Pattern Feature Detection for Camera Calibration Using Circular Sample

Dong-Won Shin and Yo-Sung Ho(✉)

Gwangju Institute of Science and Technology (GIST),
123 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
{dongwonshin,hoyo}@gist.ac.kr

Abstract. Camera calibration is a process to find camera parameters. Camera parameter consists of intrinsic and extrinsic configuration and it is important to deal with the three-dimensional (3-D) geometry of the cameras and 3-D scene. However, camera calibration is quite annoying process when the number of cameras and images increase because it is operated by hand to indicate exact points. In order to eliminate the inconvenience of a manual manipulation, we propose a new pattern feature detection algorithm. The proposed method employs the Harris corner detector to find the candidate for the pattern feature points in images. Among them, we extract valid pattern feature points by using a circular sample. Test results show that this algorithm can provide reasonable camera parameters compared to camera parameters using the Matlab calibration toolbox by hand but eliminated a burden of manual operation.

Keywords: Camera calibration · Harris corner · Feature detection

1 Introduction

The real world consists of a three-dimensional (3-D) space. However, when you capture 3-D space by a camera, it would be represented on a two-dimensional (2-D) plane. If you want to know where a point in 3-D space is placed on a 2-D plane, you should know a geometric relationship between them: a rotation and position of the camera (extrinsic parameters), a relationship between a lens and an image sensor in a camera (intrinsic parameters). In addition to the fact that we can know the relation between 2-D plane and 3-D space, a geometric relation among multi-view cameras also can be found. We call this process as the camera calibration, which find intrinsic and extrinsic parameters [1].

The camera calibration step is necessary when you deal with a geometric meaning among multiple cameras such as 3-D warping or image rectification. 3-D warping is a process to forward depth information from one view image to another view image by using camera parameters [2]. Figure 1 shows the 3-D warping process. We can forward a point p_l located in the left image to a point p_r located in the right image via a point p_w placed in the world coordinate system. It means that we can know where the point p_l is located in right image as the point p_r and vice versa.

Next, the image rectification is to align epipolar lines of the images as parallel each other [3]. It is usually performed as a pre-processing step of stereo matching. In this

Y.-S. Ho et al. (Eds.): PCM 2015, Part II, LNCS 9315, pp. 608–615, 2015.
DOI: 10.1007/978-3-319-24078-7_62

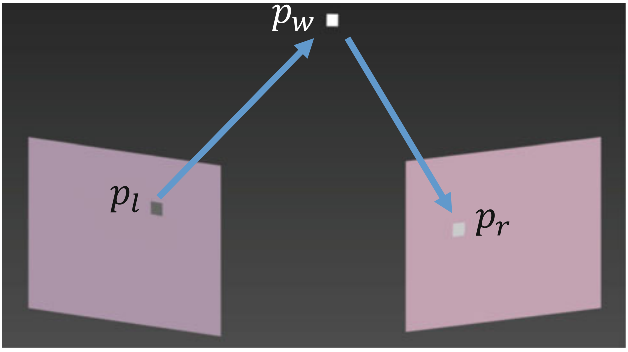


Fig. 1. 3-D warping

ideal camera setup, all cameras have the same intrinsic and extrinsic parameters except for the translation vector. In case of the translation vector, it is set to be had the same distance to the adjacent camera along the horizontal direction. Figure 2 shows the image rectification process.

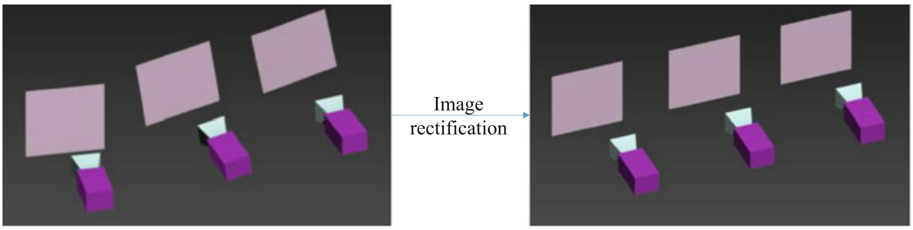


Fig. 2. Image rectification

Although the camera calibration is a necessary process in multi-view geometry, it is very time-consuming since a human needs to mark the points in every image by their hand. For example, in case of the camera calibration using a 4 × 6 planar grid pattern, we need to select four corner points of a pattern rectangle in a single image. If we have 5 cameras and capture 10 images at each camera, we need to choose 5 × 10 × 4 = 200 points by our hands. It is very inefficient and would be a big burden when the number of cameras and images increases. In order to relieve this inconvenience, we studied on an automatic pattern feature detection for camera calibration.

Recently, many researches in this area have presented lots of methods. Arturo *et al.* employed Hough transform to detect lines in a pattern image, and then choosing the intersections between horizontal and vertical lines as a pattern feature point [4]. J. Chu *et al.* proposed a chessboard corner detector based on image physical coordinates with subpixel precision, which can obtain the subpixel in last one step and decrease the computation complexity [5]. Lastly, M. Fiala *et al.* designed a discriminable marker-based calibration system. This calibration method employs an array of fiducial marker to find the correspondences among images [6].

In order to improve the efficiency of the camera calibration without a manual manipulation, we propose a circular sample with Harris corner detector. Our proposed algorithm extracts corners in an image by using Harris corner detector. Among the candidates, we obtain true pattern feature points by using a circular sample which inspect pixels around extracted Harris corner points.

The paper is organized as follows. In Sect. 2, we describe a corner detection. In Sect. 3, we explain a circular sample to find true pattern feature points. In Sect. 4, we show test results of the system running under various conditions and comparisons with the widely available Caltech Matlab camera calibration toolbox in terms of accuracy. Finally, in Sect. 5, we draw conclusions and offer discussions.

2 Corner Detection

A corner is usually defined as the intersection of two edges. It has a well-defined position and can be robustly detected. On the basis of the concept of the corner, we define a pattern feature point in this research. The pattern feature point is a point that we need to find in every image to perform the camera calibration. It is located in an interior of a planar calibration pattern and the corners include the pattern feature points. Figure 3 shows the example of corners and the pattern feature points. In Fig. 3, we represent the corners as red points and pattern feature points as blue points. As we can see in this figure, corners can be extracted not only in the planar grid calibration panel but also in the persons standing on the both sides of the panel.

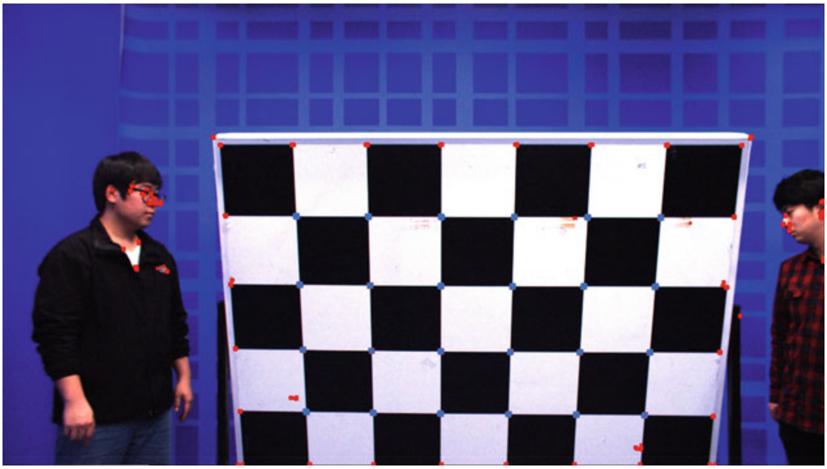


Fig. 3. Corners and pattern feature points

In the proposed method, first we extract the corners by using Harris corner detector [7]. Figure 4 shows the principle of Harris corner detector.

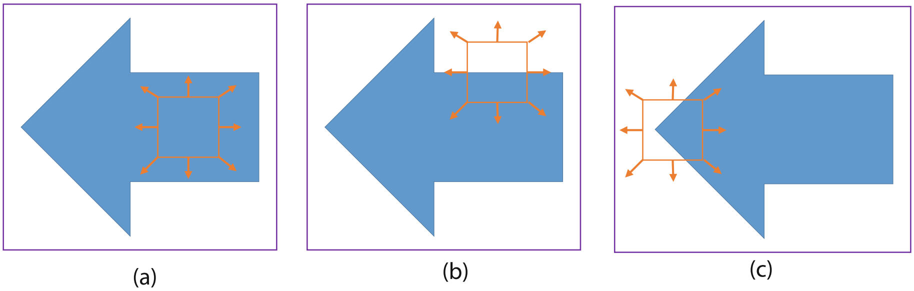


Fig. 4. Principle of Harris corner detector

When we move the window to find a corner, the corner should have large changes in any direction. Figure 4(a) shows the movement in a homogeneous region. In this case, the window does not have a change in all directions. However, in case of Fig. 4(b), the window have big changes in a horizontal direction but not a vertical direction. Finally, in case of Fig. 4(c), the window have a large amount of changes in any direction, therefore we estimate it as a corner. You can see the details of Harris corner detector in [7].

3 Circular Samples

In general, there are many corners in an image after performing Harris corner detector like Fig. 3. However, we are only interested in pattern feature points rather than all corners. Hence, we need to extract pattern feature points among all candidates. In this paper, we propose a circular sample to achieve the purpose. Figure 5 shows the diagram of the circular sample.

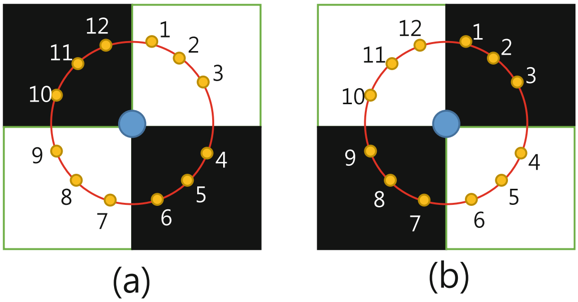


Fig. 5. Diagram of the circular sample

There are two cases of a pattern feature point which are shown in Figs. 5(a) and (b). Let's assume that the blue points in Fig. 5 are corners found by Harris corner detector. Then, we can define a circle of which center is the position of the blue point with radius *r*; the radius *r* should be smaller than the size of a single pattern. Along with the circle, we can obtain sample pixels represented by yellow points in Fig. 5. In case of Fig. 5(a), we examine whether the samples are properly located or not. For example, a color of the points 1–3 and points 7–9 should be white. Next, a color of the points 4–6 and points 10–12 should be black. We show a pseudo code of this step below.

```
Flag = true
For all samples s on a circle surrounding corner c
    If (s is in region 1 or region 3) and (Is ≠white)
         Flag = false
    If (s is in region 2 or region 4) and (Is ≠black)
         Flag = false

If flag = true
    Corner c is a pattern feature point
```

Here, *s* means a sample point on a circle surrounding the corner and I_s is an intensity value of the sample point. In case of Fig. 5(b), we can apply this algorithm inversely.

However, right after performing this step, there are many duplicate candidates indicating a same pattern feature point, which shown in Fig. 6.

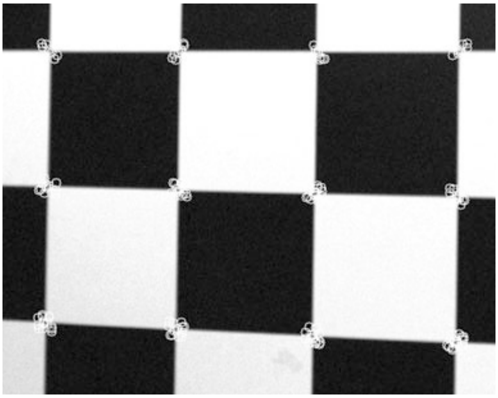
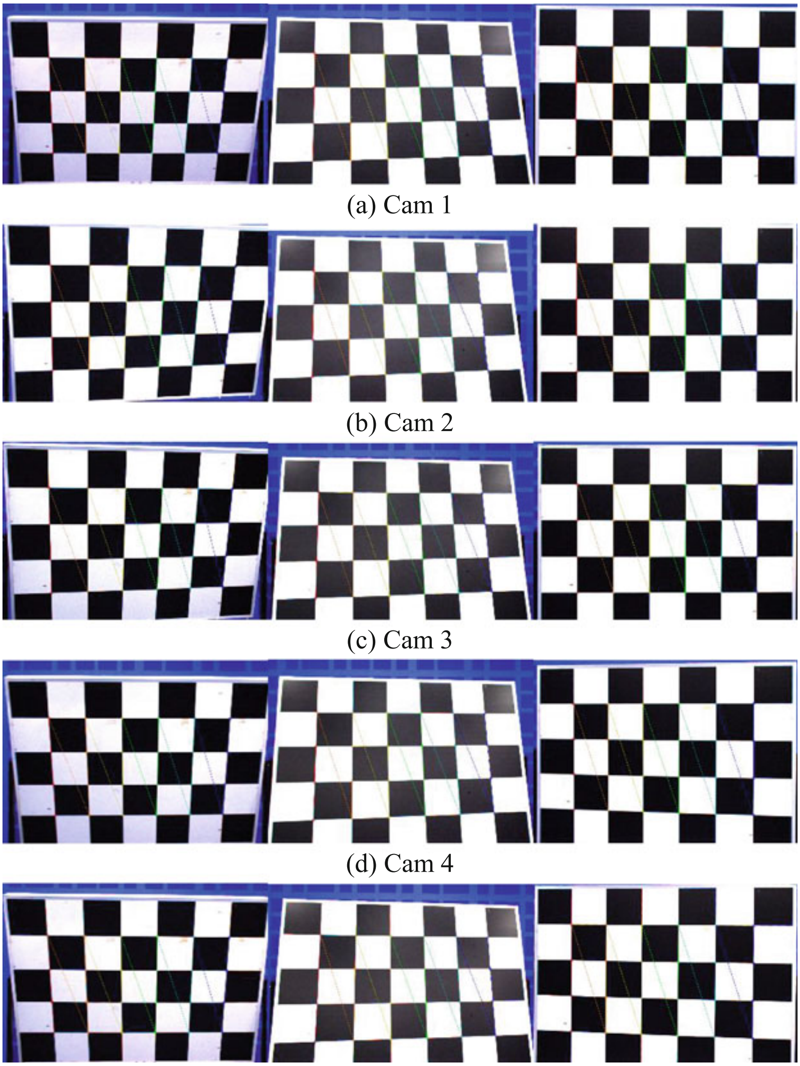


Fig. 6. Duplicated candidates for pattern feature points

In order to suppress duplicated candidates, we employ Euclidean distance among them. If a point is placed within a distance *d* from the other points, that point will be rejected. Otherwise, this point will be accepted.

4 Test Results

The proposed detecting method has been tested on calibration pattern sequences from five cameras placed in parallel. We captured 10 images for each camera. Figure 7 shows detected points by the proposed method.



(a) Cam 1

(b) Cam 2

(c) Cam 3

(d) Cam 4

(e) Cam 5

Fig. 7. Detected points by the proposed method

Here, we displayed three images for each camera and represented points as colored point and lines using cvDrawChessboardCorners() function in Opencv framework. We can find exact results of well detected points.

Next, we compared the camera parameters obtaining from Matlab calibration toolbox by hand and proposed method without a manual manipulation in Tables 1 and 2 respectively.

Finally, we show the difference between each matrix by using Euclidean norm in Table 3. we can see that the differences between camera parameters from Matlab camera calibration toolbox and the proposed method are sufficiently small to practically use.

Table 1. Camera parameters from Matlab calibration toolbox by hand

	Intrinsic matrix	Rotation	Translation
Cam1	$\begin{bmatrix} 1745.490 & 0 & 983.706 \\ 0 & 1746.419 & 514.462 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99986 & 0.00642 & 0.01500 \\ 0.00646 & -0.99997 & -0.00267 \\ 0.01498 & 0.00277 & -0.99988 \end{bmatrix}$	$\begin{bmatrix} -455.235 \\ 679.347 \\ 2805.469 \end{bmatrix}$
Cam2	$\begin{bmatrix} 1742.720 & 0 & 988.246 \\ 0 & 1744.012 & 507.778 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99982 & 0.00182 & 0.01875 \\ 0.00179 & -0.99999 & 0.00139 \\ 0.01875 & -0.00136 & -0.99982 \end{bmatrix}$	$\begin{bmatrix} -523.490 \\ 666.373 \\ 2806.729 \end{bmatrix}$
Cam3	$\begin{bmatrix} 1736.700 & 0 & 983.227 \\ 0 & 1737.410 & 521.987 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99985 & 0.00197 & 0.016926 \\ 0.00196 & -0.99999 & 0.00087 \\ 0.01692 & -0.00084 & -0.99985 \end{bmatrix}$	$\begin{bmatrix} -581.463 \\ 667.623 \\ 2801.644 \end{bmatrix}$
Cam4	$\begin{bmatrix} 1742.291 & 0 & 977.522 \\ 0 & 1743.313 & 511.662 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99942 & 0.00184 & 0.03392 \\ 0.00183 & -0.99999 & 0.00038 \\ 0.03392 & -0.00031 & -0.99942 \end{bmatrix}$	$\begin{bmatrix} -686.548 \\ 670.412 \\ 2791.815 \end{bmatrix}$
Cam5	$\begin{bmatrix} 1737.313 & 0 & 979.081 \\ 0 & 1737.073 & 527.819 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99920 & 0.00888 & 0.03888 \\ 0.00889 & -0.99996 & -4.63137 \\ 0.03888 & 0.00039 & -0.99924 \end{bmatrix}$	$\begin{bmatrix} -765.842 \\ 665.155 \\ 2788.822 \end{bmatrix}$

Table 2. Camera parameters from the proposed method without a manual manipulation

	Intrinsic	Rotation	Translation
Cam1	$\begin{bmatrix} 1744.770 & 0 & 983.243 \\ 0 & 1745.689 & 514.496 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99987 & 0.00643 & 0.01474 \\ 0.00647 & -0.99997 & -0.00269 \\ 0.01472 & 0.00279 & -0.99988 \end{bmatrix}$	$\begin{bmatrix} -454.499 \\ 679.270 \\ 2804.305 \end{bmatrix}$
Cam2	$\begin{bmatrix} 1742.702 & 0 & 987.739 \\ 0 & 1743.945 & 507.428 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99982 & 0.00180 & 0.01871 \\ 0.00177 & -0.99999 & 0.00124 \\ 0.01871 & -0.00121 & -0.99982 \end{bmatrix}$	$\begin{bmatrix} -522.660 \\ 666.949 \\ 2806.464 \end{bmatrix}$
Cam3	$\begin{bmatrix} 1736.837 & 0 & 983.280 \\ 0 & 1737.563 & 522.200 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99985 & 0.00200 & 0.01699 \\ 0.00198 & -0.99999 & 0.00099 \\ 0.01700 & -0.00095 & -0.99985 \end{bmatrix}$	$\begin{bmatrix} -581.558 \\ 667.259 \\ 2801.840 \end{bmatrix}$
Cam4	$\begin{bmatrix} 1742.837 & 0 & 977.400 \\ 0 & 1743.926 & 511.124 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99942 & 0.00184 & 0.03387 \\ 0.00184 & -0.99999 & 0.00013 \\ 0.03387 & -0.00007 & -0.99942 \end{bmatrix}$	$\begin{bmatrix} -686.338 \\ 671.267 \\ 2792.477 \end{bmatrix}$
Cam5	$\begin{bmatrix} 1737.958 & 0 & 979.117 \\ 0 & 1737.671 & 528.260 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.99921 & 0.00887 & 0.03872 \\ 0.00887 & -0.99996 & 0.00011 \\ 0.03872 & 0.00022 & -0.99925 \end{bmatrix}$	$\begin{bmatrix} -765.891 \\ 664.448 \\ 2790.042 \end{bmatrix}$

Table 3. Differences between camera parameters

	Cam1	Cam2	Cam3	Cam4	Cam5
Intrinsic matrix	0.8570	0.6170	0.2672	0.8230	0.7444
Rotation	2.6397e-04	1.5813e-04	1.3518e-04	2.4823e-04	2.2934e-04
Translation	1.3794	1.0443	0.4244	1.1014	1.4113

5 Conclusions

In this paper, we have proposed a new detection method to extract pattern feature points. From our experiments, we could obtain camera parameters with an ignorable difference with the result of Matlab calibration toolbox but eliminated the inconvenience of a manual manipulation. This proposed method is efficient when the number of camera and image increases especially in the multiview camera system. Although it can suffer from various conditions such as strong or weak illumination and a partial occlusion, it can be available under reasonable constraints. We expect that we can offer advantages of efficiency in multiview image processing and applications.

Acknowledgements. This research was supported by the 'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea(ROK). [GK15C0100, Development of Interactive and Realistic Massive Giga-Content Technology]

References

1. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1330–1334 (2000)
2. Mark, W., McMillan, L., Bishop, G.: Post-rendering 3D warping. In: Symposium on Interactive 3D Graphics, pp. 7–16 (1997)
3. Kang, Y., Ho, Y.: Geometrical compensation for multi-view video in multiple camera array. In: International Symposium ELMAR-2008 (2008)
4. Escalera, A., Armingol, J.: Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. Sensors 2010 **10**, 2027–2044 (2010)
5. Chu, J., GuoLu, A., Wang, L.: Chessboard corner detection under image physical coordinate. Opt. Laser Technol. **48**, 599–605 (2013)
6. Fiala, M., Shu, C.: Self-identifying patterns for plane-based camera calibration. MVA **19**(4), 209–216 (2008)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–151 (1988)