Effect of Checkerboard on the Accuracy of Camera Calibration

Shengju Yu, Ran Zhu, Li Yu(✉), and Wei Ai

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
{shengju_yu,hustlyu}@hust.edu.cn

Abstract. Camera calibration is an important step in many fields, and its accuracy is influenced by many factors. In this paper, the influence of different numbers and sizes of squares of the checkerboard on calibration is studied, and the errors caused by this are analyzed. We propose *SBI* algorithm to improve the calibration accuracy. In order to better judge the quality of calibration, we map the texture map onto the depth map by coordinate transformation to achieve the effect of registration. Besides, we use the Canny operator to extract the depth map edges and the mapped texture map edges, and utilize their matching results to evaluate the calibration accuracy. Experimental results show that our registration effect is the best and the number of mismatched pixels on the edges is the least, which indicates our calibration accuracy is the highest.

Keywords: Camera calibration · Image registration · Edge matching

1 Introduction

The general principle of camera calibration lies in finding the correspondence between a sufficiently large number of known 3D points in world coordinate and their projections in 2D image [1]. A lot of researchers have proposed different calibration methods. According to the dimensions of calibration objects, these methods can be divided into three categories, namely 1D, 2D, 3D objects. 1D objects, for example a wand with multiple collinear points, are the easiest to construct. They are often used in the calibration of multiple cameras since all cameras can see the same calibration object at the same time [2,3]. However, due to the lower number of calibration points, these methods provide the less accurate results. Methods based on 2D objects, for instance planar boards consisting of checkerboards [4] or circles [5], are the most popular because of the good trade-off between accuracy and simplicity. 3D objects, such as an object consisting of two or three planes orthogonal to each other, can provide more accurate results. But these approaches based on 3D objects require expensive calibration devices and elaborate setups. Therefore, this paper will focus on 2D plane-based calibration, more specifically the method proposed by Zhang [4].

R. Hong et al. (Eds.): PCM 2018, LNCS 11166, pp. 619–629, 2018.
https://doi.org/10.1007/978-3-030-00764-5_57

Zhang [4] proposed a flexible method to easily calibrate the camera, which only requires the camera to observe a planar pattern from several different directions. Not only the camera but also the planar pattern can move at an unknown scale. Although this technology has gained considerable flexibility, there is still some work to do, for instance, reducing the location error of the detected feature points, exploring the influence of the size and number of the pattern on calibration accuracy. In addition, different sets of images can lead to different calibration results even with the same pattern, and the differences between the results may be very large. So, it's essential to study how to select better images that are conducive to calibration. [4,6,7] calibrated the camera with 10, 5, 30 images respectively in their experiments. Therefore, it's also necessary to further explore the influence of the number of planar pattern images on calibration results.

We calibrate the depth camera and texture camera of kinect v1 [8] separately. And the main contents of this paper can be summarized as follows: (1) In order to select better images that are favorable for camera calibration, we propose SBI algorithm. (2) We design a series of comparative experiments to explore the influence of the size and number of black-and-white squares of the checkerboard on calibration. (3) In order to judge the quality of calibration results, we map the texture map onto the depth map by coordinate transformation to achieve the registration effect. (4) We further use the Canny operator to extract the depth map edges and the mapped texture map edges, and take advantage of [9] to match these edges. The matching results are used to evaluate the calibration accuracy.

This paper is organized as follows. Section 2 briefly describes the calibration principle. Section 3 states the proposed SBI algorithm. Section 4 describes how to register a depth camera and a texture camera. Experimental results and error analysis are reported in Sect. 5. Finally, Sect. 6 concludes the work.

2 Calibration Principle

We concentrate on the well-known pinhole camera model, which is widely used in many fields. It assumes that the camera performs a perfect perspective projection transformation from the 3D scene coordinates (x, y, z) to image plane coordinates (u, v), as shown in Fig. 1. Four coordinates are defined as follows: $O_wX_wY_wZ_w$ is the world coordinate system. $O_cX_cY_cZ_c$ is the camera coordinate system, and the projection center is the point O_c. O_ixy is the image coordinate system in millimeters, and the intersection point of the image plane and the optical axis of the camera is defined as the origin O_i. O_puv is the pixel coordinate system and the u and v axes are parallel to the x and y axes, respectively. For modern cameras, the skew coefficient of u and v axes is often equal to zero, that is, they are completely vertical. The perspective projection transformation from the world coordinate system to the pixel coordinate system can be expressed as:

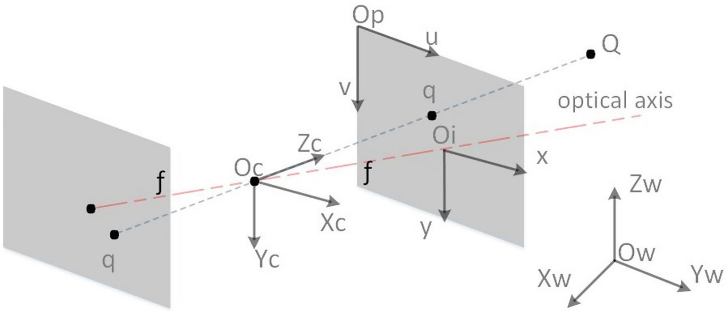


Fig. 1. Pinhole camera model

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

Where {R, T} is a rigid transformation that maps points in the world coordinate system onto the camera coordinate system. $T = [t_1, t_2, t_3]$ describes the translation between the two frames. $R = [r_{11}\ r_{12}\ r_{13}; r_{21}\ r_{22}\ r_{23}; r_{31}\ r_{32}\ r_{33}]$ is a 3 by 3 orthonormal rotation matrix and can be defined by three angles α, β, γ. If R is known, these angles can be computed using the following decomposition:

$$\begin{aligned} \alpha &= \sin^{-1}(r_{31}) \\ \beta &= \text{atan2}(-\frac{r_{32}}{\cos\alpha}, \frac{r_{33}}{\cos\alpha}) \\ \gamma &= \text{atan2}(-\frac{r_{21}}{\cos\alpha}, \frac{r_{11}}{\cos\alpha}) \end{aligned} \tag{2}$$

Without loss of generality, the model plane can be set on $z = 0$ of the world coordinate system. Equation (1) can be simplified as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}, A = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The r_i represents the i^{th} column of R. The parameters f_x, f_y, u_0 and v_0 are called intrinsic parameters and the t_1, t_2, t_3, α, β and γ are extrinsic parameters.

One model point Q and its image point q are related by a homography H, $H = [h_1\ h_2\ h_3] = A[r_1\ r_2\ T]$. r_1 and r_2 are orthonormal, so:

$$\begin{aligned} h_1^T A^{-T} A^{-1} h_2 &= 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2 \end{aligned} \tag{4}$$

Given one homography, two constraints on the intrinsic parameters can be obtained. In addition, a homography has eight degrees of freedom and there are six extrinsic parameters (three for R, three for T). Therefore, we need at least two images to solve the intrinsic and extrinsic parameters of camera.

If n images of a model plane are obtained and there are m feature points on the model plane. The maximum likelihood estimate can be achieved by minimizing the following function:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|q_{ij}-\hat{q}(A,R_i,T_i,Q_j)\|^2 \tag{5}$$

where $\hat{q}(A,R_i,T_i,Q_j)$ is the projection of 3D point Q_j in image i. This is a nonlinear minimization problem and can be solved with the Levenberg-Marquardt Algorithm as implemented in Minpack [10].

The measurement of calibration accuracy is the reprojection error $Ereproj$.

$$Ereproj=\frac{1}{S}\sum_{i=1}^{S}\sqrt{(u_i-u_i^{reproj})^2+(v_i-v_i^{reproj})^2} \tag{6}$$

where S is the number of feature points, $(u_i,\ v_i)$ is real pixel coordinate and $(u_i^{reproj},\ v_i^{reproj})$ is the reprojection coordinate.

3 Algorithm for Improving the Calibration Accuracy

Different sets of images can lead to different calibration results even with the same calibration pattern, the results are unstable and the differences between calibration results may be very large. Thus, before the camera is calibrated, it's necessary to select better images that are beneficial for calibration.

Let I denote a large set of images of the calibration pattern in different orientations, in all of which the features are correctly detected. Assuming that the size of I is N, then $I=\{image_1,image_2,\ldots,image_N\}$. Let S represent a subset of I and each element has size m. That is, $S=\{s_1,s_2,\ldots,s_n\}$, $s_i\in I^m$.

Algorithm 1. The proposed *SBI* algorithm for improving calibration accuracy.

```
1: Acquire a set $I_{orig}$ of original images with different calibration object views.

2: Those images with successful feature detection and large difference of the orientation of calibration object are chosen as a set $I$, $I \in I_{orig}$ and having length $N$.
3: Check whether $m \leq N$, and if the expression is true, proceed.
4: Select $m$ better images that favor camera calibration from set $I$:
Add a parameter index for each image. The default value of index is 0. Then
for i=1 to $N-m+1$ do
 for j=i+1 to $N-m+2$ do
  for k=j+1 to $N-m+3$ do
```

```
    - Utilize set $B = \{I_i, I_j, I_k\}$ to calibrate the camera. When $E_s$ is the smallest, calibration results and the path of images $I_i$, $I_j$, $I_k$ are preserved. $E_s$=$E_{reproj}$+$E_f$. $E_f$=$(((f_x - f_{dx})/f_{dx})^2 + ((f_y - f_{dy})/f_{dy})^2)^{1/2}$. The values of the parameter $index$ of images $I_i$, $I_j$, $I_k$ are set to 1, which indicates that they are better images.
5: In order to explore the effect of the number of images on calibration results, a total of $G$ images are sequentially selected.
6: Check whether $G \leq N$, if the expression is false, terminate. Otherwise:
for i=1 to $G - m$ do
  for g=1 to $N$ do
    if $index_g = 0$
    - Utilize set $B$ and $I_g$ to calibrate camera. $I_g$ is reserved and the value of the parameter $index_g$ is set to 1 when $E_s$ is the smallest.
    When the loop of $g$ ends, add $I_g$ to the set $B$ to form a new set $B$.
7: $G$ better images are selected and camera calibration results from $m$ to $G$ images are also preserved.
```

4 Registration of Depth Camera and Texture Camera

The principle of registration is shown in Fig. 2. Q_{c_depth} is the spatial coordinate of Q point in the depth camera coordinate system. Q_{c_rgb} is the spatial coordinate of Q point in the texture camera coordinate system. $Q_{\_depth}$ and $Q_{\_rgb}$ are the projection coordinate of the point on image plane, respectively. A_{depth} is the intrinsic matrix of the depth camera. We can get:

$$Q_{\_depth} = A_{depth} \cdot Q_{c\_depth} \tag{7}$$

Depth camera and texture camera can be linked by rotation and translation matrix $({}^{D}R_{rgb}, {}^{D}T_{rgb})$.

$$Q_{c\_rgb} = {}^{D}R_{rgb} \cdot Q_{c\_depth} + {}^{D}T_{rgb} \tag{8}$$

The $Q_{\_rgb}$ coordinate corresponding to the $Q_{\_depth}$ point can be obtained by the following formula:

$$Q_{\_rgb} = A_{rgb} \cdot Q_{c\_rgb} \tag{9}$$

Therefore, in order to determine the relationship between $Q_{\_depth}$ and $Q_{\_rgb}$, it's necessary to first solve ${}^{D}R_{rgb}$ and ${}^{D}T_{rgb}$. Using the extrinsic matrix, we can get:

$$\begin{aligned} Q_{c\_depth} &= R_{depth} \cdot Q + T_{depth} \\ Q_{c\_rgb} &= R_{rgb} \cdot Q + T_{rgb} \end{aligned} \tag{10}$$

So,

$$Q_{c\_rgb} = R_{rgb} \cdot R_{depth}^{-1} \cdot Q_{c\_depth} + T_{rgb} - R_{rgb} \cdot R_{depth}^{-1} \cdot T_{depth} \tag{11}$$

Finally,

$$
\begin{aligned}
{}^{D}R_{rgb} &= R_{rgb} \cdot R_{depth}^{-1} \\
{}^{D}T_{rgb} &= T_{rgb} - {}^{D}R_{rgb} \cdot T_{depth}
\end{aligned}
\tag{12}
$$

For each pixel $Q_{\_depth}$ in the depth map, the corresponding coordinate $Q_{\_rgb}$ in the texture map can be obtained through the above formulas, and the color information can be assigned to the corresponding depth map to achieve the registration effect.

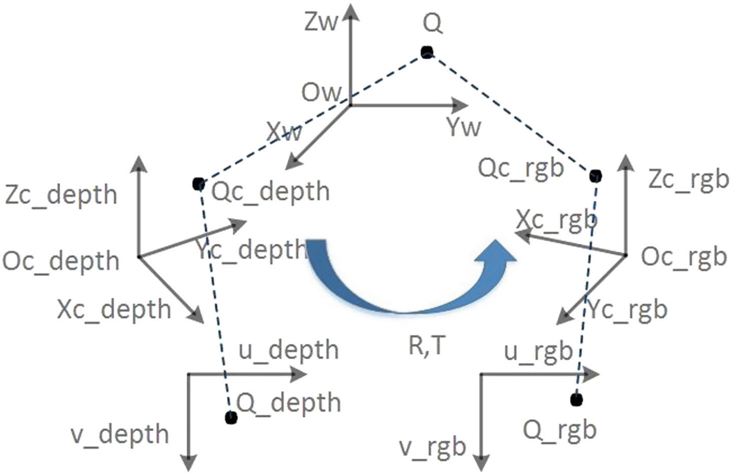
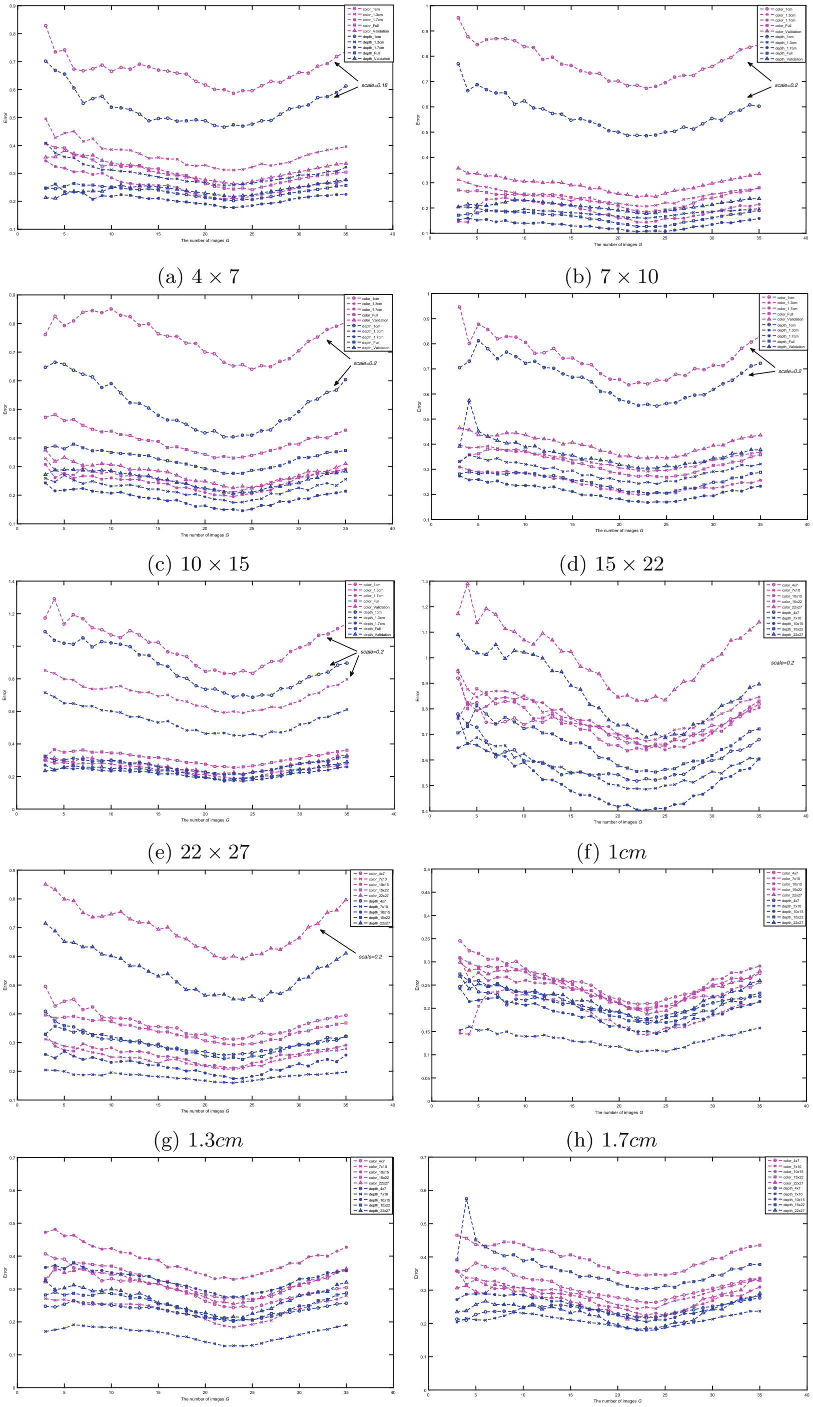


Fig. 2. Registration principle

5 Experimental Results

We use a printed flat plate with checkerboard pattern as the calibration object. The specification of checkerboard is $a \times b$. a and b indicate the number of squares in the vertical and horizontal directions, respectively. In order to explore the influence of the size of squares on calibration accuracy, we design five sets of contrast experiments, namely, 4×7, 7×10, 10×15, 15×22 and 22×27, as shown in Table 1. The checkerboard in $Full$ is as big as possible. $Validation$ is to validate our conjecture that the size of squares is not the bigger the better. At the same time, we also carry out 1.0 cm, 1.3 cm and 1.7 cm three groups of comparative experiments to study the effect of the number of squares on calibration accuracy. $Full$ and $Validation$ are also used to verify the effect of the number of images on calibration accuracy. We calibrate the depth camera and texture camera of kinect v1 separately. So, there are 50 groups of experiments in total. Theoretically, we can utilize only two images to calibrate the camera,



(a) 4 × 7

(b) 7 × 10

(c) 10 × 15

(d) 15 × 22

(e) 22 × 27

(f) 1cm

(g) 1.3cm

(h) 1.7cm

(i) *Full*

(j) *Validation*

Fig. 3. Calibration *Error* of checkerboard with different specifications

Table 1. Different specifications of checkerboard

$a \times b$					
Size	4×7	7×10	10×15	15×22	22×27
1.0 cm	1.0 cm	1.0 cm	1.0 cm	1.0 cm	1.0 cm
1.3 cm	1.3 cm	1.3 cm	1.3 cm	1.3 cm	1.3 cm
1.7 cm	1.7 cm	1.7 cm	1.7 cm	1.7 cm	1.7 cm
Full	4.6 cm	3.2 cm	2.4 cm	2.0 cm	1.9 cm
Validation	5.0 cm	2.9 cm	2.1 cm	2.2 cm	1.6 cm

but the error is too large. Therefore, we choose three images as a starting point. And $N = 100$ are set as default value. In addition, in order to further explore the influence of the number of images on calibration results, we select 3 to 35 better images in turn to calibrate the camera and observe the error $Error$ for each group.

$$Error = E_s + E_o \tag{13}$$

where $E_o = (((u_0 - u_{d0})/u_{d0})^2 + ((v_0 - v_{d0})/v_{d0})^2)^{1/2}$ indicates the degree of deviation of optical center. E_s contains the deviation of the reprojection and the focal length. f_x, f_y, u_0 and v_0 are obtained through experiments. f_{dx}, f_{dy}, u_{d0} and v_{d0} are derived from the FOV (Field of View) of the camera, which are used as default values. The $Error$ curves of each experiment are shown in Fig. 3.

Figure 3(a)–(e) show that when the number of black-and-white squares of the checkerboard is fixed, the error of 1.7 cm is smaller than others. The calibration error of bigger checkerboard in $Validation$ is larger than that in $Full$, which confirms our conjecture that the checkerboard is not the bigger the better. From Fig. 3(f)–(h), we can find that when the size of squares is fixed, the calibration result of 7×10 is better than others. So, the dense checkerboard is not good for camera calibration. Figure 3(i) and (j) indicate that it is better to use 20–26 images to calibrate the camera even though the number and size of squares are changed, and too many images will lead to the accumulation of errors and increase error probability. On the one hand, there is an error in the determination of the coordinates of corner points, and it is hard to assert that this error is consistent with Gaussian distribution. On the other hand, the nonlinear iterative optimization algorithm used in calibration can not always guarantee the optimal solution, and more images may increase the possibility of the algorithm

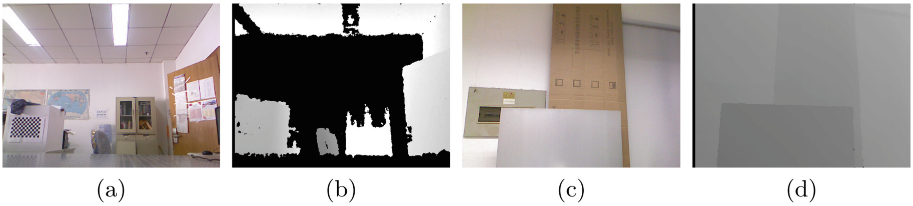
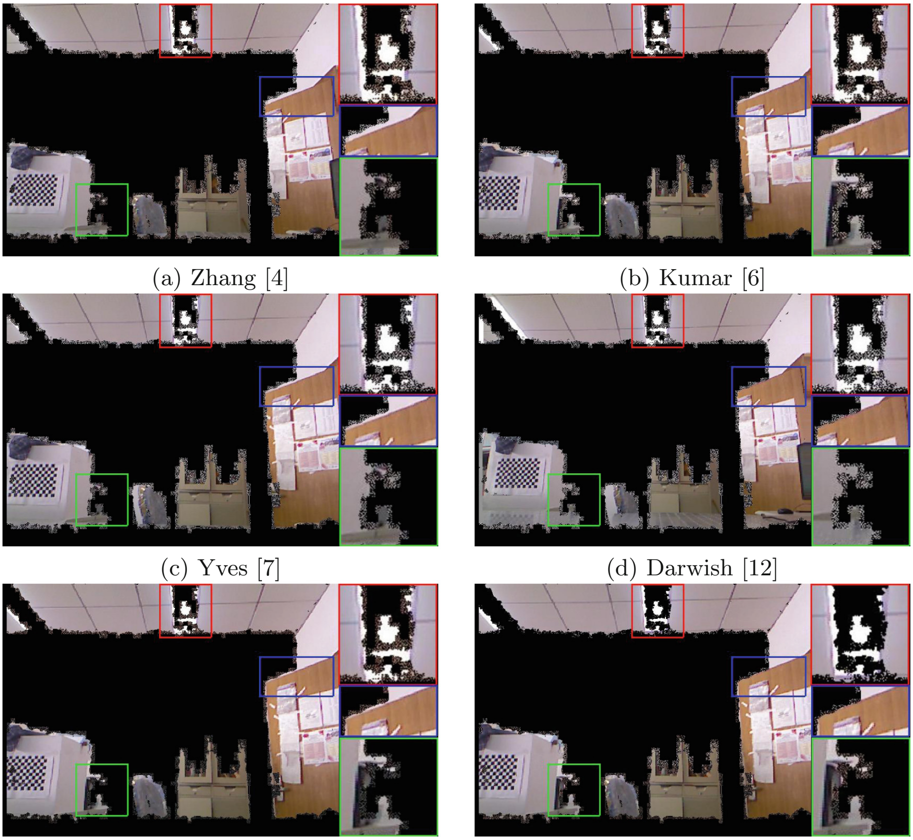


Fig. 4. Experimental images for registration and edge matching



(a) Zhang [4]

(b) Kumar [6]

(c) Yves [7]

(d) Darwish [12]

(e) Herrera [11]

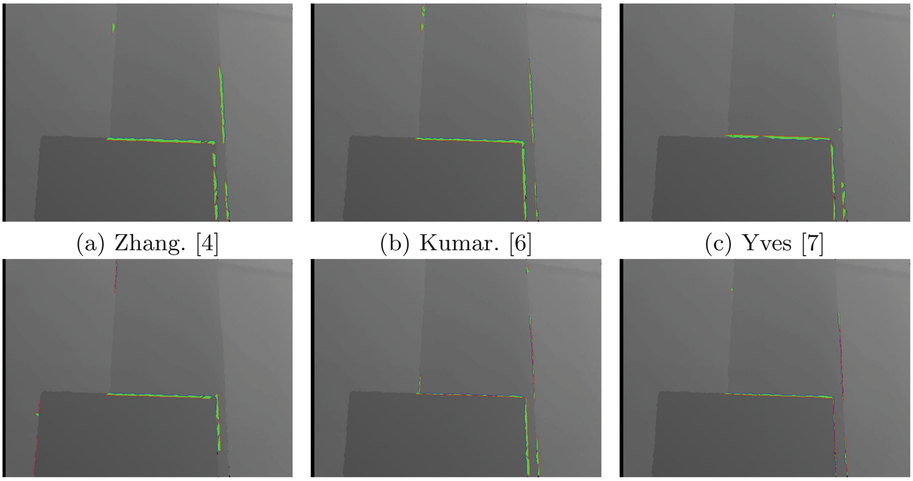
(f) *Ours*

Fig. 5. Registration results

falling into local optimal. Therefore, we choose 7 × 10 & 1.7 cm checkerboard and its better first 23 images to calibrate the kinect v1.

According to the calculated parameter values of the camera, we map the texture map onto the depth map by coordinate transformation to judge the quality of calibration results, as shown in Fig. 5. Comparing Fig. 5 with Fig. 4(a), (b), we can see that our registration result is better than others, which shows that our calibration accuracy for depth camera and texture camera is very high.

To further test the calibration results, we use the Canny operator to extract the depth map edges and the registered texture map edges, and then match these edges. The matching results are shown in Fig. 6. The blue line represents the pixels that are incorrectly matched. The red line and green line indicate the edges of the depth map and registered texture map near the mismatched pixels, respectively. One can see that the number of our mismatched pixels is the least.



(a) Zhang. [4]

(b) Kumar. [6]

(c) Yves [7]

(d) Darwish. [12]

(e) Herrera [11]

(f) *Ours*

Fig. 6. Edge matching results (Color figure online)

6 Conclusion

In this paper, the influence of different sizes and numbers of the squares on calibration results is studied. The size of squares is not the bigger the better. The dense checkerboard can't guarantee the high accuracy of calibration results. Too many images used for calibration result in the accumulation of errors, and we suggest using 20–26 images to calibrate the camera. We propose *SBI* algorithm to improve the calibration accuracy. In addition, in order to judge the quality of calibration, we map the texture map onto the depth map by coordinate transformation to register. We also use the Canny operator to extract the depth map edges and the registered texture map edges and then match them, which is more conducive to compare the effect of registration and the accuracy of calibration.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (NSFC) (No. 61231010), National High Technology Research and Development Program (No. 2015AA015901).

References

1. Carsten, S., Markus, U., Christian, W.: Machine Vision Algorithms and Applications. Trans Hsinghua University Publishing, Beijing (2008)
2. Zhang, Z.: Camera calibration with one-dimensional objects. IEEE Trans. Pattern Anal. Mach. Intell. **26**(7), 892–899 (2004)
3. Svoboda, T.: A convenient multicamera self-calibration for virtual environments. Presence **14**(4), 407–422 (2005). MIT Press
4. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)

5. Jiang, G., Quan, L.: Detection of concentric circles for camera calibration. In: International Conference on Computer Vision, pp. 333–340 (2005)
6. Kumar, A.: On the equivalence of moving entrance pupil and radial distortion for camera calibration. In: International Conference on Computer Vision, pp. 2345–2353 (2015)
7. Le Sant, Y.: Multi-camera calibration for 3DBOS. In: 17th International Symposium on Applications of Laser Techniques to Fluid Mechanics (2014)
8. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE Multimed. **19**(2), 4–10 (2012)
9. Xiang, S., Yu, L., Chen, C.W.: No-reference depth assessment based on edge misalignment errors for T+ D images. IEEE Trans Image Process. **25**(3), 1479–1494 (2016)
10. Moré, J.J.: The Levenberg-Marquardt algorithm: implementation and theory. In: Watson, G.A. (ed.) Numerical Analysis. LNM, vol. 630, pp. 105–116. Springer, Heidelberg (1978). https://doi.org/10.1007/BFb0067700
11. Herrera, D., et al.: Joint depth and color camera calibration with distortion correction. IEEE Trans. Pattern Anal. Mach. Intell. **34**(10), 2058–2064 (2012)
12. Darwish, W., et al.: A new calibration method for commercial RGB-D sensors. Sensors **17**(6), 1204 (2017)