Automatic detection of chessboard and its applications

Jong-Eun Ha
Seoul National University of Technology
Department of Automotive Engineering
172 Gongneung-2Dong
Nowon-Gu, Seoul 139-743
Korea
E-mail: jeha@snut.ac.kr

Abstract. A chessboard image can be used as a pattern for camera calibration and pose estimation. We propose an algorithm for the automatic detection of a chessboard using one circle as a probe. We first generate candidates using the Lucas–Kanade feature detector algorithm. Finally, we find control points on the chessboard using properties inherent on a chessboard pattern. We suggest that a distinct edge along the border line and black and white sequence on the chessboard can be used as constraints. A circle is used as a probe for checking those properties. We show that the proposed algorithm can be used successfully in camera calibration and pose estimation. Experimental results using images acquired under the variation of scale, pose, and illumination shows the robustness of proposed algorithm. © *2009 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.3156053]

Subject terms: camera calibration; stereo; chessboard; augmented reality; pose estimation.

Paper 090142 received Feb. 27, 2009; accepted for publication May 4, 2009; published online Jun. 26, 2009.

1 Introduction

Camera calibration is an essential step for the acquisition of metric information of the world using cameras. It usually means the computation of extrinsic and intrinsic parameters of a camera. A typical camera calibration algorithm uses one-to-one correspondence between the 3-D and 2-D control points of a camera.[1–3]

Most approaches concentrate on the calibration method itself, and little attention is given into the automatic detection of control points on the calibration target's image. Heikkila[4] presented a camera-calibration algorithm using circular control points and analysis with bias correction for circular control points. He also showed some experimental results about the effect of the lighting condition on the accuracy of the detection of circular control points. Zhang[5] presented a calibration method by observing a planar pattern at least twice. He used the calibration pattern that consisted of 9×9 squares with nine special dots to automatically identify the correspondence between the points in the calibration target and points in the image. Approaches using one plane[5,6] require more than two images of the planar pattern at different configurations between the camera and plane. There also exists a degenerate configuration.[5,6] Matsunaga and Kanatani[7] designed an optimal grid pattern whose shape determined that the cross-ratio of adjacent intervals is different everywhere. An optimal Markov process that maximizes the accuracy of the matching is used in the generation of the cross-ratios.

Little attention is given to the automatic detection of control points on an image, and each lab has its own semiautomatic or automatic code specifically tuned for its own purpose. In OpenCV,[8] there is a code for the automatic detection of control points on a chessboard image, and it contains many user-defined threshold values.[9,10] Camera calibration algorithm in OpenCV is based on Zhang's algorithm,[5] and it requires multiple images of a planar chessboard pattern under different poses for the calibration of a camera. Claus and Fitzgibbon[11,12] proposed a robust fiducial detection algorithm using machine learning techniques and focus on the robust detection of a circular template to use it in a marker-based tracking system. Lavest et al.[13] proposed a calibration algorithm that simultaneously computes the 3-D coordinates of the control points and extrinsic and intrinsic parameters of the camera. They also emphasize that the accurate detection of control points on an image is important in calibration.

In our previous paper,[14] we presented an algorithm for the automatic detection of calibration markers on a chessboard image. Two circles are used as a probe for investigating symmetric properties on the chessboard image, and

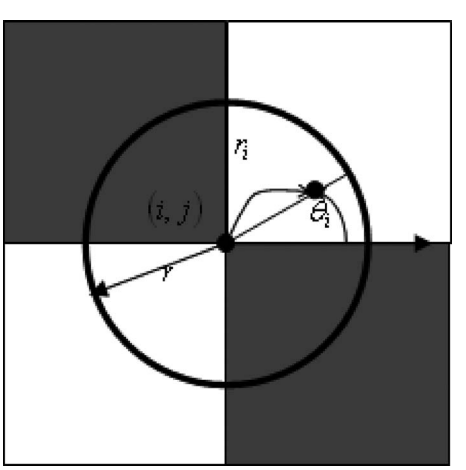


Fig. 1 Circular projection along the radius.

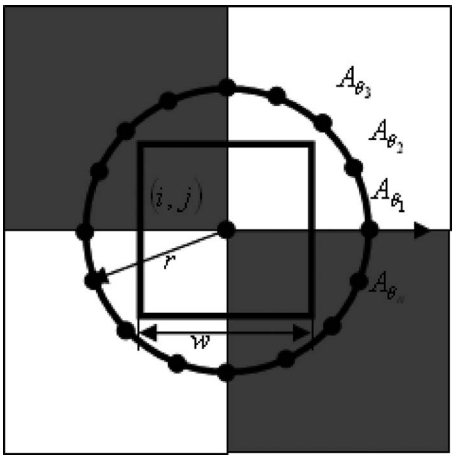


Fig. 2 The checking of brightness change at a calibration marker.

we scanned the entire image to find the location of markers. In this paper, we generate candidates by applying the Lucas–Kanade feature-detection algorithm[15] and then apply the proposed algorithm only for them, which results reduced computation time. The proposed algorithm also uses one circle as a probe that requires fewer thresholds compared to the previous algorithm. We applied our algorithm in two areas of the calibration of a camera and pose estimation.

2 Chessboard Detection through Two Stages

We consider the center position of the chessboard as control point. First, we reduce search space by finding candidates using the Lucas–Kanade feature detector algorithm.[15] Control points on the chessboard have a distinct brightness distribution so that the Lucas–Kanade feature-detector algorithm can find them without missing any. But, it also generates multiple points on the image that are not control points. We remove these false positives using properties inherent on the chessboard.

The chessboard pattern has the following properties. First, the magnitude of edge has the largest value along the four border lines because the chessboard pattern has a white and black pattern in sequence. Four border lines projected to lines on the image, while an angle is not preserved under the perspective projection. Second, a white and black pattern appears in sequence along the control point.

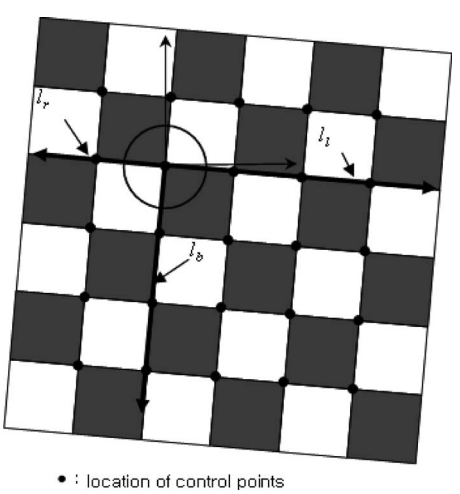


Fig. 3 The grouping of calibration markers.

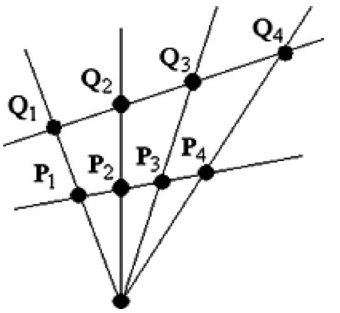


Fig. 4 The cross-ratio of four points on a line.

We examine those two properties using a profile that is computed by accumulating an edge magnitude along radial direction of a circle for predefined angles. We can consider it as circular probe. For each angle on a circle, we add an edge magnitude from a center to the circumstance of a circle along the radial direction, which is shown in Fig. 1. A value of a profile at a particular angle θ_i could be represented as

$$A_{\theta_i}(x,y) = \sum_{r=0}^{r=r_e} E(x + r\cos\theta_i, y + r\sin\theta_i). \quad (1)$$

(x,y) is the location on image, $E(x,y)$ is the magnitude of edge at location (x,y), and r_e is the radius of a circle.

For each point, we can obtain a one-dimensional profile. Next, we find four positions corresponding to border lines at a control point, they would have the four largest magnitudes in a profile. Thus, we can find them by detecting local maxima positions in a profile. We set the four directions as θ_1, θ_2, θ_3, and θ_4, in ascending order. A line is projected into a line under the perspective projection so that an angle between the same border lines should be 180 deg. We compute the difference of angles among the four directions as follows:

$$\theta^1_{\text{diff}} = (\theta_3 - \theta_1) - 180$$

$$\theta^2_{\text{diff}} = (\theta_4 - \theta_2) - 180. \quad (2)$$

Table 1 The values of parameters used in the experiments.

Symbol	Term	Value
r_e	Radius of a circle in Eq. (1)	10 pixels
$\theta^{\text{th}}_{\text{diff}}$	Angle difference threshold in Eq. (3)	30 deg
c_{th}	Threshold in Eq. (5)	70%
d^{th}_{l}	Distance threshold in Eq. (6)	10 pixels
δ_{cr}	Threshold in Eq. (8)	0.05

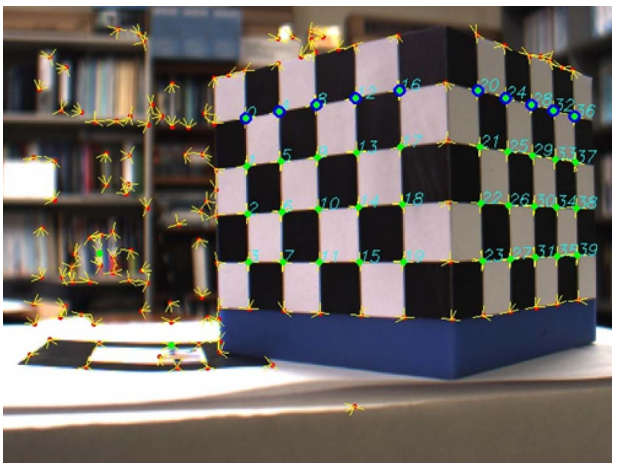


Fig. 5 The result of a detection of calibration markers by the proposed algorithm: blue and green circles are the points detected by the Lucas–Kanade detector,[15] green circles are after applying the symmetric constraints of the chessboard pattern, and blue circles are the chosen points as group's starting point, where four yellow lines are the four directions found in the profile by Eq. (1). (Color online only.)

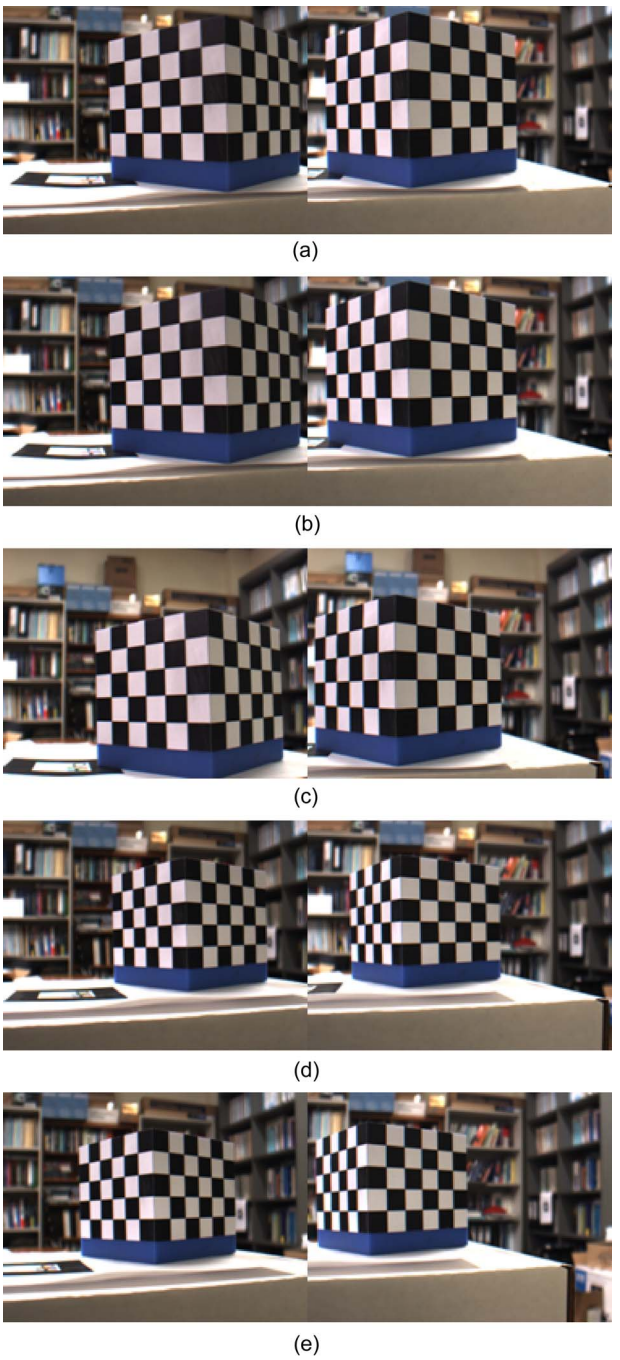
If a current location corresponds to a calibration marker, then θ^1_{diff} and θ^2_{diff} will have a value close to 0. We use an average value of θ^1_{diff} and θ^2_{diff}. If it is smaller than the threshold value of $\theta^{\text{th}}_{\text{diff}}$, then it is considered as a candidate for a calibration marker,

$$\theta_{\text{diff}} = \frac{\theta^1_{\text{diff}} + \theta^2_{\text{diff}}}{2} < \theta^{\text{th}}_{\text{diff}}. \tag{3}$$

Next, we investigate the change of brightness values along a control point (see Fig. 2) and use the detected four directions of θ_1, θ_2, θ_3, and θ_4 in this step. The opposite side of a boarder line at a control point will have a distinct brightness distribution of black and white. The mean value of intensity within the areas of $w \times w$ is used for checking sign change. In the previous step, we have found four directions corresponding to border lines. In each interval between two consecutive positions, we count the number of pixels that is higher or lower than the mean value. If the number of pixels that is greater than the mean value is larger than predefined value, then we set the region as a plus. Otherwise, the region is set as a minus, which is shown in Eqs. (4) and (5). If we follow along a circle whose center is a control point, then the sequence of sign value would be either $+ \rightarrow - \rightarrow + \rightarrow -$ or $- \rightarrow + \rightarrow - \rightarrow +$. We use this property and Eq. (3) as two constraints in verifying control points.

$$c^{\text{upper}}_{\theta_i} = \text{number of pixels that satisfy } I(x,y) > m[\theta_i, \theta_{i+1}]$$

$$c^{\text{lower}}_{\theta_i} = \text{number of pixels that satisfy}$$
$$I(x,y) < m[\theta_i, \theta_{i+1}] \ (for\ i = 1,2,3,4), \tag{4}$$



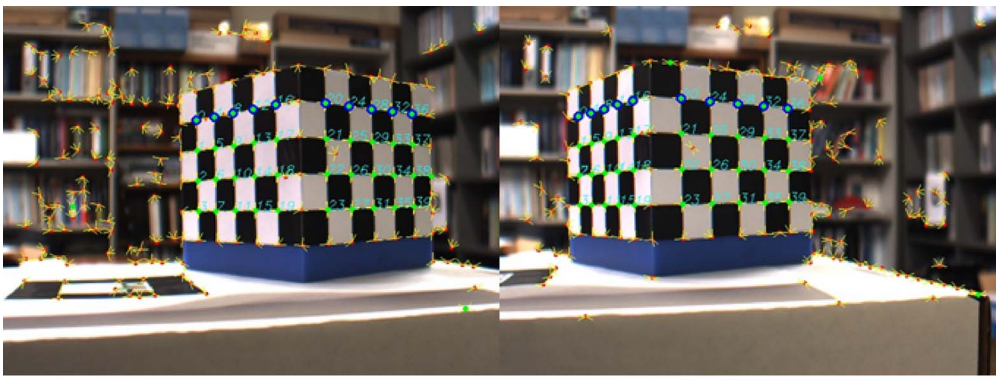
(a) (b) (c) (d) (e)

Fig. 6 Stereo images used in calibration (left side: left image, right side: right image) (a) set 1, (b) set 2, (c) set 3, (d) set 4, and (e) set 5.

$$\begin{cases} c^{\text{upper}}_{\theta_i} > c_{\text{th}} \rightarrow + \\ c^{\text{lower}}_{\theta_i} > c_{\text{th}} \rightarrow - \end{cases}. \tag{5}$$

3 Grouping of control points for calibration

One-to-one correspondence between 2-D coordinates on image and 3-D coordinates of calibration markers is required for the calibration of a camera. We present a grouping method for one-to-one correspondence using angles and the property of a cross-ratio.



(a) (b)

Fig. 7 The automatic detection of calibration markers by the proposed algorithm using set 4 of Fig. 5 (a) left image and (b) right image.

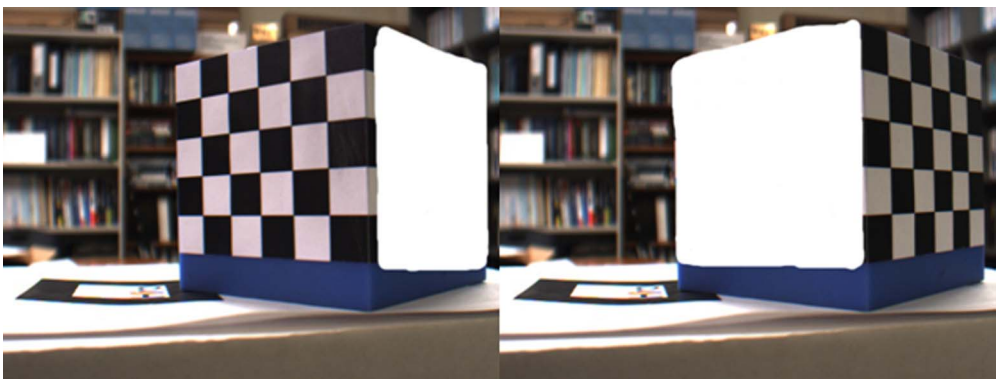


Fig. 8 Some images adjusted for the GML C++ camera calibration toolbox.[17]

(a)

(b)

Fig. 9 Images acquired using Sony XC-55 under the variation of (a) illumination and (b) pose.

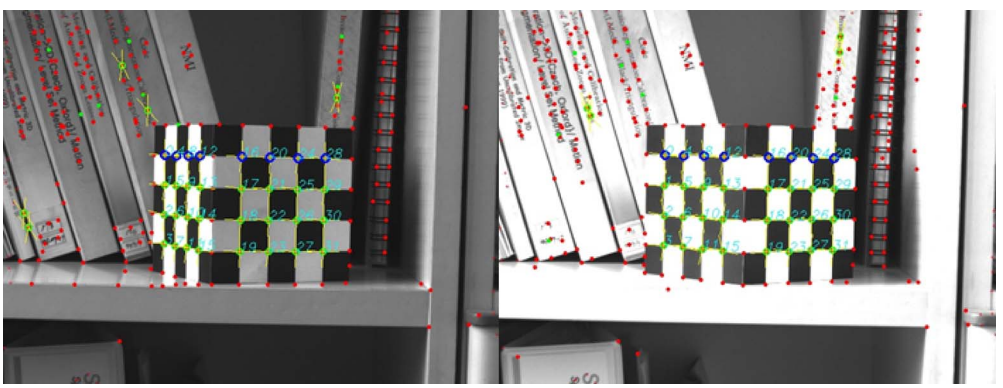


Fig. 10 Some results by the proposed algorithm using images of Fig. 11.

Table 2 Calibration results by Tsai algorithm[2] using automatic detection of calibration markers of the proposed algorithm.

		Intrinsic parameters			Extrinsic parameters		Lens distortion	3-D reconstruction error (mm)	
		Focal length	Scale factor (s_x)	Image center (pixel)	Translation ([mm)	Rotation (deg)	κ_1 ($1/mm^2$)	Mean	Std. dev.
Set 1	L	6.3049	0.9971	(323.7,257.5)	(39.3,19.5,224.7)	(90.7,–46.9, –1.6)	5.6272×10^{-3}	0.64	0.26
	R	6.1247	0.9979	(325.5,244.0)	(–30.6, 20.9,219.3)	(90.2,–45.7, –1.0)	3.2229×10^{-3}		
Set 2	L	6.3082	0.9970	(322.7,255.1)	(39.8,20.1,225.3)	(90.4,–46.7, –1.4)	5.6567×10^{-3}	0.59	0.31
	R	6.1338	0.9984	(325.2,243.0)	(–30.6,19.6,220.3)	(90.6,–45.6, –1.3)	3.3587×10^{-3}		
Set 3	L	6.3484	0.9968	(323.9,249.0)	(30.8,48.0,230.5)	(82.5,–48.5,4.4)	5.9637×10^{-3}	2.30	2.15
	R	3.9268	1.0143	(267.3,257.9)	(–24.8,38.5,149.1)	(89.3,–40.1,0.4)	-4.5772×10^{-2}		
Set 4	L	6.3054	0.9967	(328.3,270.7)	(20.6,14.3,299.9)	(91.6,–39.2, –2.2)	5.9801×10^{-3}	4.52	4.90
	R	4.4465	1.0313	(288.8,224.4)	(–33.7,28.0,222.8)	(89.1,–34.0, –0.0)	-4.5438×10^{-2}		
Set 5	L	6.1132	0.9982	(336.4,248.7)	(–6.2,22.2,286.3)	(89.6,–43.9, –1.0)	2.2129×10^{-3}	4.54	4.85
	R	4.7160	1.019	(293.8,226.9)	(–60.2,23.9,232.2)	(90.6,–39.9, –1.1)	-1.2166×10^{-2}		

Usually a chessboard as a calibration pattern consists of $m\times n$ control points. First, we check the existence of other control points along the left and right direction for each control point. The corresponding search direction of left and right is chosen among four directions of θ_1, θ_2, θ_3, and θ_4. We can generate lines along these directions. We can compute the distance from the control point to this line as Eq. (6). If it is smaller than predefined value, then it is further considered for grouping. If there are control points along either the right or left direction, then we search along the downward direction. Also, the corresponding search direction is chosen among the four directions of θ_1, θ_2, θ_3, and θ_4. We find control points whose distance from the line is smaller than the predefined value and select one who has the minimum distance. The same procedure iterates until there are no control points. This process is shown in Fig. 3, and we can obtain a list of control points along the downward border line

$$d_l = \frac{|ax' + by' + c|}{\sqrt{a^2 + b^2}} < d_l^{\text{th}}. \tag{6}$$

where (a,b,c) is coefficients of a line and (x',y') is a location of a point. If the distance is smaller than threshold d_l^{th}, the point is linked to a current point.

We use a cross-ratio of four points on a line as a verification step. The cross-ratio of four points on a line is preserved under projective transformations.[16] The cross-ratio is defined as

$$\text{Cr}(\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3,\mathbf{P}_4) = \frac{(X^3 - X^1)(X^4 - X^2)}{(X^3 - X^2)(X^4 - X^1)}, \tag{7}$$

where $\{X^1,X^2,X^3,X^4\}$ are the corresponding positions of each point along the line.[16]

Cross-ratio of four points is computed and compared to true value (see Fig. 4). If the difference of cross-ratio is within the predefined range as in Eq. (8), finally four points are considered as control points

$$\left|\frac{\text{Cr}(\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3,\mathbf{P}_4) - \text{Cr(True)}}{\text{Cr(True)}}\right| \leqslant \delta_{cr}. \tag{8}$$

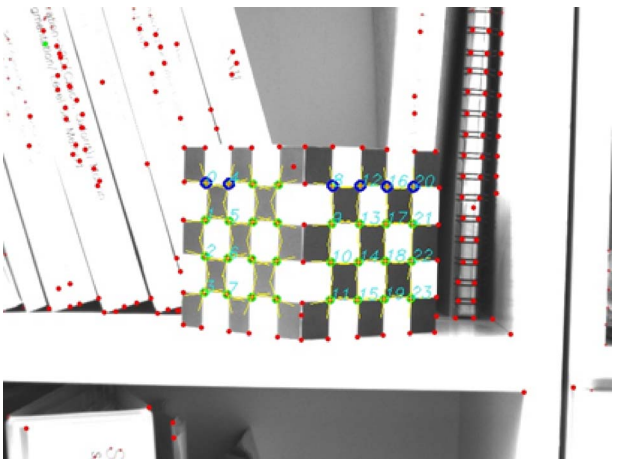
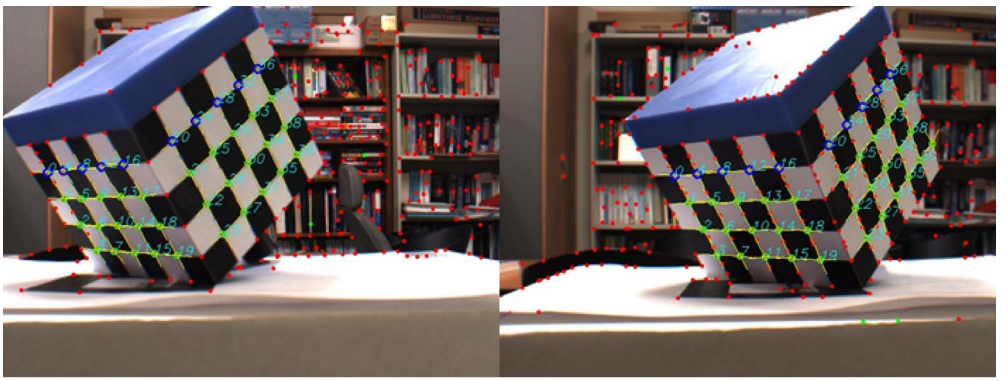


Fig. 11 A failure case in a grouping step (missing two groups) in experiments using images of Fig. 9.



(a) (b)

Fig. 12 Images acquired under large rotation of the calibration box and results by proposed algorithm.

Finally, we could get a one-to-one correspondence between 2-D and 3-D for all calibration markers. Therefore, we can automate the whole procedure of the calibration process by using the proposed algorithm of automatic detection of calibration markers on image and grouping of control points.

4 Experimental Results

In this section, we show the robustness of the proposed algorithm in various aspects. First, we compare the proposed algorithm to the GML C++ camera calibration toolbox.[17] Next, we show results using images acquired from different cameras. Finally, we show the result of proposed algorithm using images acquired under the variation of illumination and pose. We used the same parameters for the whole experiments and they are shown in Table 1.

Figure 5 shows the results including the initial points detected the Lucas–Kanade[15] feature detector and finally detected points by the proposed algorithm. The red circle represents points detected by Lucas–Kanade[15] algorithm. The green circle shows finally detected points that satisfy the two symmetric constraints of the chessboard. Four lines at a point represent the corresponding directions that have maximum magnitudes in a profile of Eq. (1). The starting point of a group is shown in a blue circle. In the experiment, we used chessboard pattern that had 20 control points on a plane. Each group consisted of four control points. Numbers are displayed on control points.

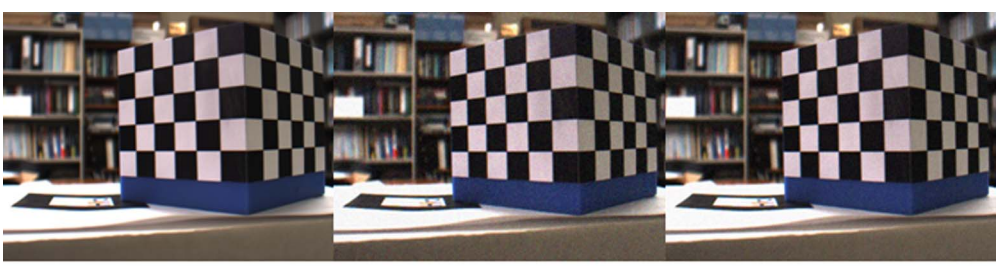
Figure 6 shows five different stereo images acquired using the same camera. The camera is a CMOS-based IEEE 1394–type camera (model Point Grey's Flea). The setting of the camera is fixed during image acquisition. Figure 7 shows some results by the proposed algorithm, which could detect whole control points successfully for the entire sequence in Fig. 6.

We compare the proposed algorithm with GML C++ camera calibration toolbox,[17] whose code is included in OpenCV,[10] for the automatic detection of calibration markers on chessboard. Figure 8 shows some images that are adjusted to run the GML C++ camera calibration toolbox. We prepared 10 images from the images on the left in Fig. 6. We tried both the holes and squares methods in GML for the detection of calibration markers on a chessboard. GML could not detect control points for the whole test images.

We tested using images acquired from different types of camera, and we also used the same parameters. For this experiment, we used a Sony XC-55 CCD–type camera and acquired images under the variation of illumination and pose, which are shown in Fig. 9. Ten images are used, and the proposed algorithm successfully detected all 32 control points. Some representative results are shown in Fig. 10. Grouping failed for only one image, which is shown in Fig. 11. In this case, the proposed algorithm detects all control points but the grouping failed due to large errors in four directions at the control points caused by excessive lighting.

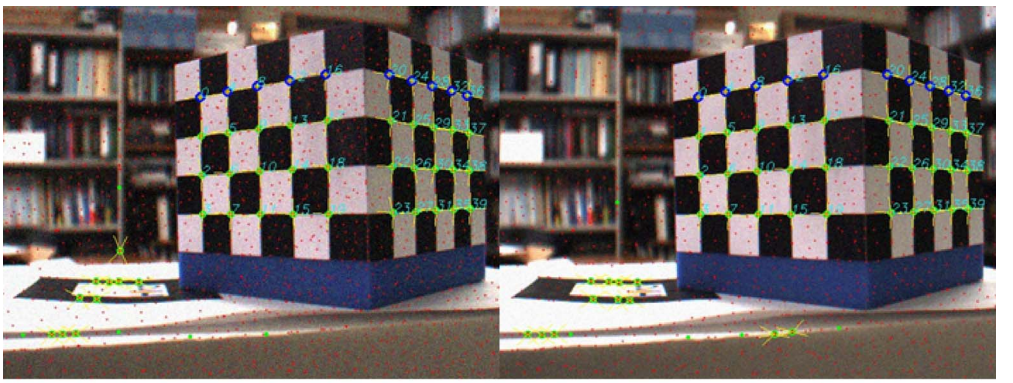
Figure 12 shows results using images acquired under large perspective distortion. The proposed algorithm could detect the whole control points on both images.

Figure 13 shows images with Gaussian noise that is



(a) (b) (c)

Fig. 13 Images with Gaussian noise (added using Adobe Photoshop): (a) original image, (b) 5% and (c) 7%.



(a) (b)

Fig. 14 Results using an image with Gaussian noise in Fig. 14: (a) 5% and (b) 7%.

added using Adobe Photoshop with amounts of 5 and 7%, and we generated 10 images for each noise level. Figure 14 is a result of the proposed algorithm, and we could successfully detect control points for all images.

Figure 15 shows the application of proposed algorithm in pose estimation of a humanoid robot. Chessboard patterns are attached on the surface of the body. The pose of a robot can be computed using a stereo system. The proposed algorithm detects the whole control points.

In the stereo system, we can calibrate both cameras using one shot of the calibration target. This makes it possible to see the sensitivity of the calibration algorithm on real time. Table 2 shows the result of camera calibration by the Tsai[2] algorithm and the error of reconstructed 3-D points. We no that the calibration algorithm heavily depends on the relative pose between the camera and calibration target. Calibration results are different, case by case even though we used the same camera without adjustment. The dimension of a box is $100 \times 100 \times 100$ mm. We could obtain a 3-D reconstruction with an accuracy of $0.6/100 \cong 1/166.7$.

We showed that the proposed algorithm can robustly detect control points on the chessboard under the variation of scale and rotation. Also, it can cope with perspective distortion. The proposed algorithm can be used in various applications, such as camera calibration and pose estimation.

5 Conclusion

A robust algorithm for the automatic detection of control points on chessboard is presented. The proposed algorithm uses inherent properties on the chessboard pattern using a circle as a probe. Candidates are generated using Lucas–Kanade feature detectors and refined through the investigation of properties on the chessboard. Control points on the chessboard could be found under the variation of scale, rotation, and perspective distortion. The proposed algorithm can be used in various applications, such as camera calibration and pose estimation.

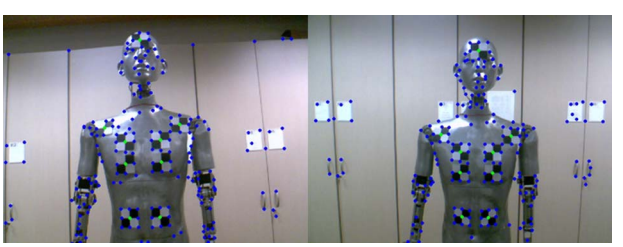


Fig. 15 Application of the proposed algorithm in pose estimation of a humanoid robot (blue circles are detected points by the Lucas–Kanade[15] feature detector, and green circles are finally the detected ones by the proposed algorithm). (Color online only.)

References

1. D. C. Brown, "Close-range camera calibration," *Photogramm. Eng.* **37**(8), 855–866 (1971).
2. R. Y. Tsai, "A versatile camera calibration techniques for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Rob. Autom.* **3**(4), 323–344 (1987).
3. O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA (1993).
4. J. Heikkila, "Geometric Camera Calibration Using Circular Control Points," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(10), 1066–1077 (2000).
5. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1330–1334 (2000).
6. P. Sturm and S. Maybank, "On plane-based camera calibration: a general algorithm, singularities, applications," in *Proc. 7th Int. Conf. Computer Vision and Pattern Recognition*, pp. 432–437, IEEE, Piscataway, NJ (1999).
7. C. Matsunaga and K. Kanatani, "Optimal grid pattern for automated matching using cross ratio," presented at *IAPR Workshop on Machine Vision Applications*, pp. 561–564 (2000).
8. ⟨www.intel.com/research/mrl/research/opencv/⟩.
9. C. Harris and M. Stephens, "A combined corner and edge detector," presented at *4th Alvey Vision Conf.*, pp. 147–151 (1988).
10. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698 (1986).
11. D. Claus and A. W. Fitzgibbon, "Reliable fiducial detection in natural scenes," presented at *Eur. Conf. on Computer Vision* (2004).
12. D. Claus and A. W. Fizgibbon, "Reliable automatic calibration of a marker-based position tracking system," in *7th IEEE Workshops on Applications of Computer Vision*, 5–7 January 2005, Breckenridge, CO, vol. 1, pp. 300–305, IEEE, Piscataway, NJ (2005).
13. J. M. Lavest, M. Viala, and M. Dhome, "Do we really need an accurate calibration pattern to achieve a reliable camera calibration?," presented at *Eur. Conf. on Computer Vision*, pp. 158–174 (1998).
14. J. E. Ha, "Automatic detection of calibration markers on a chessboard," *Opt. Eng.* **46**(10), 107203 (2007).
15. B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," presented at *Int. Joint Conf. on Artificial Intelligence*, pp. 674–679 (1981).
16. J. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, MA (1992).
17. Graphics E Media Lab, Moscow State University, Russia, ⟨http://research.graphicon.ru/calibration⟩.

Jong-Eun Ha received his BS and ME in mechanical engineering from Seoul National University, Seoul, Korea, in 1992 and 1994, respectively, and his PhD degree in robotics from the Computer Vision Lab at KAIST in 2000. During February 2000 through August 2002, he worked at Samsung Corning, where he developed a machine vision system. From 2002 to 2005, he worked at Multimedia Engineering in Tongmyong University. Since 2005, he has been assistant professor at the Department of Automotive Engineering in Seoul National University of Technology. His current research interests are intelligent robots/vehicles, object detection, and recognition.