

Article



# Automatic Camera Calibration Using Active Displays of a Virtual Pattern

Lei Tan <sup>1,\*</sup>, Yaonan Wang <sup>1,2</sup>, Hongshan Yu <sup>2,\*</sup> and Jiang Zhu <sup>3</sup>

- <sup>1</sup> College of Electrical and Information Engineering, Hunan University, Changsha 410082, China; wangyaonan@hnu.edu.cn
- <sup>2</sup> National Engineering Laboratory for Robot Visual Perception and Control Technology, Hunan University, Changsha 410082, China
- <sup>3</sup> College of Information Engineering, Xiangtan University, Yuhu District, Xiangtan 411105, China; jiang126@126.com
- \* Correspondence: leit@hnu.edu.cn (L.T.); yuhongshan@hnu.edu.cn (H.Y.); Tel.: +86-731-8882-2224 (L.T.)

Academic Editor: Vittorio M. N. Passaro Received: 12 January 2017; Accepted: 18 March 2017; Published: 27 March 2017

Abstract: Camera calibration plays a critical role in 3D computer vision tasks. The most commonly used calibration method utilizes a planar checkerboard and can be done nearly fully automatically. However, it requires the user to move either the camera or the checkerboard during the capture step. This manual operation is time consuming and makes the calibration results unstable. In order to solve the above problems caused by manual operation, this paper presents a full-automatic camera calibration method using a virtual pattern instead of a physical one. The virtual pattern is actively transformed and displayed on a screen so that the control points of the pattern can be uniformly observed in the camera view. The proposed method estimates the camera parameters from point correspondences between 2D image points and the virtual pattern. The camera and the screen are fixed during the whole process; therefore, the proposed method does not require any manual operations. Performance of the proposed method is evaluated through experiments on both synthetic and real data. Experimental results show that the proposed method can achieve stable results and its accuracy is comparable to the standard method by Zhang.

**Keywords:** camera calibration; 2D pattern; active display; lens distortion; closed-form solution; maximum likelihood estimation

# 1. Introduction

Camera calibration is the first process for 3D computer vision which recovers metric information from 2D images. There are two types of approaches for calibration: photogrametric calibration uses both 2D information and knowledge of the scene such as coordinates of 3D points, shape of reference objects, direction of 3D lines, etc.; self-calibration does not require any knowledge but only 2D information. Generally speaking, the former approaches give more stable and accurate calibration results than the latter because using the knowledge reduces the number of parameters. The proposed method in this paper belongs to the photogrametric approaches.

The standard photogrametric calibration is Zhang's method [1] which uses a 3D plane called a chessboard or checkerboard, even though many methods have been proposed which use perpendicular planes [2,3], circles [4,5], spheres [6,7], and vanishing points [8,9]. The merits of Zhang's method are the ease of use and its extensibility. The requirement is only a camera and a paper on which a pattern is printed. Pattern images are captured by moving either the camera or the plane manually. Then, camera parameters are estimated by decomposing the homography between 3D points on the plane and their 2D projections on the image. The basic idea of Zhang's method is not only

for a single camera calibration, but also applicable to multiple camera calibration [10], projector-camera calibration [11], and depth sensor-camera calibration [12].

Most parts of Zhang's conventional method, such as checkerboard detection, can be automatically processed by software [13,14]. However, a manual part remains at the capture step. This part makes a calibration result unstable although it takes a lot of time. For stable calibration, many images under varied motions, generally  $\geq$ 20 images, are required so that all detected points are distributed uniformly. Figure 1a shows an example in which all points from four images are scattered over the camera view. Otherwise, in a situation like Figure 1b, the conventional method does not give an accurate result for any trials.

To get well distributed points, robust methods are proposed for detecting partial occluded patterns [15–17]. By using those methods, if a part of the pattern is outside of the camera view, visible points including those near the image boundary are helpful for improving calibration accuracy. However, the manual part still exists.

This paper proposes a full-automatic calibration method to resolve the two problems caused by the manual operation: the time consuming problem and the point distribution problem. Instead of a physical pattern, the proposed method uses a virtual pattern which is transformed in the virtual world coordinates and projected on a fixed screen. The pattern on the screen is captured by a fixed camera, then, the proposed method performs calibration by using point correspondences between the virtual 3D points and their 2D projections. The virtual pattern can be actively displayed on the screen so that all points are uniformly distributed. Also, the camera and the screen are fixed during the whole process. Therefore, the proposed method can be stable and fully automatic.

This paper is organized as follows. Section 2 describes Zhang's conventional method from basic equations. Although the derivation of Zhang's method is widely known, it is highly related to the proposed method in Section 3. In Section 4, experimental results on synthetic and real images are provided and discussed. Finally, Section 5 gives the conclusions.



**Figure 1.** Distribution of Detected Points. (**a**) Detected points are distributed uniformly across the image; (**b**) Detected points are mainly located at the center part of the image.

## 2. Conventional Method

Zhang's conventional calibration method estimates the intrinsic and the extrinsic parameters of a camera from images of a physical planar pattern. Figure 2a shows an overview where the camera is moved by hand to take the pattern images.





**Figure 2.** Overview of the conventional method and the proposed method. (**a**) The conventional method; (**b**) The proposed method.

# 2.1. Basic Equations

Assume that *n* 3D points are on a z = 0 plane and the plane is shot by a pinhole model camera with *m* times. In a *j*-th shot ( $j \le m$ ), the relation between a 3D point  $X_i = [x_i, y_i, 0]^T$  ( $i \le n$ ) and its 2D projection  $m_{ij} = [u_{ij}, v_{ij}]^T$  can be expressed by

$$\begin{bmatrix} m_{ij} \\ 1 \end{bmatrix} \propto K \begin{bmatrix} R_j & t_j \end{bmatrix} \begin{bmatrix} X_i \\ 1 \end{bmatrix}$$
(1)

where  $\propto$  denotes equality up to scale,  $R_j$  is a *j*-th 3 × 3 rotation matrix,  $t_j$  is a *j*-th 3 × 1 translation vector, and *K* is a 3 × 3 upper triangular matrix given by

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$
(2)

with  $[u_0, v_0]$  the principal point, *s* the skewness, and  $[f_x, f_y]$  the focal length for *x* and *y* axis.

The third column of  $R_i$  can be eliminated due to z = 0. From Equation (1), then we have

$$\begin{bmatrix} m_{ij} \\ 1 \end{bmatrix} \propto K \begin{bmatrix} r_{j1} & r_{j2} & t_j \end{bmatrix} \begin{bmatrix} x_i \\ 1 \end{bmatrix}$$
(3)

where  $x_i = [x_i, y_i]^T$ ,  $r_{jk}$  denotes the the *k*-th column of  $R_j$ . Furthermore we can simplify this projection by using a 3 × 3 matrix

$$H_j \propto K \begin{bmatrix} r_{j1} & r_{j2} & t_j \end{bmatrix}.$$
(4)

 $H_j$ , called a homography matrix, is given by at least four point correspondences  $m_{ij}$  and  $X_i$  [1]. Multiplying  $K^{-1}$  from the left side of Equation (4) and using the orthogonality of  $R_j$ , we obtain two constraints for K:

$$h_{j1}^T B h_{j2} = 0,$$
 (5)

$$h_{j1}^T B h_{j1} - h_{j2}^T B h_{j2} = 0 (6)$$

where  $B \propto K^{-T}K^{-1}$ , and  $h_{jk}$  denotes the *k*-th column of  $H_j$ . *B* is a 3 × 3 symmetric matrix and has a six components. However, the degrees of freedom is five due to the scale ambiguity.

#### 2.2. Estimating Parameters

Equations (5) and (6) are linear to *B*. Therefore, we can obtain *B* by solving

$$Vvec(B) = 0, (7)$$

where V is a 2 m  $\times$ 6 matrix and vec() is a vectorization operator. Note that the dimension of vec(B) is six. In a general case, where all the intrinsic parameters are unknown,  $m \ge 3$  observations are required for getting a unique solution of vec(B). After getting B, K is extracted by decomposing B. More details on estimating the intrinsic parameters are described in [1] and [18].

Once *K* is known,  $R_i$  and  $t_i$  can be recovered as

$$R_{j} = \begin{bmatrix} \lambda K^{-1} h_{j1} & \lambda K^{-1} h_{j2} & r_{j1} \times r_{j2} \end{bmatrix},$$

$$t_{j} = \lambda K^{-1} h_{j3}$$
(8)
(9)

$$t_j = \lambda K^{-1} h_{j3} \tag{9}$$

with scale factor  $\lambda = 1/||K^{-1}h_{i1}|| = 1/||K^{-1}h_{i2}||$ . Because of noisy data,  $R_i = [r_{i1}, r_{i2}, r_{i3}]$  derived from the above equation does not generally satisfy the properties of a rotation matrix. The best rotation matrix from a general  $3 \times 3$  matrix can be estimated through singular value decomposition [18].

#### 2.3. Nonlinear Refinement

The estimated parameters above are not accurate because they are derived by linear methods based on the algebraic error without lens distortion. To refine the linear estimation, a nonlinear optimization is carried out by minimizing the re-projection error:

$$\begin{array}{ll} \min_{K,R_j,t_j \ \forall j \in m} & \sum_{j \forall m} \sum_{i \forall n} \|m_{ij} - p(X_i, K, R_j, t_j, d)\|^2 \\ \text{s.t.} & R_j^T R_j = I \ \forall j \in m \end{array}$$
(10)

where *I* is the  $3 \times 3$  identity matrix, and *p* is a projective function with lens distortion parameter *d*.

# 3. Proposed Method

As shown in Figure 2b, the proposed method uses a virtual calibration pattern instead of a physical one. The virtual pattern is transformed by some pre-generated parameters and projected onto a screen, then, the pattern on the screen is captured by a fixed camera. For stable calibrations, the virtual pattern is actively displayed on the screen and these pre-generated parameters ensure that all 2D projections of the corner points are uniformly distributed in the camera coordinates. The proposed method estimates the intrinsic and the extrinsic parameters from correspondences between the virtual world points and their 2D projections.

In contrast to the conventional method, the proposed method does not require moving either the camera or the pattern. Since the camera and the screen are fixed during the whole process, the proposed method can be implemented as a fully automatic calibration software.

# 3.1. Basic Equations

Let  $P = K \begin{bmatrix} R & t \end{bmatrix}$  be the projection from the screen to the camera and  $P_j^s = K^s \begin{bmatrix} R_j^s & t_j^s \end{bmatrix}$  be the projection from the virtual pattern to the screen where  $K^s$ ,  $R_j^s$ , and  $t_j^s$  are the screen's intrinsic and *j*-th extrinsic parameters, respectively.

Then, the projection between a virtual world space 3D point  $X_i$  and a 2D image point  $m_{ii}$  can be expressed by

$$\begin{bmatrix} m_{ij} \\ 1 \end{bmatrix} \propto \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} P \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} P_j^s \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_i \\ 1 \end{bmatrix}$$
(11)

where 0 is a  $3 \times 1$  zero vector.

Let us consider the two projections separately. The first projection by  $P_i^s$  can be rewritten by

$$\begin{bmatrix} P_j^s \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_i \\ 1 \end{bmatrix} = \begin{bmatrix} K^s & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} R_j^s & t_j^s \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_i \\ 1 \end{bmatrix}$$
(12)

$$= \begin{bmatrix} K^s & 0\\ 0^T & 1 \end{bmatrix} \begin{bmatrix} r^s_{j1} & r^s_{j2} & t^s_{j}\\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i\\ 1 \end{bmatrix}$$
(13)

$$= \begin{bmatrix} H_j^s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ 1 \end{bmatrix}$$
(14)

where  $r_{jk}^s$  denotes the *k*-th column of  $R_j^s$ , and  $H_j^s = K^s \left[ r_{j1}^s \quad r_{j2}^s \quad t_j^s \right]$ .  $K^s$  is the screen's intrinsic parameters which are preset in the calibration, and  $R_j^s$  and  $t_j^s$  are the extrinsic parameters of the screen at the j-th capture in the calibration. Since the virtual pattern is transformed by pre-generated parameters,  $R_j^s$  and  $t_j^s$  are actually known. Also the second projection by *P* can be rewritten by

$$\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} P \\ 0^T & 1 \end{bmatrix} = P \tag{15}$$

$$= K \begin{bmatrix} R & t \end{bmatrix}.$$
(16)

Letting  $h_{jk}^s$  be the *k*-th column of  $H_j^s$ , and from Equations (14) and (16), we can write Equation (11) by using a 3 × 3 homography:

$$\begin{bmatrix} m_{ij} \\ 1 \end{bmatrix} \propto H_j \begin{bmatrix} x_i \\ 1 \end{bmatrix}$$
(17)

where

$$H_j \propto K \begin{bmatrix} Rh_{j1}^s & Rh_{j2}^s & Rh_{j3}^s + t \end{bmatrix}.$$
 (18)

Similarly to the conventional method, given virtual world space 3D points and their 2D image projections, homography  $H_j$  can be calculated using the same technique introduced in Zhang's paper [1]. However, we cannot extract constraints from Equation (18) in the same way as Equations (5) and (6) since the form of  $H_j$  is not identical. The proposed method uses the ratio constraints of the vector dot product instead of the orthogonality.

Multiplying  $K^{-1}$  from the left side of Equation (18), we have three equations from the first and the second columns:

$$\|K^{-1}h_{j1}\|^2 \propto \|h_{j1}^s\|^2 \tag{19}$$

$$\|K^{-1}h_{j2}\|^2 \propto \|h_{j2}^s\|^2 \tag{20}$$

$$\left(K^{-1}h_{j1}\right)^{T}\left(K^{-1}h_{j2}\right) \propto h_{j1}^{sT}h_{j2}^{s}$$
 (21)

where  $h_{jk}$  denotes the *k*-th column of  $H_j$ . If we take a ratio from any two of the above equations, we can obtain one constraints. For example, picking Equations (19) and (20), we have

$$\|h_{j2}^{s}\|^{2}\|K^{-1}h_{j1}\|^{2} - \|h_{j1}^{s}\|^{2}|K^{-1}h_{j2}\|^{2} = 0.$$
(22)

There are three possible combinations, but only two of them are linearly independent. Thus, we have two constraints by taking any two of them, e.g.,

$$\|h_{j2}^{s}\|^{2}h_{j1}^{T}Bh_{j1} - \|h_{j1}^{s}\|^{2}h_{j2}^{T}Bh_{j2} = 0,$$
(23)

$$\left(h_{j1}^{sT}h_{j2}^{s}\right)h_{j1}^{T}Bh_{j1} - \|h_{j1}^{s}\|^{2}h_{j2}^{T}Bh_{j1} = 0$$
<sup>(24)</sup>

with  $B \propto K^{-T}K^{-1}$ . Note that  $h_{jk}$  and  $h_{ik}^{s}$  are known but only *B* is unknown.

## 3.2. Estimating Parameters

As shown in Equations (23) and (24), we have two constraints from an  $H_j$ . Therefore, we can solve *B* and extract *K* in the same manner as the conventional method. On the other hand, a new approach is required for estimating the extrinsic parameters.

As soon as *K* is computed, a linear method can be employed to solve the extrinsic parameters. Stacking  $K^{-1}H_j$  and  $H_j^s$  for  $\forall j \in m$  horizontally, we have

$$\underbrace{\begin{bmatrix} K^{-1}H_1 & \cdots & K^{-1}H_m \end{bmatrix}}_{C} = \begin{bmatrix} R & t \end{bmatrix} \underbrace{\begin{bmatrix} \mu_1 H_1^s & \cdots & \mu_m H_m^s \\ 0 & 1 & \cdots & 0 & 1 \end{bmatrix}}_{D}$$
(25)

where  $\mu_{j} = \|K^{-1}h_{j1}\| / \|h_{j1}^{s}\|$  is a scaling factor.

Then, Equation (25) can be linearly solved by

$$\begin{bmatrix} R & t \end{bmatrix} = CD^T \left( DD^T \right)^{-1}.$$
 (26)

#### 3.3. Nonlinear Refinement

Nonlinear refinement must be applied to the linear estimation for more accuracy. The nonlinear optimization for the proposed method can be written by

$$\min_{K,R,t} \sum_{j \forall m} \sum_{i \forall n} \|m_{ij} - p(X_i, K^s, R_j^s, t_j^s, K, R, t, d)\|^2$$
s.t.  $R^T R = I$ ,
$$(27)$$

where  $p(X_i, K^s, R^s_j, t^s_j, K, R, t, d)$  is the projection of point  $X_i$  onto the image,  $d = [k_1, k_2]$  denotes the lens distortion coefficients and all the screen parameters  $K^s$ ,  $R^s_j$ , and  $t^s_j$  are known. In our implementation, this optimization is also solved by using the Levenberg- Marquardt algorithm [19,20].

Distortion coefficients are estimated based on Zhang's method [18] and included while minimizing Equation (27). For simplicity, only the first two coefficients of radial distortion  $k_1$  and  $k_2$  are considered, since the distortion function is mainly dominated by the radial components, especially the first term [2]. The relationship between the distortion-free pixel (x, y) and the distorted point ( $x_d$ ,  $y_d$ ) is presented by

$$x_d = x(1 + k_1 r^2 + k_2 r^4), (28)$$

$$y_d = y(1 + k_1 r^2 + k_2 r^4) \tag{29}$$

where  $r^2 = x^2 + y^2$ . Readers can refer to [3] for more details on lens distortion model and how to compensate lens distortion.

### 3.4. Summary

The procedure of the proposed method is very similar to the conventional one and includes the following steps:

- 1. Place the camera in front of the screen and adjust its position and orientation;
- 2. Fix the camera when the whole camera view is covered by the screen and it contains as much part of the screen as possible;
- 3. Take a few images of the screen while the virtual checkerboard is being transformed and displayed;
- 4. Detect the corner points in the images;
- 5. Estimate focal length  $f_x$  and  $f_y$ , principal point  $[u_0, v_0]$ , skewness *s*, rotation matrix *R* and translation vector *t* using the closed-form solution as stated in Section 3.2;
- 6. Refine intrinsic and extrinsic parameters, including lens distortion coefficients, by nonlinear optimization as described in Section 3.3.

## 4. Experiments and Discussion

To demonstrate the validity and robustness of the proposed method, experiments on both synthetic data and real data have been conducted.

## 4.1. Experiment Setup

Before starting the calibration, the camera to be calibrated needs to be setup to ensure that the whole camera view is covered by a screen. To start with, the screen is placed within the working distance of the camera and the camera is looking straight to the screen. Ideally, using a screen with appropriate size and let the optical axis of a camera cross vertically with the screen at the center, the aforementioned condition should be satisfied. This setup may not work for a real camera, since its principal point is usually not at the center of the image. Also a real camera has lens distortion. Therefore, we still need to manually adjust the orientation and position of the camera, and fix the camera until its entire image is covered by the screen.

Then, a set of parameters about orientation and position are generated. They are used to transform the virtual pattern in the experiments. The orientation of the pattern is generated as follows: the pattern is parallel to the screen at first; a rotation axis is randomly chosen from a uniform sphere; the pattern is then rotated around that axis with an arbitrary angle  $\theta$  between 40° and 50°. The reason for choosing  $\theta$  in that range is because it achieves the best performance according to the experimental results in [18]. The position of the pattern can be expressed by the 3D coordinate of its center point T = [x, y, z] in the screen's coordinates. In order to generate appropriate position for the pattern, following scheme is adopted. The pattern and the screen are initially on the same plane, and the center of the pattern coincides with the center of the screen. The pattern is then moved along the positive direction of *Z* axis. When the projection of the pattern on the screen is about 1/4 size of the screen, the value of *z* is fixed. The value of *x* and *y* are determined by randomly choosing points on the plane Z = z, within the screen's field of view. If given enough number ( $\geq 20$ ) of patterns, all the 2D projections of the corner points should scatter all over the image and the uniform distribution is achieved.

## 4.2. Experiment on Synthetic Images

In the computer simulation, a simulated camera is created with the following intrinsic parameters:  $f_x = 1417$ ,  $f_y = 1420$ ,  $u_0 = 942$ ,  $v_0 = 547$ , s = 0,  $k_1 = -0.0806$ ,  $k_2 = -0.0393$ . The screen which has  $1920 \times 1080$  resolution can be described using ideal pinhole model with 2500 (in pixels) focal length, and the principal point is located at the center of the screen. The virtual checkerboard contains  $16 \times 10 = 160$  corner points, and each square has 100 units per side. To investigate the performance of the proposed method regarding the noise level and the number of images of the calibration pattern, the following two experiments are designed and conducted. The method used for corner detection in the experiments is the method developed by Vezhnevets Vladimir, which is also integrated in OpenCV [21].

**Performance regarding the noise level.** To start with, virtual patterns with 20 different orientations and positions are synthesized. Then noisy images are created by adding Gaussian noise with a mean of  $\mu = 0$  and a standard deviation of  $\sigma$  to the projected image points. The noise level varies from  $\sigma = 0.1$  to  $\sigma = 1.5$ . For each noise level, our method is tested with 100 independent trials and assessed by comparing the results with the ground truth. Figure 3a,b show the relative error for focal length and absolute error for principal point respectively. As we can see in Figure 3, the average errors increases as the the noise level rises and the relationship between them is almost linear. When the noise level increases to  $\sigma = 0.5$ , which is larger than the normal noise in practical calibration [18], the relative errors in focal length  $f_x$  and  $f_y$  are less than 0.1%, and the absolute errors in principal point  $u_0$  and  $v_0$  are around 1 pixel.



Figure 3. Errors regarding the noise level of the image points. (a) Relative error for focal length;(b) Absolute error for principal point.



**Figure 4.** Errors regarding the number of the calibration pattern. (**a**) Relative error for focal length; (**b**) Absolute error for principal point.

**Performance regarding the number of images.** This experiment is designed to explore how the number of images of the calibration pattern impacts the performance of our method. Starting from two, we increase the number of images by one each time until it reaches twenty. For each number, Gaussian noise( $\mu = 0, \sigma = 0.5$ ) is first added to the images, calibration is then conducted with these independent images for 100 times. The errors are calculated based on the calibration results and ground truth data as in the previous experiment. The mean values of the errors are shown in Figure 4. The errors decrease

and tend to be stable as the number of image increases. Note that the errors decrease significantly when the number increases from 2 to 3.

# 4.3. Experiments on Real Images

To test our method on real images, we use a 24 inch LCD monitor to display the virtual pattern. Parameters of the screen and the virtual pattern are the same as in the computer simulation. The camera to be calibrated is the color camera of a Microsoft Kinect for Windows V2 sensor. As shown in Figure 5, the camera is fixed approximately 40 cm away from the screen using a tripod, looking straight to the screen, so that the whole camera view is covered by the screen. Ten independent trials are performed with images of  $1920 \times 1080$  resolution. In each trial, virtual pattern is transformed using parameters randomly chosen from the synthetic data and shown on the monitor. Meanwhile, the screen is captured by a real camera and 20 different images are used in each calibration. Figure 6a shows sample images captured in this experiment. The images are collected automatically by computer program, and the screen and the camera are fixed during the whole process. We use the same method as in the synthetic experiments for corner detection.



Figure 5. Setup of the real experiment.



**Figure 6.** Two calibration images captured in real experiment. (**a**) Image of a virtual checkerboard shown on screen; (**b**) Image of a physical checkerboard.

In comparison, we also calibrated the real camera using a physical checkerboard. The pattern is printed by a high-quality printer and attached to a glass board with guaranteed flatness. It contains the same number of squares as the virtual pattern, and each square is  $15 \text{ mm} \times 15 \text{ mm}$ . The camera is fixed by a tripod, and images are collected while the checkerboard is being manually moved. A sample

images used in this experiment is shown in Figure 6b. Ten independent trials are performed, with 20 images each time.

Explicit calibration experiments results are reported in Tables 1 and 2. For the first 10 lines in the tables, each line shows the result obtained in an independent trial, which are the 6 camera parameters and the root mean square error(RMSE). Here, the RMSE is defined as the root mean square distance between every detected corner point and the re-projected one using the estimated parameters. The mean and standard deviation values of the estimated parameters are listed in the last two lines. As we can see in Table 1, results obtained using the proposed method are very consistent with each other and the standard deviations for all parameters are pretty small, which suggests that our method is very robust. Contrarily speaking, performance of the conventional results are not as stable as the proposed one. Since we don't have ground truth data of the real world experiment, the camera parameters estimation result is evaluated based on re-projection error. With the proposed method and the conventional one, the mean value of the RMSE are 0.1855 and 0.2337 pixels, respectively. And the lowest RMSE, which is 0.1460, is achieved by the proposed method. We choose the best calibration results obtained by our method and the conventional method, and plot the localization errors of the control points in Figure 7. The results indicate that the proposed method outperforms the conventional one in terms of stability and accuracy in real world experiments.

Table 1. Calibration result for real images using the proposed method.

	$f_x$	$f_y$	u <sub>0</sub>	$v_0$	$k_1$	$k_2$	RMSE
Trial 1	1050.2120	1045.9939	957.1198	519.4579	0.0448	-0.0468	0.1502
Trial 2	1052.1709	1047.9542	957.1122	519.7247	0.0456	-0.0494	0.2021
Trial 3	1048.5039	1044.3648	956.7213	519.2291	0.0442	-0.0462	0.2061
Trial 4	1051.0054	1046.8187	956.8339	519.2194	0.0455	-0.0486	0.1756
Trial 5	1050.8918	1046.7329	956.8582	519.4178	0.0460	-0.0498	0.1944
Trial 6	1051.2977	1047.1457	956.8358	519.4241	0.0454	-0.0481	0.1460
Trial 7	1050.4691	1046.3180	956.5077	519.6354	0.0446	-0.0467	0.1699
Trial 8	1052.8643	1048.7560	956.4323	519.5267	0.0452	-0.0473	0.2077
Trial 9	1051.0076	1046.8497	956.9606	519.6325	0.0461	-0.0489	0.1952
Trial 10	1049.4690	1045.3789	956.4628	519.2602	0.0460	-0.0494	0.2076
Mean	1050.7892	1046.6313	956.7845	519.4528	0.0453	-0.0481	0.1855
Deviation	1.2463	1.2397	0.2515	0.1791	0.0006	0.0013	0.0236

Table 2. Calibration result for real images using the conventional method.

	$f_x$	$f_y$	$u_0$	$v_0$	$k_1$	<i>k</i> <sub>2</sub>	RMSE
Trial 1	1048.0347	1044.0247	956.8945	519.3556	0.0438	-0.0464	0.2595
Trial 2	1047.9891	1043.7756	956.6410	519.7846	0.0458	-0.0485	0.2153
Trial 3	1051.6414	1047.3967	957.3807	519.6939	0.0458	-0.0486	0.2029
Trial 4	1052.1863	1048.0365	957.2387	519.3653	0.0454	-0.0470	0.2948
Trial 5	1050.3806	1046.1871	956.9527	519.0593	0.0446	-0.0452	0.2469
Trial 6	1049.6929	1045.5486	956.8276	519.5737	0.0451	-0.0475	0.2210
Trial 7	1048.9989	1044.8639	956.8082	519.3750	0.0449	-0.0465	0.1747
Trial 8	1050.1785	1046.0461	956.7260	519.5743	0.0439	-0.0457	0.2672
Trial 9	1050.3922	1046.2240	956.5963	519.5850	0.0437	-0.0445	0.1787
Trial 10	1051.6436	1047.4263	956.8238	519.8481	0.0450	-0.0459	0.2757
Mean	1050.1138	1045.9530	956.8889	519.5215	0.0448	-0.0466	0.2337
Deviation	1.4674	1.4353	0.2487	0.2362	0.0008	0.0014	0.0414



**Figure 7.** Scatter plots for the RMSE between the detected corner points and the re-projected ones with the estimated calibration parameters. (**a**) Localization errors by the proposed method; (**b**) Localization errors by the conventional method.

# 4.4. Discussion

The above experiments show not only the practicality but also the advantage of the proposed method. In conventional calibration method, a key step is to capture images while manually moving a physical calibration pattern. Usually, this step takes as long as several minutes. In contrast, our method takes much less time to prepare calibration pattern and collect high quality data, and the whole procedure is done fully automatically within one minute.

The use of virtual pattern affects the calibration result in the following aspects. First, virtual pattern is transformed by computer program so that all the control points are uniformly distributed in the image. Well distributed points usually lead to more stable and accurate calibration result. Second, since the screen is fixed in the calibration, image blur caused by motion can be eliminated, therefore, control points can be more precisely localized. Otherwise, in a blurry image which is taken by a moving camera like Figure 8, the observed feature location in the image may deviate from the actual feature location. Even though the checkerboard patten can be detected by some algorithms (e.g., OpenCV's checkerboard detection algorithm [21]), uncertainty in the localizations of the control points yields incorrect correspondences which lead to performance degradation of the calibration.



Figure 8. A blurry image captured in real experiment.

However, the proposed method also shows some limitations. An essential requirement of this method is that the entire camera view has to be covered by a screen. In some cases, it is difficult to satisfy the above requirement. For a camera with large working distance or wide field of view, it is necessary to use a large size screen, e.g., flat screen TV, to cover the entire image of the camera. However, screen size cannot be increased without limitation, our method may not be applicable if the camera has very large working distance or very wide field of view. The proposed method also does not work in some certain applications, such as high precision visual measurement, where the camera to be calibrated has very short working distance or very high resolution. In this case, the resolution of the camera is usually higher than that of the screen. Hence the image of a screen is discretized, and corner point detection and localization can be a problem. Although the effect of discretization can be reduced by using high resolution screen, it still affects the accuracy of calibration unless it is completely eliminated.

## 5. Conclusions

The conventional calibration technique using a 2D planar object is widely used due to its ease of use. Although many efforts have been focused on making the whole calibration procedure as automatic as possible, there is still a manual part at the capture step which takes a lot of time and makes the result unstable. In this paper, we proposed a full-automatic method for camera calibration to resolve the issues brought about by manual operations. Different from the conventional method, we use a virtual pattern which is transformed in the virtual world coordinates and projected on a fixed screen. The pattern shown on the screen is then captured by a fixed camera. Calibration is performed by using point correspondences between the virtual 3D points and their 2D projections, and the solution to camera parameters estimation is very similar to the conventional method.

Owing to the use of virtual pattern, there is no need to manually adjust the position and orientation of the checkerboard during calibration. Moreover, the virtual pattern can be actively displayed on the screen so that all corner points are uniformly distributed. Once the camera and the screen are set up, they are fixed during the whole calibration process. Thus, the proposed method can be fully automatic and the problems caused by manual operation are resolved without loss of usability. Experimental results show that our method is more robust and accurate than the conventional method.

Acknowledgments: This work has been supported by National Natural Science Foundation of China (Grant No. 61573134, 61573135), National Key Technology Support Program (Grant No. 2015BAF11B01), National Key Scientific Instrument and Equipment Development Project of China (Grant No. 2013YQ140517), Key Research and Development Project of Science and Technology Plan of Hunan Province(Grant No. 2015GK3008), Key Project of Science and Technology Plan of Guangdong Province(Grant No. 2013B011301014).

**Author Contributions:** The paper was a collaborative effort between the authors. Lei Tan, Yaonan Wang and Hongshan Yu proposed the idea of the paper. Lei Tan and Jiang Zhu implemented the algorithm, designed and performed the experiments. Lei Tan and Hongshan Yu analyzed the experimental results and prepared the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

#### References

- 1. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 2000, 22, 1330–1334.
- 2. Tsai, R.Y. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE J. Robot. Autom.* **1987**, *3*, 323–344.
- Heikkila, J.; Silven, O. A four-step camera calibration procedure with implicit image correction. In Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PuertoRico, 17–19 June 1997; pp. 1106–1112.
- 4. Chen, Q.; Wu, H.; Wada, T. Camera calibration with two arbitrary coplanar circles. In *Computer Vision-ECCV* 2004; Springer: New York, NY, USA, 2004; pp. 521–532.

- Bergamasco, F.; Cosmo, L.; Albarelli, A.; Torsello, A. Camera calibration from coplanar circles. In Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 2137–2142.
- Agrawal, M.; Davis, L.S. Camera calibration using spheres: A semi-definite programming approach. In Proceedings of the 2003 Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 1–8.
- 7. Wong, K.Y.K.; Zhang, G.; Member, S.; Chen, Z. Calibration Using Spheres. *Image* 2011, 20, 305–316.
- 8. Caprile, B.; Torre, V. Using vanishing points for camera calibration. Int. J. Comput. Vis. 1990, 4, 127–139.
- Radu, O.; Joaquim, S.; Mihaela, G.; Bogdan, O. Camera calibration using two or three vanishing points. In Proceedings of the 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, Poland, 9–12 September 2012; pp. 123–130.
- Li, B.; Heng, L.; Koser, K.; Pollefeys, M. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 1301–1307.
- Moreno, D.; Taubin, G. Simple, accurate, and robust projector-camera calibration. In Proceedings of the 2012 2nd International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Zurich, Switzerland, 13–15 October 2012; pp. 464–471.
- 12. Raposo, C.; Barreto, J.P.; Nunes, U. Fast and accurate calibration of a kinect sensor. In Proceedings of the 2013 International Conference on 3DTV-Conference, Seattle, WA, USA, 29 June–1 July 2013; pp. 342–349.
- Rufli, M.; Scaramuzza, D.; Siegwart, R. Automatic detection of checkerboards on blurred and distorted images. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3121–3126.
- 14. Donné, S.; De Vylder, J.; Goossens, B.; Philips, W. MATE: Machine Learning for Adaptive Calibration Template Detection. *Sensors* **2016**, *16*, 1858.
- 15. Pilett, J.; Geiger, A.; Lagger, P.; Lepetit, V.; Fua, P. An all-in-one solution to geometric and photometric calibration. In Proceedings of the 2006 Fifth IEEE/ACM International Symposium on Mixed and Augmented Reality, Santa Barbar, CA, USA, 22–25 October 2006; pp. 69–78.
- 16. Atcheson, B.; Heide, F.; Heidrich, W. CALTag: High Precision Fiducial Markers for Camera Calibration. *Vis. Model. Vis.* **2010**, *10*, 41–48.
- 17. Oyamada, Y. Single Camera Calibration using partially visible calibration objects based on Random Dots Marker Tracking Algorithm. In Proceedings of the IEEE ISMAR 2012 Workshop on Tracking Methods and Applications (TMA), Atlanta, GA, USA, 5–8 November 2012.
- 18. Zhang, Z. *A Flexible New Technique for Camera Calibration*; Technical Report MSR-TR-98-71; Microsoft Research: Redmond, WA, USA, 1998.
- 19. Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441.
- 20. Levenberg, K. A method for the solution of certain problems in least squares. Q. Appl. Math. 1944, 2, 164–168.
- 21. Vezhnevets, V. OpenCV Calibration Object Detection, Part of the Free Open-Source OpenCV Image Processing Library. Available online: http://graphicon.ru/oldgr/en/research/calibration/opencv.html (accessed on 20 December 2016).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).