduction of M function into the construction of this fitness function aims to avoid local minimum.

Step 9. After obtaining A, B , and C , 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 in a local minimum near the origin of the camera's coordinates system, we add a constraint.

$$s = \left| \frac{Z_{i^*}(s)}{\sqrt{A^2 + B^2 + C^2}} \right| > \Phi,$$

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.

Algorithm 1: Estimating the model parameters

Input: the projected points $p_i(s)$, $p_i(r)$, and $p_i(q)$ on the image series, m frames

Output: the normal vector (A, B, C) of plane π_1

1. Constructed parameter $Z_{i^*}(s) = Z_1(s) = 10$, $i^* = 1$
 2. **while** $i \leq m - 1$
 3. Describe $Z_i(s)$, $Z_i(r)$, $Z_i(q)$, $Z_{i+1}(s)$, $Z_{i+1}(r)$, $Z_{i+1}(q)$ by parameters A , B , and C using Eq. (9)
 4. Describe the (X, Y, Z) coordinates of $P_i(s)$, $P_i(r)$, $P_i(q)$, $P_{i+1}(s)$, $P_{i+1}(r)$, $P_{i+1}(q)$ by parameters A , B , and C using Eq. (1)
 5. $i = i + 1$
 6. **end while**
 7. Describe the fitness function $\min(d)$ in Eq. (10) by the parameters A , B , and C
 8. Using a genetic optimization algorithm to solve the optimal solution of the model, the optimized parameters are A , B , and C
 9. **return** the normal vector (A, B, C) as the relative pose between the camera and plane
-

2.4 Planar Rectification

Based on steps 2 to 6 in Algorithm 1, 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. But the coordinates system must be transformed in order to obtain the orthographic projected 2D trajectory. So, we construct a target 3D coordinates system (u'_x, u'_y, u'_z) , as shown in Fig. 4.

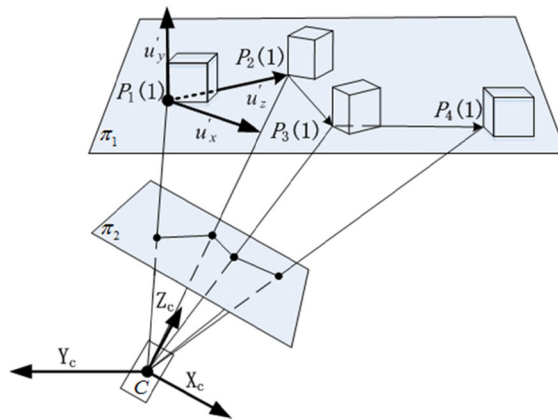


Fig. 4. The constructed target 3D coordinates system (u'_x, u'_y, u'_z)

3 Experimental Results and Performance Analysis

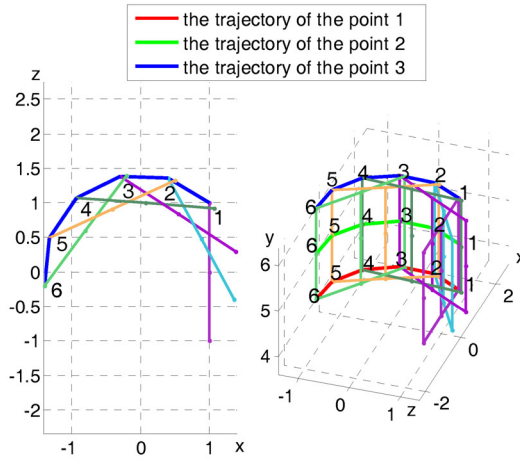
The experimental operating hardware platform comprised an Inter(R) Core(TM) i3-2120, 4 G, 3.30 GHz. The software environment was MATLAB R2013a in Windows 7.

3.1 Simulation Experiment

A simulation experiment was performed.

The moving target was a square shape and there were nine feature points in each frame. The movement time was 1~6 s, and we extracted one key frame in each moment. In the first key frame, the initial feature points were $(1, 4, 1)$, $(1, 5, 1)$, $(1, 6, 1)$, $(0, 4, 1)$, $(0, 5, 1)$, $(0, 6, 1)$, $(-1, 4, 1)$, $(-1, 5, 1)$, and $(-1, 6, 1)$.

The object's movements were defined as follows: the object moved along the plane parallel to $y = 0$, with rotation angle θ and translation (S_x, S_y) . When the object's moving parameters were $\theta = 0.5$, $S_x = 1$, and $S_y = 1$, the observed first point motion on the plane $y = 4$ and the object's motion in 3D space were as shown in Fig. 5(a) and Fig. 5(b), respectively.



(a) First point's motion on $y = 4$ (b) Object motion in 3D space

Fig. 5. Motions of the object in simulated 3D scene

The numbers in Fig. 5 are the time order of the feature points. From the first frame, we selected three points (1, 4, 1) (1, 5, 1), and (1, 6, 1) as feature points 1, 2, and 3, respectively. And we mainly analyzed the movements of these three points. These rules were also applied to obtain Fig. 6, Fig. 7, and Fig. 8.

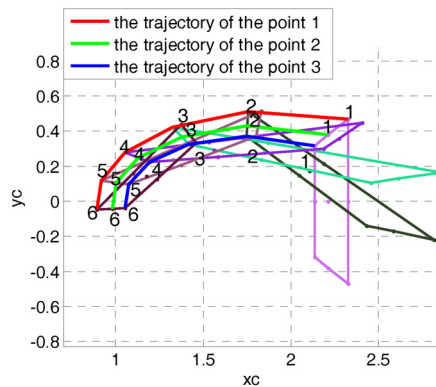


Fig. 6. Observed trajectories on the image plane in image physical coordinates system (x_c, y_c)

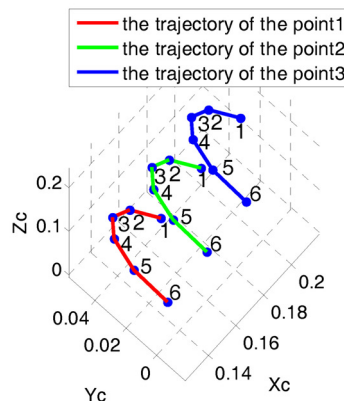


Fig. 7. Reconstructed 3D trajectories in camera's coordinates system (X_c, Y_c, Z_c)

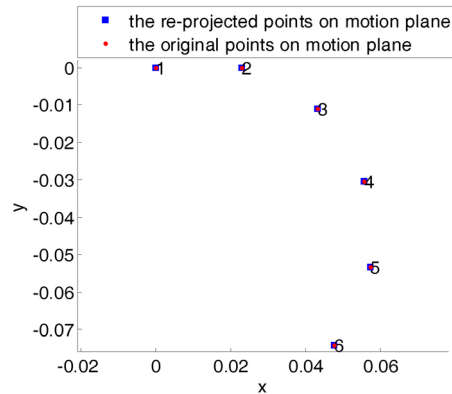


Fig. 8. Planar rectification of the first point and comparison with the original points

Next, we added a camera to the scene. The internal camera parameter matrix was $[600 \ 0 \ 300; 0 \ 600 \ 300; 0 \ 0 \ 1]$ and the extrinsic parameter matrix was $[0.5 \ 0 \ 0.87; 0 \ 1 \ 0; -0.87 \ 0 \ 0.5]$. The observed trajectories on the image plane are shown in Fig. 6. The camera pose self-calibration model was used to calibrate the relative pose of the camera and the motion plane. The results obtained for the relative camera pose and 3D trajectories are shown in Fig. 7. The orthographic projected 2D trajectory of the first feature point was obtained after planar rectification. Then it was compared with the original trajectory on the motion plane, as shown in Fig. 8.

According to Fig. 7, the reconstructed trajectories were similar to the real trajectories. After translation and scaling were used to align the first time feature points and the sixth time feature points of the two methods, we compare the two trajectories in Fig. 8. According to Fig. 8, the two trajectories were almost the same, which indirectly proves the correctness of the model.

Model solving efficiency analysis and comparison. Different genetic parameters were used to solve the model. The normal vectors obtained for the motion plane are compared in Table 2. Furthermore, the real normal vector of the motion plane was calculated based on the parameters constructed in the simulation experiment, as shown in Table 2. Then, a comparison was made using different parameters in genetic algorithm in Table 2. From comparison, the genetic generation number and the population size had a direct proportional relationship with the solving time and an inverse relationship with the fitness function. Thus, as the genetic generation number grows larger, the population size grows larger, and the result of the target planar normal vector is more accurate. Thereby it demonstrated that the proposed approach is adequate. Based on the results, we selected a generation number of 200 and population size of 200 for the following experiment because the model could be solved quickly and accurately with these settings.

Table 2. Model solving efficiency analysis and comparison

Comparison	Normal vector			Generation number	Population size	Fitness function	Time duration (s)
	x	y	z				
Real situation	1	0	0.5774	–	–	–	–
Proposed model	1	1.94×10^{-7}	0.5774	200	200	1.15×10^{-15}	9.467
	1	2.24×10^{-7}	0.5774	200	2000	1.28×10^{-15}	94.905
	1	-2.86×10^{-7}	0.5774	2000	200	8.75×10^{-16}	15.649

3.2 Experiment with a Calibration Board

The points on three calibration boards were used to simulate the feature points of a moving target in order to ensure a good comparison and analysis. In the experiment, we manually placed three boards in parallel. We designed virtual target moved parallel with the boards. We selected the matched points in images of the three boards to simulate the point projections of a virtual moving target. Next, we reconstructed the 3D information for the feature points.

We used the proposed model to calibrate the relative pose between the boards and the camera. As Zhang Zhengyou’s board calibration method has been recognized as the standard in camera calibration [15], we compared our results with those obtained by the calibration method proposed by Zhang Zhengyou [16].

Calibration using the camera pose self-calibration model. The calibration board comprised 9×7 squares. The projected image of the calibration board is shown in Fig. 9, and we extracted 6×9 , 5×5 , and 5×5 points.

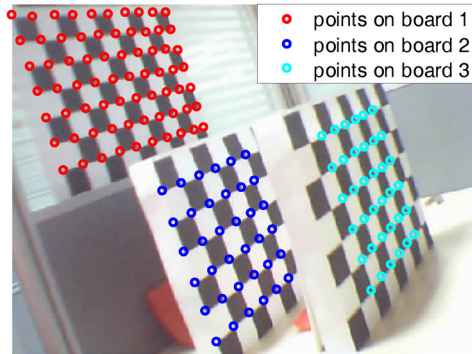


Fig. 9. Projected image of the calibration board

As shown in Fig. 10, we numbered the points on the three boards and we used the 36 feature points to form the motion trajectories. We selected all of the points [1: 36] on boards 2 and 3. The selected feature points on board 1 were [30: 35, 37: 42, 44: 49, 51: 56, 58: 63, 65: 70; 1: 36; 1: 36]. The three selected feature points were connected to form the black trajectories shown in Fig. 10.

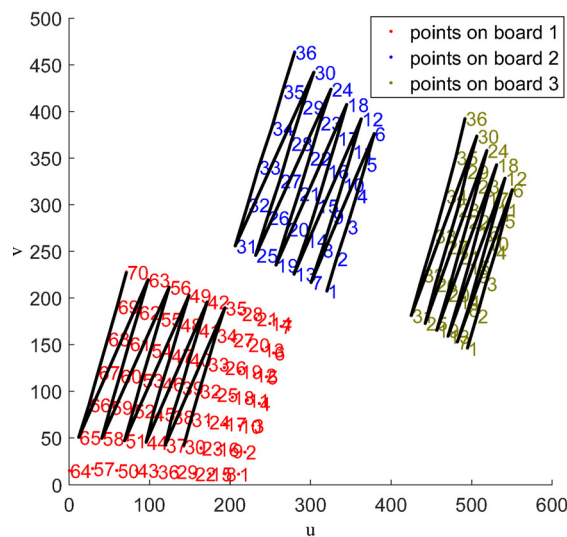


Fig. 10. Selected calibration feature point matches and trajectories in image pixel coordinates system (u, v)

We matched the three feature points on the three boards in order and we used the proposed model to calibrate the relative pose between the boards and the camera. Using the calibrated pose, the 3D coordinates of the feature points were reconstructed as shown in Fig. 11.

After image rectification, the projected image and the feature points in board 1 (as shown in Fig. 9) were rectified to obtain the orthographic projection, as shown in Fig. 12. From Fig. 12, it is demonstrated that a good result was obtained after image rectification.

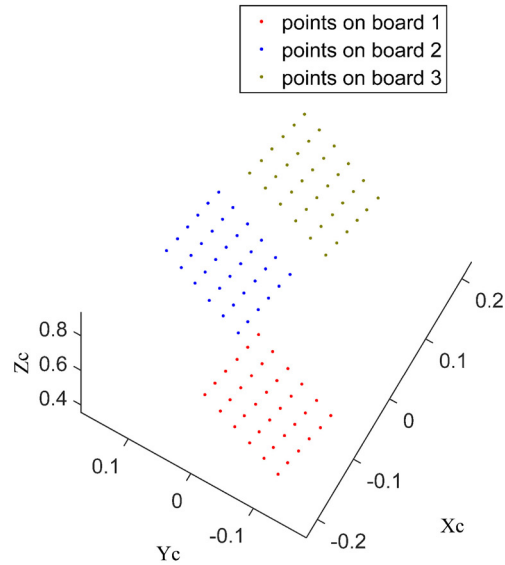


Fig. 11. Reconstructed 3D coordinates of the feature points in camera's coordinates system (X_c , Y_c , Z_c)

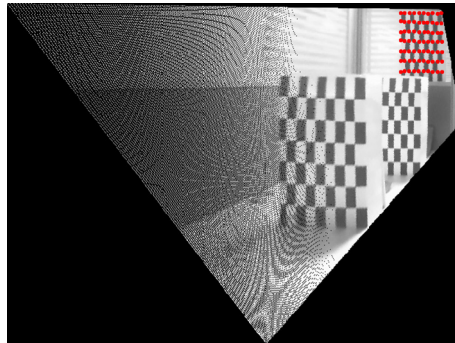


Fig. 12. Final rectified image

Comparison with Zhang Zhengyou's calibration method. We performed a quantitative comparison with the Zhang Zhengyou's calibration method. We obtained the target plane's normal vector using the method of Zhang Zhengyou and the proposed method. As Zhang Zhengyou's calibration method only uses the data of one board to calibrate, we obtained 3 calibrated results using the data from 3 boards separately. After normalizing the first dimension of the normal vectors to 1, we compared the normal vectors obtained by the two methods, as shown in Table 3.

Table 3. Normal vectors and errors

Comparison		The target planar normal vector			Straight edge length error	Diagonal length error
The proposed method		1	-0.3024	-0.7612	0.4381	0.4374
Zhang's method	Calibrated from board 1	1	-0.3371	-0.7726	0.4489	0.448
	Calibrated from board 2	1	-0.3117	-0.7678	0.4437	0.443
	Calibrated from board 3	1	-0.3073	-0.7576	0.4395	0.4388

On calibration boards, the straight edge lengths were equal and the diagonal lengths were equal for all the small squares. We designed two error indices for comparison: the straight edge length error and the diagonal length error. To obtain the straight edge length error, we computed all the straight edge lengths of the small squares as the straight edge length sequence A . And we designed the straight edge length error as $var(A)/mean(A)$, where var is the variance of the sequence and $mean$ is the mean of the sequence. To obtain the diagonal length error, we computed all the diagonal lengths of the small squares as the diagonal length sequence B . And the diagonal length error are designed as $var(B)/mean(B)$. Comparisons of the errors are shown in Table 3. The results show that the proposed method had the best performance

and it was better than the method of Zhang Zhengyou at trajectory motion analysis.

Robustness analysis. The robustness of the model was assessed in the following experiment.

Robustness of feature point extraction. To solve the model, we selected the second group of 36 feature points from board 1 as the input feature point matches and the results are shown in Table 4. According to Table 4, the error was even lower for the new input (line 2). In fact, the normal vectors and the errors changed slightly when we varied the selected feature point pairs, which demonstrated the robustness of the model.

Table 4. Results obtained after selecting different feature points pairs (36 frames)

Feature points selected on board 1	Target's planar normal vector		Straight edge length error	Diagonal length error
[16: 21, 23: 28, 30: 35, 37: 42, 44: 49, 51: 56; 1: 36; 1: 36]	1	-0.2977 -0.7541	0.4371	0.4365
[30: 35, 37: 42, 44: 49, 51: 56, 58: 63, 65: 70; 1: 36; 1: 36]	1	-0.3024 -0.7612	0.4381	0.4374

Feature point pairs required. We used the feature point matches for the numbers 1~ n , where $n = 4\sim 32$, and we compared the solutions and errors.

As shown in Fig. 13, we obtained the normal vectors of the motion plane with various values for n . The normal vector's x coordinates were transformed to one, so we only compared their y coordinates and z coordinates. The normal vector's x and y coordinates varied slightly as n increased with the proposed method. Thus, we demonstrated that the proposed model was stable based on this analysis.

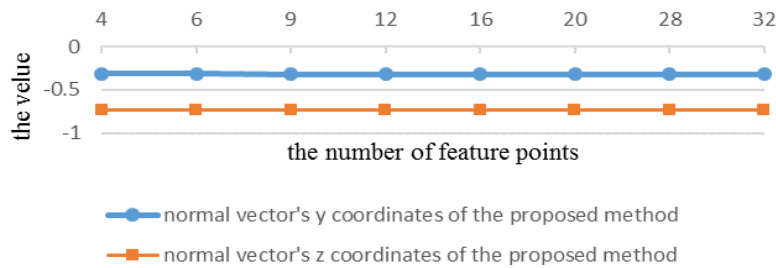


Fig. 13. Normal vectors of the motion plane

According to Fig. 14, we obtained the errors for the rectified boards using the proposed method with various values for n , and we compared them with the best results produced by the method of Zhang Zhengyou. As shown in Fig. 14, the straight edge error and diagonal error were lower with the proposed method than the Zhang's method. As n increased, the straight edge error and diagonal error all tended to be increased by the added noise. At last, the errors have a tendency to stable. This analysis demonstrates that the proposed model was robust and accurate.

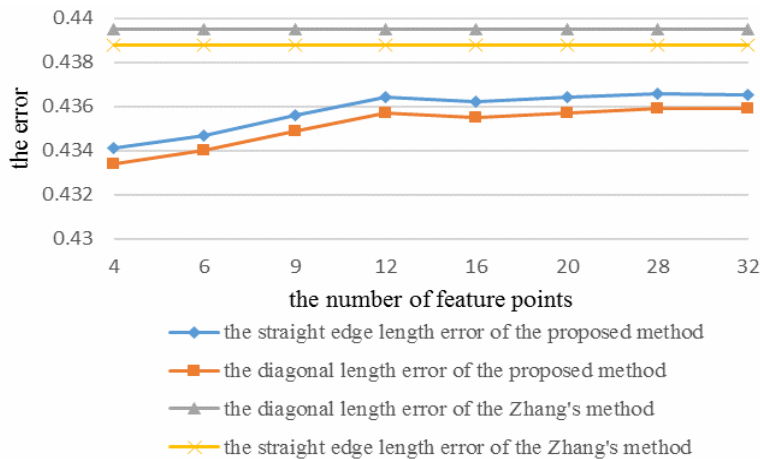


Fig. 14. Comparison of the errors in the rectified shape

3.3 Gesture Drawing Rectification Experiment

For the gesture drawing rectification experiment, we prepared a video of a hand drawing a star and the original star picture is shown in Fig. 15.



Fig. 15. Original star image

The video comprised 660 frames and during the drawing process, the projections of the three target feature points were extracted as camera calibration feature points. Figure 1c shows 20 frames from the video where the extracted calibration feature points are marked by red circles. According to Fig. 1(c), we can see that the object exhibits rotation in its planar motion.

The three feature points were extracted from 33 key frames to form three trajectories, as shown in Fig. 1(d). The numbers marked in the figure represent the sequence numbers of the feature points. The image pixel coordinates system is used in Fig. 1(d). The three trajectories were not the same because the object exhibited rotation during its motion.

The proposed model was used for camera pose calibration. Based on the calibrated camera pose, the reconstructed 3D feature points and the rectified trajectories of the feature points are shown in Fig. 1(e) and Fig. 1(f), respectively. The numbers mark the sequence numbers for the calibration feature points. Obviously, the trajectory of feature point 1 was similar to that of the original star image after planar rectification.

4 Conclusions

In this study, we proposed a model for camera pose self-calibration by using the feature points extracted from the trajectory where this model is then employed for view-invariant trajectory reconstruction. Using this approach, we solved the problem of view-invariance in trajectory recognition.

In overview, the framework for view-invariant trajectory analysis which has three levels was introduced. Then, the feature extraction, camera pose self-calibration model and planar rectification are described separately. In feature extraction phase, we extracted the feature points from the projected image series to compose the trajectory analyzed. In camera pose self-calibration model phase, using the feature points extracted, the camera pose self-calibration model is built to calibrate the relative pose of the camera in the scene. In planar rectification phase, the trajectory can be rectified to get the view-invariant trajectory.

In the approach, the most innovative part is the camera pose self-calibration model. In this part, starting from the problem description, combining the camera imaging model, the camera pose self-calibration model is built. Then, the model is analyzed in theory and solved by genetic optimization algorithm.

We did three kinds of experiment in total. In simulation experiment, our method can work perfectly in ideal condition. In calibration board experiment, our method can also work well with robustness, and it was even more accurate compared with the classic Zhang Zhengyou's calibration method. In gesture drawing rectification experiment, the trajectories are analyzed to get the view-invariant feature, which showed the applicable and time efficiency of the proposed method.

The main advantage of this model is that the camera pose can be calibrated using motion features, which is more suitable for target motion-related video applications. The proposed method can be used widely in the field of trajectory analysis and it may facilitate pose calibration for industrial robots.

This method is more suitable for analyzing the motion of a rigid body target. Rigid assistance is needed for gesture recognition. In future research, we plan to consider motion trajectory reconstruction for non-rigid objects.

Acknowledgements

This study was supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 201700005), and the Key Laboratory of Gansu Advanced Control for Industrial Processes (No. XJK201816). The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

- [1] S. Singh, C. Arora, C.V. Jawahar, Trajectory aligned features for first person action recognition, *Pattern Recognition* 62(2017) 45-55.
- [2] Y. Shang, J.B. Liu, T.T. Xie, A monocular pose measurement method of a translation-only one-dimensional object without scene information, *OPTIK* 125(15)(2014) 4051-4056.
- [3] Y.M. Wang, X.M. Yan, M.F. Jiang, J.B. Zheng, 3D non-rigid structure from motion based on sparse approximation in trajectory space, *International Journal of Robotics & Automation* 33(2)(2018) 111-117.
- [4] V.D.H. Anton, Y. Dai, B. Li, M. He, A relaxation method to articulated trajectory reconstruction from monocular image sequence, in: *Proc. IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, 2014.
- [5] L. Li, K. Ota, M. Dong, W. Borjigin, Eyes in the dark: distributed scene understanding for disaster management, *IEEE Transactions on Parallel & Distributed Systems* 28(12)(2017) 3458-3471.
- [6] A.A. Liu, N. Xu, W.Z. Nie, Y.T. Su, Y.D. Zhang, Multi-domain & multi-task learning for human action recognition, *IEEE Transactions on Image Processing* 28(2)(2018) 853-867.
- [7] J. Zhang, H.P.H. Shum, J. Han, L. Shao, Action recognition from arbitrary views using transferable dictionary learning, *IEEE Transactions on Image Processing* 27(10)(2018) 4709-4723.
- [8] M.Y. Zhang, Q.Y. Zhang, H.X. Duan, H.Y. Wei, View-invariant hand gesture planar trajectory recognition on monocular vision, *Journal of Information Hiding and Multimedia Signal Processing* 8(1)(2017) 76-85.
- [9] L. Lu, X. Lu, S. Ji, C. Tong, A traffic camera calibration method based on multi-rectangle, *IFIP Advances in Information & Communication Technology* 432(2016) 230-238.
- [10] X. Wu, D. Zhou, P. Wen, A MVS based automatic 3D model reconstruction system from turntable image sequence, in: *Proc. IEEE International Conference on Information & Automation*, 2016.
- [11] M.Y. Zhang, Q.Y. Zhang, H.X. Duan, H.Y. Chen, Self-calibration based view-invariant hand gesture trajectory analysis, *Journal of Information Hiding and Multimedia Signal Processing* 9(5)(2018) 1114-1127.
- [12] S. Thompson, D. Stoyanov, C. Schneider, Hand-eye calibration for rigid laparoscopes using an invariant point, *International Journal of Computer Assisted Radiology & Surgery* 11(6)(2016) 1071-1080.
- [13] Y. Lv, W. Liu, X. Xu, Methods based on 1D homography for camera calibration with 1D objects, *Applied Optics* 57(9)(2018) 2155-2164.
- [14] K.S. Changan, P.G. Chilveri, K.S. Changan, Stereo image feature matching using Harris corner detection algorithm, in: *Proc. International Conference on Automatic Control & Dynamic Optimization Techniques*, 2017.
- [15] L. Wang, F.Q. Duan, K. Lv, Camera calibration with one-dimensional objects based on the heteroscedastic error-in-variables model, *Acta Automatica Sinica* 40(2014) 643-652.
- [16] Z. Zhang, *Geometric Calibration*, in: K. Ikeuchi (Ed.), *Computer Vision*, Springer, Boston, MA, 2014.