

# Improved Direct Linear Calibration Based on a New Corner Detection Algorithm

Chunfang Wang\*, Ying Yang and Yuyu Gao

Experiment Center, LiRen College of Yanshan University, Qinghuangdao, Hebei, 066004, China

**Abstract:** This paper highlighted the technology of gray transformation to target the image captured using the camera, which converted a colored image into grayscale, filtered out the noise using the median, and extracted the image edge by using Canny operator. Because the extracting edges were not clear, it could not obtain the target pixel point accurately. For improving the accuracy and real-time feature extraction, corner feature points detection algorithm via randomized Hough transform based on spatial moment was proposed. To obtain accurate coordinates of the corners, the image was processed using matlab2011 software in the experiment. In order to achieve camera calibration rapidly and effectively, the linear calibration algorithm was improved. Using the least square method, the calculation process was simplified, and the calibration error was reduced. The experimental results show that the proposed algorithm was simple and effective, which improved the calibration accuracy, and verified its validity and feasibility.

**Keywords:** Image processing, Camera calibration, Random Hough transform, Spatial moments, Corner detection, Direct linear calibration.

## 1. INTRODUCTION

For analyzing the geometry of a picture and for determining the shapes and locations of objects, camera calibration is essential in visual serving system. Camera calibration is carried out according to the relationship between 3D geometric position of the target feature points in the spatial coordinate system and its corresponding pixel points in the image plane. The imaging model of a camera is established, and by analyzing the known coordinates and image coordinates of the feature points intrinsic and extrinsic parameters of the camera are determined. These parameters which cannot be obtained by measurement under most conditions must be obtained by camera calibration experiments. Accurate camera calibration can not only improve the accuracy of visual measurement, but also lay a good foundation for subsequent 3D reconstruction and stereo matching in visual serving [1].

Two-dimensional images are a manifestation of visual information containing rich pattern features. Which of these features are useful and how these features are extracted from the background, are key problems in visual serving research and in finding a necessary link for camera calibration. Therefore, image processing greatly affects the accuracy, stability and robustness of camera calibration, therefore, a rapid, accurate, and stable image feature extraction method is necessary. Image processing includes image preprocessing, image segmentation, feature extraction, etc. [2]. First the edges of the image are processed through gray level

transformation, followed by median filtering and edge detection using Canny operator. Following this, by using traditional corner detection method, corner features are extracted. But, some problems emerge, for instance, the effect is not ideal resulting in low precision of camera calibration, and the relative orientation and distance between the robot and the target cannot be calculated accurately, thus, it cannot meet the requirement of real-time system. In recent years, a large number of fast corner detection algorithms were studied. Li *et al.* detected the corners, by calculating the maximum peak of the local curvature function and setting the threshold, which stably detected the corners, and was effective in dealing with rotation, scale and noise [3]. Hou and Lin adjusted neighborhood parameter of non-maximum suppression and the threshold for Harris algorithm, and incorporated estimation formula with the number of the corners, to realize the automatic extraction algorithm of chessboard corners [4]. Lan and Zhang introduced a fast corner detection algorithm based on dual circular mask. It first detected whether a pixel is located in a flat area to avoid unnecessary calculation, and calculated the corner response function using the comparison function, which has good real-time performance, but is sensitive to noise [5]. To solve these problems, according to the characteristics of corners in Hough parameter space, a new corner detection method via randomized Hough transform based on spatial moments (SMRHT) was proposed. Finally, the image processing algorithms were simulated using Matlab, with the effects of three extraction methods including SMRHT, Harris algorithm [6] and SUSAN algorithm [7].

After extracting corners of the image, it needs camera calibration, in which parameters of the imaging geometric model are calculated according to image information. There are many methods of camera calibration. In this paper,

\*Address correspondence to this author at the Experiment Center, LiRen College of Yanshan University, Qinghuangdao, Hebei, 066004, China; Tel: +8613513358296; E-mail: [luckywcf@126.com](mailto:luckywcf@126.com)

traditional direct linear camera calibration was modified. The method simplified the computational process of camera parameters, to obtain accurate position of the camera, which reduced the error of camera calibration. The experimental results show that the proposed method performed rapid and effective camera calibration and improved the accuracy of calibration.

## 2. IMAGE PREPROCESSING

In the visual servo system of mobile robot, the technology of image processing is used. The information of the image needed is obtained through image gray-scale transformation and edge extraction. Image processing technology by using the computer for processing and recognizing 2D Continuous image camera in the robot's head input, can determine the object's position, direction, and other properties. When the camera captures images, due to uneven ground and the changes in lighting conditions, the obtained image generates noise and distortion. Image processing, for the processing of target images, can remove random noise and distortion, improve the quality of the output image, and make the image clearer [8].

### 2.1. Grayscale Transformation

In image processing, first of all, in order to convert color images into gray-scale image, the collected images follow gray-scale transform. The general formula is as follows:

$$GRAY = 0.30R + 0.59G + 0.11B \quad (1)$$

Where GRAY represents the gray of pixels in 8 bit image, R represents the red component of the corresponding point in 24bit color image, G represents the green component and the blue component, B represents the blue component. In the gray-scale transformation of a colored image, because of the sensitivity of human eye to different colors, various color components have different weights. The transformation uses floating point arithmetic, but the floating point operation is slow, and cannot meet the requirements of localization with robot vision for real-time. Therefore, an attempt was made to improve the computing speed.

In order to avoid the floating point arithmetic, the gray-scale transformation formula was changed into the following form:

$$GRAY = (30 \times R + 59 \times G + 11 \times B) / 100 \quad (2)$$

In which (2) made a division algorithm than (1) but avoided the floating point arithmetic and effectively improved the computing speed.

### 2.2. Median Filter

When the camera captures images, the target image obtained can generate random noise, due to various reasons. Median filter is a nonlinear smoothing technique, which can filter out strong interference noise, and can also protect the boundary of signal change. Especially for salt and pepper noise effect, its filtering effect is remarkable,

as it does not depend on typical values of the surrounding aspects. The basic principle of median filter is that the gray level of each pixel in digital images or image sequences is set to the median of gray level of all pixels within the surrounding window of the point, to eliminate the isolated noise points, in order to make the surrounding pixel value approximate the true value [9].

### 2.3. Canny Edge Detection

Canny operator is an optimal edge detection operator, which determines edge pixels in the image through finding maximum value of the signal function. It gives three criterions to measure the performance of edge detection: good detection performance, high positioning accuracy and the least number of edge responses [10].

Firstly, Canny operator using Gaussian smoothing filters out the noise the image contains, and calculates the convolution of first order differential of Gaussian filter impulse response function and original image function. Following this, by using non maxima suppression, the width of the edge obtained by calculating gradient is decreased, and the maximum points of gradient magnitude are removed. Finally, edges are connected using double threshold detection algorithm. Canny operator can produce good results of edge detection, and the influence of noise is also minor.

## 3. RHT BASED ON SPATIAL MOMENT

Corner features are important features of the image, because the angle is composed of two lines, and the direction of its gradient is different compared with other points around it. Extracting corner features not only accurately determines the target position in the image, but also greatly reduces data quantity in image processing. Therefore, it can increase operational speed of the system and achieve real-time calculation, which is important for accurate robot vision.

### 3.1. Corners in Hough Space

According to the corner detection algorithm based on Hough transform proposed by Guo [11], the concept of corner is defined more intuitively which is more suitable to man's cognitive concept perspective. According to the definition, corner is defined as the center point, in whose domain, exist two straight lines of different slopes through the point and the line at least. This not only defines the corner but also defines direction of the two edge of the angle. According to this definition, a corner is the intersection of two straight lines ideally, but actually will result in the deformation resulting in blurred shape. Because angle represents the graphical features in a certain area at an abstract level, an angle can be considered an intersection of two straight lines in a sufficiently small area. A corner can be determined by three pixel points in image space. There is a model of angle in the image space, formed by three pixel points  $q_1, q_2$  and  $q_3$ , which is transformed into the Hough transform space, as shown in Fig. (1). Three points are expressed as

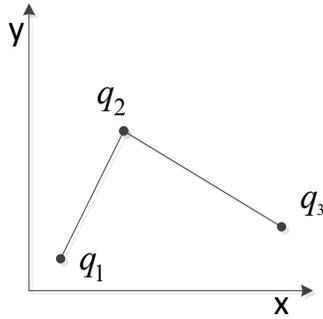


Fig. (1). Corners in image space.

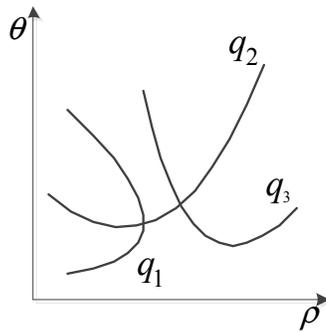


Fig. (2). Corners in hough space.

a sine curve in Hough space. There is an intersection of the corner and every curve in Hough space, but there is also an intersection of  $q_1$  and  $q_2$ , which shows that a peak of parameter space may correspond to a corner in the image space, as shown in Fig. (2). In order to obtain corners in image space, Hough transform of these peaks are inversely transformed using the inversion principle of inverse Hough transform.

Inverse Hough transform is the inversion process of Hough transforms, which can determine the curve by detecting spatial peak of sine curve in Hough space which is the decomposition process of the accumulator array reconstructing image space according to pixel level [12]. Because the feature point of the sine curve in Hough space is nonzero, the curve of the pixel point in image space corresponds to the accumulator array, which means that the pixel value is 1 in the image space [13]. At this time, the image space is quantified for an accumulator array. Subsequently, for each parameter space point in randomized Hough transforms, as long as values of the elements in the accumulator are met, all elements of the accumulator plus one. Finally, those elements whose values are more than one in the accumulator are searched, which may correspond to a corner.

### 3.2. Determine Parameter Space Points

Respectively, image coordinate system and local coordinate system are established in the image space, whose x axis and y axis are parallel and have the same directions. The origin of local coordinate system is random sampling edge

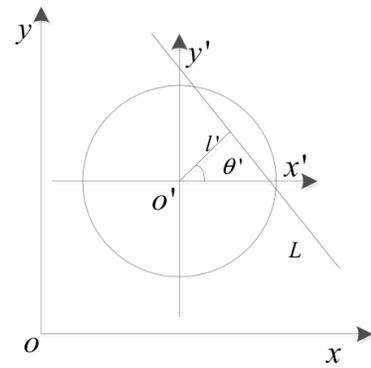


Fig. (3). Image coordinate system and local coordinate system in image space.

point  $(u, v)$ , as shown in Fig. (3). The relationship between two coordinate systems is expressed as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} \quad (3)$$

According to the spatial moment sub-pixel edge detection algorithm, edge parameters  $l'$  and  $\theta'$  in the coordinate system can be calculated. Therefore, sub-pixel coordinate of any point on a straight line in the image coordinate and local coordinate were calculated, respectively [14]:

$$\begin{cases} x_i = x + \frac{N}{2} l \cos \theta \\ y_i = y + \frac{N}{2} l \sin \theta \end{cases} \quad (4)$$

$$\begin{cases} x'_i = x' + \frac{N}{2} l' \cos \theta' \\ y'_i = y' + \frac{N}{2} l' \sin \theta' \end{cases} \quad (5)$$

Straight line L Linear in image coordinate and local coordinate are expressed respectively as :

$$l = x_i \cos \theta + y_i \sin \theta \quad (6)$$

$$l' = x'_i \cos \theta' + y'_i \sin \theta' \quad (7)$$

(7) Are substituted into (3), and results:

$$l' + u \cos \theta' + v \sin \theta' = x_i \cos \theta + y_i \sin \theta \quad (8)$$

Through (6) and (8), it can get:

$$\begin{cases} l = l' + u \cos \theta' + v \sin \theta' \\ \theta = \theta' \end{cases} \quad (9)$$

According to spatial moment algorithm, when the edge of an image represents an ideal edge model, equation (9) becomes the equation of the straight line in image coordinate system. But in reality, the image edge generally does not represent the ideal edge model. Because the image has

complex background and noise, it must have principle error in the calculation of edge parameters  $l'$  and requires error compensation for edge parameters, which increases complexity of the algorithm. Therefore, equation (9) is regarded as a parameter space point through which two edge points by random sampling are determined. However, in order to judge whether it is the real line, random Hough transform is used and the distribution unit of parameter space point is accumulated.

### 3.3. Straight Line Extraction

In the paper, straight line extraction process using randomized Hough transform based on spatial moment was discussed as follows:

Step 1. The coordinates of all edge points were determined in image space of the all edge points into the edge point set  $D$ . Following this, parameter unit set  $P$  was created, and  $k = 0$  was initialized.

Step 2. Randomly two points and  $d_2$  were selected. Restriction condition was set in order to reduce the invalid sampling and accumulation. That is, when the line was perpendicular to the gradient direction or  $d_2$ , these two points were selected. Following this, by using the edge point pair and equation (9) the parameters of linear space point  $p = (l, \theta)$  were determined, otherwise, two points again were randomly selected.

Step 3. A tolerance  $\delta$  was set, and set of an element  $p_c$  was located. When the condition  $\|p_c - p\| < \delta$  was met, Step 5 was followed, otherwise, step 4.

Step 4. Attach to an accumulating cell, whose score is one, and insert it into set as a new element. Go to Step 6.

Step 5. A threshold  $n_t$  was set, the score of the accumulating cell of  $p_c$  was increased by one. When  $score_{p_c} < n_t$ , step 6 was followed, otherwise, step 7.

Step 6.  $k = k + 1$  The largest number of random sampling points was obtained detected on a line. When  $k > k_{max}$ , step 8 was followed, otherwise, followed step 2.

Step 7.  $p_c$  was taken as the parameter of a candidate line, and all the pixels lying on the line were determined. When the number of points lying on the line  $m_{p_c} > m_{min}$ , step 8 was followed, otherwise, the line was taken as false line, and the process was repeated  $P$ , followed by the execution of step 2.

Step 8. A line represented by has been detected. Reset and  $k = 0$ , and go to Step 2.

### 3.4. Corner Detection

In order to extract corners, the parameters of each line were determined as a two-dimensional matrix  $S = (\rho, \theta)$ . The image space as a two-dimensional array  $C(x, y)$  was

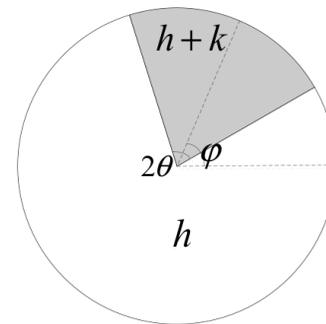


Fig. (4). Angle model.

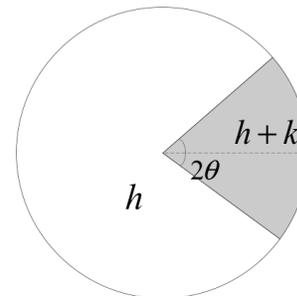


Fig. (5). Rotated angle model.

quantified and attached to an accumulating cell, followed by initialization of the score of all elements to zero. For every element  $(\rho_i, \theta_i)$ , when  $\rho_i = x \cos \theta_i + y \sin \theta_i$  was achieved, the score of element was increased by one. Finally, elements whose scores were greater than one were identified, which may correspond to a corner of the original image.

In order to remove unwanted intersections and false corners, the corners which did not contain any edge were eliminated in the paper using spatial moment. A corner model was defined on the unit circle, as shown in Fig. (4), where  $2\theta$  is the characteristic angle, which is the angle between the characteristic bisector angles and x axis is the background grayscale, the grayscale of the angular region. By rotating angle  $\varphi$ , let the characteristic angle be symmetrical relative to x axis, as shown in Fig. (5).

$f(x, y)$  is the gray-value of the pixel whose coordinate is  $(x, y)$ . Spatial moment of rotated angular characteristic function is:

$$M'_{00} = \iint_{x^2+y^2 \leq 1} f'(x, y) dx dy \tag{10}$$

$$= h \int_0^{2\pi} \int_0^1 \rho d\rho dv + k \int_{-\theta}^{\theta} \int_0^1 \rho d\rho dv = h\pi + k\theta$$

$$M'_{10} = \iint_{x^2+y^2 \leq 1} xf'(x, y) dx dy \tag{11}$$

$$= h \int_0^{2\pi} \int_0^1 \rho^2 \cos v d\rho dv + k \int_{-\theta}^{\theta} \int_0^1 \rho^2 \cos v d\rho dv$$

$$= 0 + \frac{2k \sin \theta}{3}$$

Table 1. The location of corner coordinates using three detection methods.

Real Value	SUSAN	Harris	SMRHT	Salt-pepper Noise	
				Harris	SMRHT
(10,10)	(10,10)	(10,10)	(10,10)	(11,10)	(10,10)
(10,40)	(10,41)	(10,40)	(10,40)	(10,40)	(10,40)
(10,90)	(10,90)	(10,91)	(10,90)	(9,89)	(9.5,90.5)
(20,80)	(20,79)	(20,80)	(20,80)	(20,80)	(20,80)
(30,70)	(30,70)	(31,70)	(30,70)	(30,71)	(30,70)
(40,60)	(40,60)	(40,60)	(40,60)	(38,60)	(40,60)
(50,30)	(50,30)	(49,30)	(50,30)	(51,30)	(50.5,30)
(50,75)	(51,76)	(50,75)	(50,75)	(50,77)	(49.5,74.5)
(60,60)	(61,60)	(60,60)	(60,60)	(61,60)	(60,60)
(80,60)	(80,60)	(81,60)	(80,60)	(80,60)	(80,60)
(90,10)	(90,11)	(90,10)	(90,10)	(90,10)	(90,10.5)
(90,75)	(90,75)	(90,75)	(90,75)	(89,74)	(90,76)
Errors	(0.16,0.33)	(0.25,0.08)	(0,0)	(0.58,0.5)	(0.125,0.21)

$$\begin{aligned}
 M'_{01} &= \iint_{x^2+y^2 \leq 1} yf'(x,y) dx dy \\
 &= h \int_0^{2\pi} \int_0^1 \rho^2 \sin v d\rho dv + k \int_{-\theta}^{\theta} \int_0^1 \rho^2 \sin v d\rho dv \\
 &= 0
 \end{aligned} \tag{12}$$

$$\begin{aligned}
 M'_{20} &= \iint_{x^2+y^2 \leq 1} x^2 f'(x,y) dx dy \\
 &= h \int_0^{2\pi} \int_0^1 \rho^3 \cos^2 v d\rho dv + k \int_{-\theta}^{\theta} \int_0^1 \rho^3 \cos^2 v d\rho dv \\
 &= \frac{\pi}{4} h + \frac{k}{4} \theta + \frac{k}{8} \sin 2\theta
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 M'_{20} &= \iint_{x^2+y^2 \leq 1} x^2 f'(x,y) dx dy \\
 &= h \int_0^{2\pi} \int_0^1 \rho^3 \sin^2 v d\rho dv + k \int_{-\theta}^{\theta} \int_0^1 \rho^3 \sin^2 v d\rho dv \\
 &= \frac{\pi}{4} h + \frac{k}{4} \theta - \frac{k}{8} \sin 2\theta
 \end{aligned} \tag{14}$$

According to (10) ~ (14), the corner model meets:

$$2(M'_{20} + M'_{02}) - M'_{00} = 0 \tag{15}$$

$$M'_{20} - M'_{02} = \frac{k \sin \theta \cos \theta}{2} \tag{16}$$

Therefore, if the spatial moment of candidate corner meets (15) and (16), it is determined as a real corner. Corner detection algorithm proposed in this paper not only improved the detection speed and reduced the calculation time, but also

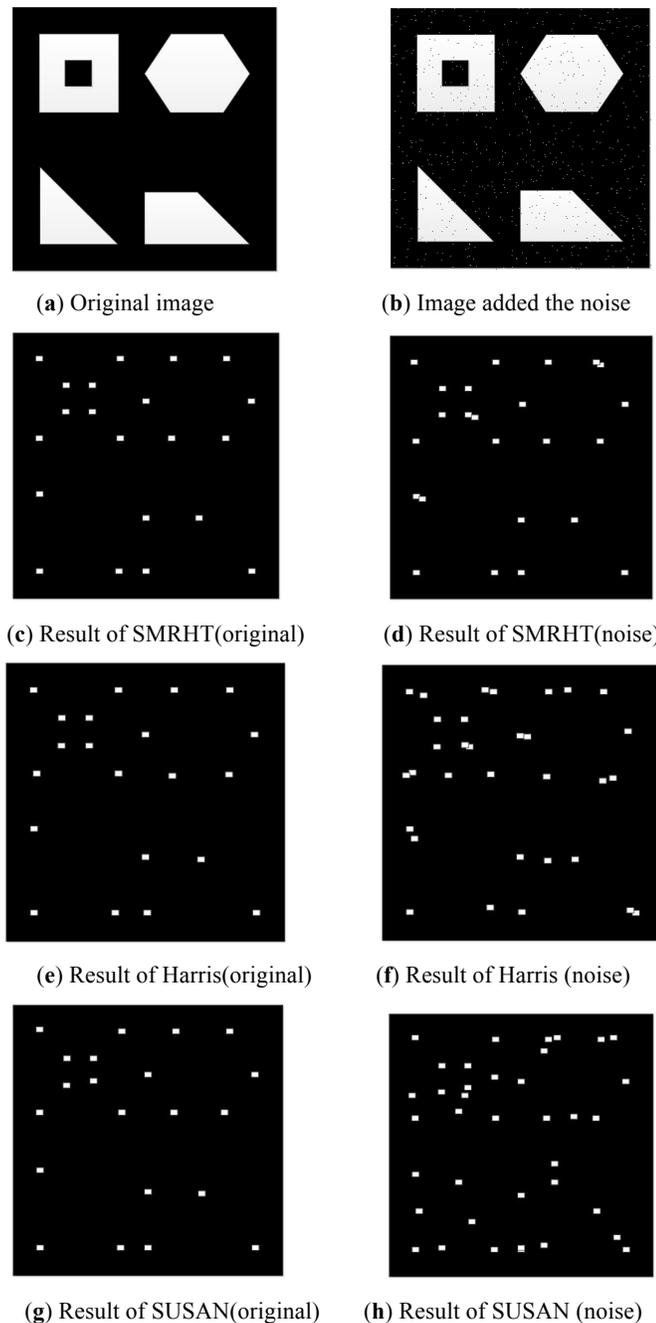
minimized the influence of noise on linear parameter, and greatly reduced invalid sampling and accumulation, Since spatial moment algorithm has the advantages of high detection accuracy, simple calculation process and computational...? the process is relatively simple and good in being anti-noise. On the other hand, when calculating parameter space points and removing the false corners, by using sub-pixel coordinate information in image space, the calculation accuracy was improved, which extracted accurate corners of the image coordinate, and met the real-time robot visual location.

### 3.5. Analysis of Extraction Results

In order to test the accuracy and robustness of SMRHT algorithm, the synthetic and real images were selected, which were used to test the algorithm in the paper.

By using computer an image was synthesized, as shown in Fig. (6). In the image, there were twenty-one corners, including 45 degree, 90 degree and 135 degree angles. Table 1 shows the location of corner coordinates using three detection methods. It can be seen that in the ideal case, the test results of SUSAN algorithm and Harris algorithm had respectively five and four error corners, and average error was (0.33, 0.16) and (0.25, 0.08), while the error of RMRHT was (0,0). In the case of using salt-pepper noise, due to the impact of noise, detection error increased, and the error of Harris method observed was (0.58,0.5), while the error of SMRHT calculated was only (0.125, 0.21).

Fig. (6) shows the results of Harris method, SUSAN method and SMRHT method in ideal and salt-pepper noise



**Fig. (6).** Experimental results of synthetic image.

cases. From Fig. (6), in the case of salt-pepper noise whose variance was 0.01, SMRHT still obtained a good detection result, while the result of Harris was slightly poor. The result of SUSAN algorithm was affected by the noise largely, therefore, there were many false corners. Table 1 and Fig. (6) illustrate that SMRHT algorithm had better localization performance and robustness than SUSAN and Harris.

In order to test robustness and localization performance of SMRHT algorithm, the true image size captured by the camera, was used to conduct another experiment, as shown in Fig. (7). The experimental results show that SMRHT algorithm had a high corner detection rate. There were 374 corners detected, among which 10 were false corners and 14

were missed corners. The average error of SMRHT algorithm was only 0.79 pixels, the success rate was 94%, and the time was 470 ms. There were total 143 corners detected and many missed corners in the results of Harris, whose average error was 1.32 pixels and the time was 712ms. Detection and success rate of SUSAN were the worst, whose results had a large number of false and missed corners, while the average error was 2.32 pixels, and the time was 456 ms. These errors clearly showed that SMRHT algorithm is better than SUSAN and Harris in corner detection stability and accuracy, and lays the foundation for camera calibration, and meets the system reliability.

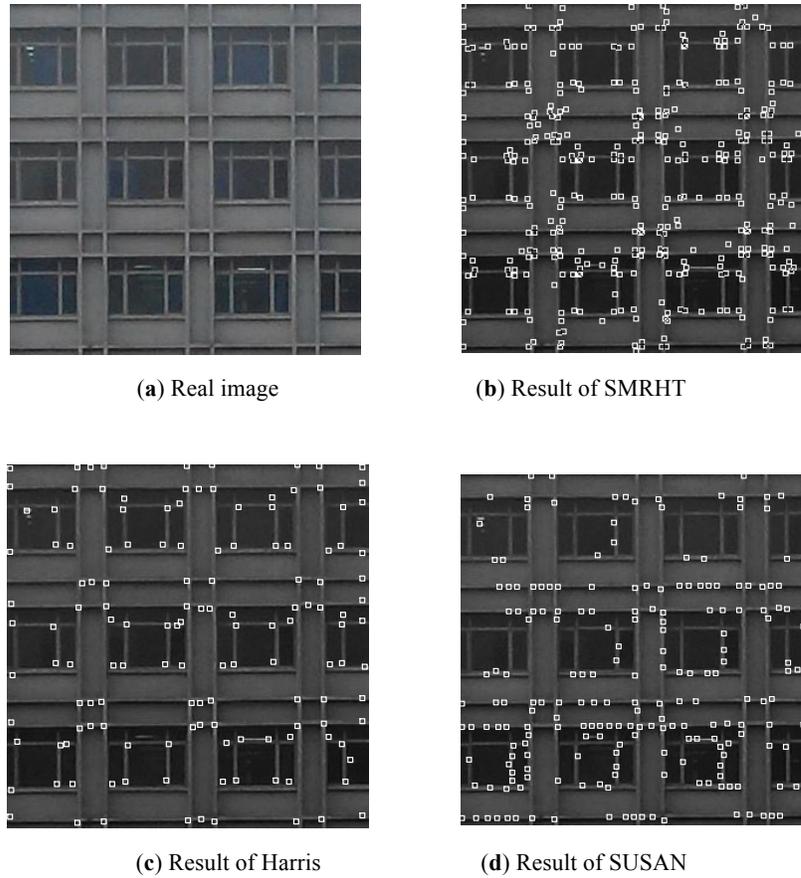


Fig. (7). Experimental results of real image.

#### 4. CAMERA CALIBRATION

##### 4.1. Improved Direct Linear Calibration

Camera calibration, which is used to calculate the intrinsic and extrinsic parameters of a cameraview, is one of the basic tasks for analyzing the geometry of a picture/scene and determining the shapes and locations of objects in vision. Therefore, the search for a fast camera calibration method is an important problem in computer vision applications. Because the linear calibration method needs 3D template whose production process is complicated, the goal is not only hard to achieve, but also requires using at least 6 group of equations, while solving the projection matrix. Therefore, in order to camera calibrate quickly and effectively, planar calibration target is used which is simple and easy to implement, while direct linear calibration needs to be improved [15].

Because the position of world coordinate system can be chosen arbitrarily when doing camera calibration, it was assumed that the template plane was located in xy plane of the world coordinate system, that is  $z = 0$ . The relation between the image coordinates and the world coordinates was transformed into:

$$\begin{aligned}
 \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \\ x_w \\ y_w \\ 1 \end{bmatrix} \\
 &= \mathbf{K}[\mathbf{R} \ \mathbf{T}] \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}
 \end{aligned} \tag{17}$$

Where  $\mathbf{k}$  is camera intrinsic parameters matrix, determined by the internal parameters  $(\alpha_x, \alpha_y, u_0, v_0)$  of the camera,  $\mathbf{R}$  and  $\mathbf{T}$  are rotation matrix and translation vector respectively, and  $\mathbf{M}$  is the projection matrix.

During calibration, the coordinate of an arbitrary point in the coordinate system is  $(x_{wi}, y_{wi}, 0)$ , whose corresponding image coordinate is measured. Each feature point can be expressed:

$$\begin{aligned}
 \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} &= \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} x_{wi} \\ y_{wi} \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} m_{11} & m_{12} & m_{14} \\ m_{21} & m_{22} & m_{24} \\ m_{31} & m_{32} & m_{34} \end{bmatrix} \begin{bmatrix} x_{wi} \\ y_{wi} \\ 1 \end{bmatrix}
 \end{aligned} \tag{18}$$

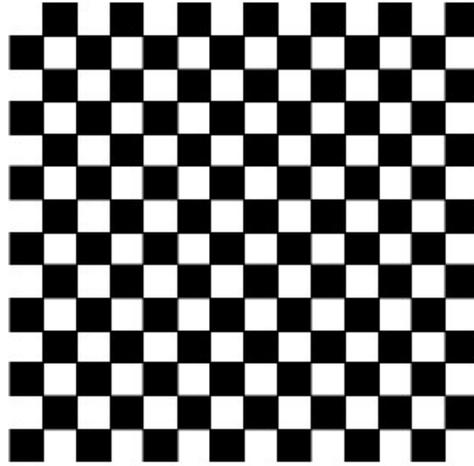


Fig. (8). Calibration target.

$$\begin{cases} x_{wi}m_{11} + y_{wi}m_{12} + m_{14} - u_i x_{wi}m_{31} - u_i y_{wi}m_{32} = u_i \\ x_{wi}m_{21} + y_{wi}m_{22} + m_{24} - v_i x_{wi}m_{31} - v_i y_{wi}m_{32} = v_i \end{cases} \quad (19)$$

Set  $m_{34}=1$  and solve the over-determined equations. There are 8 unknowns in matrix  $\mathbf{M}$ , which can be calculated using five coordinates using least square method. After obtaining the matrix  $\mathbf{M}$ , camera parameters can be calculated by (20):

$$\begin{bmatrix} m_{11} & m_{12} & m_{14} \\ m_{21} & m_{22} & m_{24} \\ m_{31} & m_{32} & m_{34} \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 & r_1 & t_x \\ r_3 & r_4 & t_y \\ r_6 & r_7 & t_z \end{bmatrix} \quad (20)$$

The real solution of is:

$$\rho \begin{bmatrix} m_1^T & m_{14} \\ m_2^T & m_{24} \\ m_3^T & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1^T & t_x \\ r_2^T & t_y \\ r_3^T & t_z \end{bmatrix} \quad (21)$$

Where  $\rho$  is unknown proportional Coefficient, is row vector Composed of the first two elements in the  $i$ -row of matrix  $\mathbf{M}$ , is the  $i$ -row in the column vectors  $r_1$  and  $r_2$  of the rotation matrix, is the three components of the translation vector.

Rows of the rotation matrix make the unit length, therefore, :

$$\rho = \frac{1}{|m_3|} \quad (22)$$

Thus, the values of each element in the matrix were obtained. Through intrinsic parametric matrix titled  $\mathbf{K}$ ,  $r_1$  and  $r_2$  in rotation matrix  $\mathbf{R}$  and translation vector using RQ decomposition can be obtained. Because  $r_3 = r_1 \times r_2$ , rotation matrix can also be determined.

### 4.2. Calibration Experiment

A checkerboard calibration target was adopted in the experiment, as shown in Fig. (8). In order to obtain camera calibration parameters, corner feature points were extracted using SMRHT corners detection algorithm, and using improved direct linear calibration algorithm, the calibrating experiments were carried out. Calibration accuracy was measured through the difference between the real and reconstructed 3-D coordinates of corners. That is:

$$\mathcal{E} = \frac{\sum_{i=1}^N |X_u - X_d|}{N} \quad (23)$$

Where... is the 3D coordinate of real corner, which is the reconstructed 3D coordinate of corner, and  $N$  is the number of corner points.

According to the world coordinates and image coordinates of the corners, the values of each element in matrix were calculated by least square method using Matlab. The calibration results are as follows:

$$\text{Projection matrix } \mathbf{M} = \begin{bmatrix} 2.9746 & 0.3586 & 310.6712 \\ 0.0701 & 0.3588 & 467.5197 \\ 0.0006 & 0.0021 & 1.0000 \end{bmatrix},$$

Intrinsic parameters matrix

$$\mathbf{K} = \begin{bmatrix} 935.5713 & 0 & 318.9520 \\ 0 & 935.6254 & 256.4186 \\ 0 & 0 & 1.0000 \end{bmatrix},$$

$$\text{Rotation matrix } \mathbf{R} = \begin{bmatrix} 0.6606 & 0.0326 & 0.7899 \\ -0.4601 & 0.0362 & 0.6348 \\ 0.0012 & -0.8661 & 0.1572 \end{bmatrix},$$

$$\text{and translation vector } \mathbf{T} = \begin{bmatrix} 11.6345 \\ 23.8498 \\ 16.4972 \end{bmatrix}$$

Table 2. Corners coordinates of 9 images in image and actual coordinates.

Image Number	$(x_a, y_a)$		$(x_b, y_b)$	
	Image Coordinates (pixel)	Actual Coordinates (cm)	Image Coordinates (pixel)	Actual Coordinates (cm)
1	(124.9,566.5)	(-0.1,-80.2)	(212.3,558.5)	(40.1,-80.2)
2	(118.3,538.7)	(-0.2,-70.4)	(213.6,530.1)	(40.4,-70.3)
3	(127.1,457.9)	(0.1,-63.2)	(208.9,460.7)	(40.3,-63.3)
4	(464.4,425.8)	(90.1,35.6)	(458.8,290.2)	(89.5,65.1)
5	(459.1,499.8)	(90.2,25.5)	(462.2,340.0)	(90.5,56.4)
6	(463.3,575.2)	(89.8,16.3)	(451.7,558.5)	(90.7,45.9)
7	(213.6,300.1)	(40.1,70.1)	(219.1,199.2)	(40.5,111.1)
8	(216.7,320.6)	(40.2,59.8)	(210.4,210.6)	(40.4,98.4)
9	(214.9,363.1)	(40.4,50.1)	(212.3,253.8)	(40.3,89.6)

Table 3. Robotic calculated displacement, actual displacement, and absolute error.

Image	Displacement in X Direction (cm)			Displacement in Y Direction (cm)		
	Actual Value	Calculated Value	Absolute Error	Actual Value	Calculated Value	Absolute Error
1→2	0	0.1	0.1	-10	-9.6	0.4
2→3	0	0.2	0.2	-10	-9.8	0.2
3→4	0	-0.1	0.1	-10	-10.2	0.2
4→5	-10	-9.8	0.2	0	0.1	0.1
5→6	-10	-10.1	0.1	0	-0.2	0.2
6→7	0	0.1	0.1	10	9.7	0.3
7→8	0	-0.2	0.2	10	10.2	0.2
8→9	0	-0.1	0.1	10	9.9	0.1

Finally, calculated calibration accuracy was  $\mathcal{E} = 0.048mm$ .

The data indicate that, the method had low requirement for experiment, It had a high calibration accuracy and rapid computing speed, which can accurately estimate the pose of the target with respect to the camera, therefore, the influence of external factors on positioning was small and it can meet the general requirements of industrial vision.

### 5. VISUAL POSITIONING EXPERIMENTS

For visual positioning experiments in the paper, the characteristics of the ground images which the camera in robot captured, were identified and analyzed. In the experiment, the intersection of grid lines of tile which is a square of about 40cm per side on the ground was used as the feature points of images, known as corners. Firstly, the image processing and SMRHT algorithm were used for extracting corners and obtaining pixel values of these corners. Following this, using improved direct linear calibration, the image pixel values

were mapped to the coordinate system of actual ground and the actual coordinate values of the corners were obtained. Finally, the actual coordinates of 4 corners were matched with the two images of current moment and next moment, and according to the displacement relationship of these corners, combined with their own information, and displacement and direction of the robot in X and Y direction were calculated.

The robotic center was set as the origin of the coordinate system, *i.e.* when the robot acquired the first image, its coordinate was  $(0, 0)$ . In moving process of the robot, the relative positions of the intersections are unchanged. To calculate the robotic pose when it acquired the second image, it selected two corners on the same edge and calculated the average value of 2 corner's displacement in X and Y direction. The displacement of robot in X and Y direction with and  $D_y$  are expressed as:

Table 4. Real values, calculated values and absolute error of direction.

Image	Origin Coordinate (cm)		Direction Angle (Degree)		
	Actual Value	Calculated Value	Actual Value	Calculated Value	Absolute Error
1	(0,0)	(0.2,0.3)	0	0.1	0.1
2	(0,10)	(0.1,9.7)	0	-0.4	-0.4
3	(0,20)	(0.3,19.6)	0	0.3	0.3
4	(0,30)	(0.1,19.7)	90	89.3	-0.7
5	(-10,30)	(-9.8,30.1)	90	88.9	-0.1
6	(-20,30)	(-19.9,29.8)	180	178.8	-1.2
7	(-20,20)	(-20.3,20.2)	180	180.7	0.7
8	(-20,10)	(-19.8,9.9)	180	180.3	0.3
9	(-20,0)	(-20.2,0.2)	90	90.5	0.5

$$\begin{cases} D_x = \frac{(x_{a2} - x_{a1}) + (x_{b2} - x_{b1})}{2} \\ D_y = \frac{(y_{a2} - y_{a1}) + (y_{b2} - y_{b1})}{2} \end{cases} \quad (24)$$

The moving direction of the robot is represented as the angle between origin of the coordinate of robot and X axis, that is  $\theta = \arctan(D_y/D_x)$ .

In the indoor environment, the robotic camera is placed horizontally. In moving process of the robot, using the camera, nine consecutive ground images are captured. Due to limited space, the paper only included the former two images, as shown in Fig. (9).

In order to obtain the coordinates of corners in the actual coordinate system, the coordinates of the corners in image coordinate system are substituted into (18), as shown in Table 2.

When calculating robotic displacement in X and Y direction, it required similarity between the two adjacent images. To calculate these displacements, the coordinates of image feature points in actual coordinate in Table 2 were substituted into (24), and the actual displacements of the robot were measured. The obtained results are shown as Table 3. The accuracy of the actual coordinate values directly affects the robotic displacement.

Through (25), in order to acquire the direction of the robot, the actual coordinate origin of the robot was calculated and measured, as shown in Table 4.

The results in Table 2 and Table 4 show that there were little errors between the calculated and the actual moving displacement of the robot. The absolute values of displacement error in X and Y direction were in between 0 cm to 0.2 cm and 0cm to 0.4cm respectively. The absolute values of direction angle error were between 0 degree and 1.2 degree. With minor errors, the system was capable to meet the demands of project application.

## CONCLUSION

Camera calibration is a necessary step to obtain accurate 3D information from 2D images, whose accuracy depends largely on the accuracy of image positioning of control points on calibration target. In view of this, using spatial moment's operator and random Hough transform theory, with corner detection algorithm *via* random Hough transform based on spatial moment was studied. Because the peaks in Hough space can correspond to the corners in image space, firstly, by using randomized Hough transform and spatial moment theory, linear parameters were extracted in the image space. Subsequently, according to the inversion principle of inverse Hough transform, the intersections of these lines were obtained, which served as the candidate corners in the image space. Finally, real corners were determined through the feature of corner model of spatial moments. Using SMRHT algorithm corner pixels were acquired rapidly and accurately, which reduced the error of camera calibration and improved positioning accuracy.

In order to obtain the position of the target points in the actual coordinate system, an improved direct linear camera calibration method was proposed in the paper, which reduced the effects of errors observed, and greatly simplified the calculation process using least square method. The experimental results show that the method is not only simple and effective, and needs no special experimental conditions and can be easily operated.

## CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

## ACKNOWLEDGEMENTS

This work was supported by The National Natural Science Foundation, China (No.60970073). This work was also supported by HeBei National Natural Science Foundation,

China (No.F2012203084) and HeBei National Natural Science Foundation, China (No.A2012203124).

## REFERENCES

- [1] M.L. Qiu, S.D. Ma, and Y. Li. "Overview of camera calibration for computer vision", *Acta Automatica Sinica*, vol. 26, no. 1, pp. 43-55, 2000.
- [2] J. Ramesh, R. Kasturi, and B.G. Schunck, *Machine Vision*. China Machine Press: Beijing, 2003.
- [3] M. Hou, and S. Xing. "Study of improving the stability of SUSAN corner detection algorithm", In: *IEEE Press, International Conference on ICCASM. Taiyuan, China*, vol. 11, pp. 627-630, 2010.
- [4] W. Li, Z. Shen, and B. Li, "Corner detection based on local curvature function", *Computer Engineering and Design*, vol. 28, no. 11, pp. 2595-2596, 2007.
- [5] J. Hou, and Y.I. Lin. "Adaptive Harris X-corner detection algorithm", *Computer Engineering and Design*, vol. 30, no. 20, pp. 4741-4743, 2009.
- [6] H.Y. Zhang, Y.Y. Li, and C.Y. Chu. "Multi-scale Harris corner detection based on image block", *Journal of Computer Applications*, vol. 31, no.1, pp. 356-357, 2001.
- [7] M.L. Hou, and S.B. Xing. "Study of improving the stability of SUSAN corner detection algorithm", In: *International Conference on ICCASM. Taiyuan, China: IEEE Press*, vol. 11, no.1, pp. 627-630, 2010.
- [8] P. Xavier, A. Burgos, A. Ribeiro, M. Guuijarro, and G. Pajares, "Real-time image processing for crop weed discrimination in maize fields", *Computers and Electronics in Agriculture*, vol. 75, no.2, pp. 337-346, 2011.
- [9] X.M. Zhang, B.S. Xu, and S.Y. Dong, "Adaptive median filtering for image processing", *Journal of Computer-aided Design & Computer Graphics*, vol. 17, no. 2, pp. 295-299, 2005.
- [10] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no.6, pp. 679-698, 1986.
- [11] S.Y. Guo, Y.G. Kong, and X.F. Zhang, "Corner detection algorithm based on Hough transform", *Chinese Journal of Scientific Instrument*, vol. 29, no. 11, pp. 2424-2429, 2008.
- [12] A.L. Kesidis, and N. Papamarkos, "On the inverse Hough transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1329-1343, 1999.
- [13] Y.M. He, and S.G. Dai, "Detecting the corners of light shape of automotive low-beam headlamp by inverse hough transform", In: *International Conference on Industrial and Information Systems*. Haikou, China: IEEE Press, vol. 63, pp. 309-311, 2009.
- [14] E.P. Lyvers, O.R. Mitchell, M.L. Akey, and A.P. Reeves, "Subpixel measurements using a moment-based edge operator", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1293-1309, 1989.
- [15] W.D. Chen, H.J. Yin, and Q.G. Zhu, "Improved direct linear calibration in mobile robot vision location", *Journal of Chinese Computer Systems*, vol. 34, no. 4, pp. 872-876, 2013.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Wang *et al.*; Licensee Bentham Open

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.