Robust and Accurate Calibration Point Extraction with Multi-scale Chess-Board Feature Detector

Yang Liu[1], Yisong Chen[2], and Guoping Wang[2,*]

[1] Shenzhen Graduate School, Peking University
[2] Graphics Lab of EECS, Peking University
Beijiing Engineering Technology Research Center of Virtual Simulation and Visualization
{lypku,chenys,wgp}@pku.edu.cn

Abstract . Chess-board grid has been widely used for camera calibration and the associated feature point extraction algorithm draws much attention. In this paper, a multi-scale chess-board feature point detector is proposed, along with a chess-board matching algorithm for a specific marker used in our 3D reconstruction system. Experiments show that our method is more robust and accurate compared to commonly used approaches.

Keywords: Multi-scale chess-board corner detection, camera calibration, 3D reconstruction, calibration point extraction.

1 Introduction

Modern image-based 3D reconstruction systems reconstruct 3D models from a series of pictures capturing the objects. In general a specific chess-board grid is printed beforehand and put into the scene to help calibrate the cameras.

The chess-board grid we use for camera calibration consists of two parts (As shown in **Fig. 1**): a blank square area in the middle for placing objects to be reconstructed, and a chess-board grid region around the blank area. In Fig. 1b the feature points are circled in green. The four corners of the grid are designed with distinctive style for identifying global orientation.

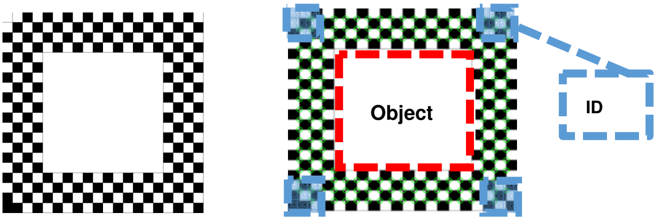
To ensure the integrity and accuracy of reconstruction, we have to capture the objects in as many views as possible and make sure they are well-focused. In most circumstances, the chess-board is arbitrarily placed, as shown in **Fig. 2**, and it often occurs that the board is ill-focused. These make it difficult to extract the chess-board vertices robustly and precisely. Besides, the variation of image size, the location of the board and the features of the objects increase the difficulty of calibration feature extraction as well.

In this paper, we propose a multi-scale chess-board feature detector for robust and precise detection of the calibration markers to meet these challenges. First, we use this

[*] Corresponding author.

T. Tan et al. (Eds.): IGTA 2014, CCIS 437, pp. 153–164, 2014.

detector to obtain all chess-board features in the image. Then we use the gradient information of the image to help eliminate outliers and organize the true vertices with a brand new bidirectional growth algorithm. Finally we map the features to 3D locations in the board to fulfill the camera calibration. As it shows in the experiments, our detector is quite stable and precise, even in extreme conditions like imperfect focus and ill-illuminations. Our method is completely automatic and performs better than commonly used methods.



(a) **(b)**

Fig. 1. Chess-board marker designed in this paper. (a) calibration chess-board (b) different parts consisted in this calibration board.

Fig. 2. Chess-board plane and the image plan have a large angle

2 Related Work

There are various published techniques for finding the chess-board vertices. H. Moravec [1] designed an algorithm which locates a corner by computing the gradients along eight directions with the anticipation that large response can be found along edges, but it is very prone to noises. Harris corner [2] is proposed based on [1] with an additional non-maximum suppression. F. Mokhtarian and R. Suomela [3] detected corners through curvature scale space with the help of Canny [4]. They didn't make use of the properties of the feature and are likely to suffer information loss by wide kernel filtering.

E. Rosten and T. Drummond [5] designed a quick corner detecting method called FAST which takes samples around a pixel around and determine whether it is a corner.

The Harris and Stephens [2] corner detector is adopted in [6] to locate a grid, before Hough transform is employed to constraint linearity and discard false responses.

This scheme performed badly when the grid is distorted because of optical distortion or on a non-planar surface.

Yu and Peng[7] adopt a pattern-match method to find specific features by measuring the correlation over all the image. This method fails when the grid is rotated relative to the patterns in store.

Sun et al. [8]describe a method which places a rectangular or circular window over every position of the image before achieving a 1D binarized vector along the perimeter. The positions where the vector has four regions are determined to be chess-board vertices. This method is somehow rather slow and prone to noises. The performance also relies on the result of binarization.

Shen Cai and Zhuping Zang [9] designed a deformed chessboard pattern for automatic camera calibration, but the precision and robustness of their method needs to be improved.

Stuart Bennett and Joan Lasenby [10] offer an instructive approach where they emphasize the importance of chess-board detector and design ChESS which is both simple and quick. It is rather accurate but tends to produces much more false features. The sing-scale property also aggravates its limitation.

Besides, some open source labs such as OpenCV [11] and Matlab [12] tools are widely used for chess-board grid detection for their convenience, but they both have some problems concerning to accuracy and robustness.

3 Corner Detection Algorithm

Our chess-board marker detection algorithm for camera calibration mainly consists of two steps. First, we obtain all the chess-board features with the help of the multi-scale chess-board feature detector and eliminate outliers from them. Then we organize the vertices found in former step and map them to 3D locations in the marker. The detailed algorithm will be introduced by these two steps.

3.1 Detecting Chess-Board Vertices

In this section, we propose and justify several properties of the chess-board **calibration pattern**, based on which we design a single-scale annular chess-board **feature** detector and then develop it into a multi-scale detector to ensure the robustness and precision of the detection.

3.1.1 Annular Chess-Board Detector

When we put a sample circular window over a chess-board vertex, the points on opposite sides **of a diameter** tend to have similar intensity and those on perpendicular **radii** should have very different intensity (**Fig. 3**). Based on this observation, we build an annular chess-board detector which has two kinds of energies: E_{sym} and E_{dif}.

Unlike [10] where the two properties are combined into one, we find it produces much fewer false features if considering them separately.

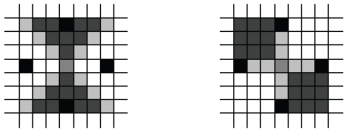


Fig. 3. points on opposite and perpendicular sides [10]

As illustrated in **Fig. 4**, taking c_i as a sample point on the circular window, c_i' is the point on the opposite side and b on the perpendicular side, N is the number of sampling points on the semicircle, these two energies can be formulated as:

$$E_{sym} = \sum_{i=0}^{N}(c_i - c_i')^2 \tag{1}$$

$$E_{dif} = \sum_{i=0}^{N}(c_i - b)^2 \tag{2}$$

The former energy should be small over **the semicircle** while the latter relatively larger.

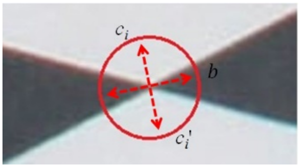


Fig. 4. Chess-board vertex

3.1.2 Multi-scale Annular Chess-Board Detector

If we adopt single-scale detector over the image, the radius of sampling window must be carefully chosen. Small rings over the blurred region of a vertex can't provide discriminating features while too large rings risk sampling pixels from unrelated squares which doesn't form the current features (**Fig. 5**). This creates a dilemma when processing different images. It is both unrealistic to apply only one sampling radius over all images and also inconvenient to choose an appropriate radius for every image.

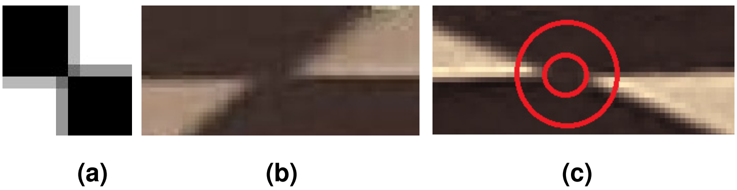


Fig. 5. sampling circle (a) ideal vertex (b) vertex with blurriness(c)sampling circles of different sizes

To solve these problems, we employ a pragmatic multi-scale scheme which carries out sampling windows of different radius over a position, just like [8] (**Fig. 6**). The construction of multi-scale annular chess-board detector is quite quick using similar method to SURF [13]; however, it's much slower than that of the single-scale detector. We propose some strategies to speed up the detection in 3.1.4, which makes our detector more competitive.

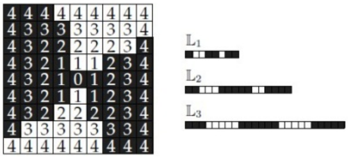


Fig. 6. Sun's layers [8]

3.1.3 Feature Selection

Our detector is quite distinctive and can almost detect all the visible chess-board vertices, however, false features might exist. Eliminating the false features is crucially important. In our study, the vertices must satisfy these constraints while false ones can't satisfy all of them:

1. $E_{sym} < E_1, E_1$ is an upper bound we set
2. $E_{dif} > E_2, E_2$ is an lower bound we set
3. There must be four color regions on the circular window
4. The path between two neighboring vertices must be along the edge of a chess-board square

The construction of multi-scale detector over every pixel is time consuming, so we start with a small radius and check the first three constraints, and increment it gradually if it fails the constraint 2 and 3 while satisfies constraint 1. Constraint 4 is employed by computing the sobel response of the image (**Fig. 7**). Large response tends to be along the edges of the square. Non-minimum suppression of E_{sym} is used over a small region where several features exist.

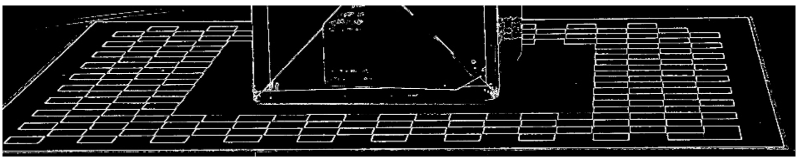


Fig. 7. Sobel response of the image

As shown in **Fig. 8**, sufficiently many vertices are detected while few false ones left.

Fig. 8. Vertices detected

3.1.4 Speed-Up Strategy

We propose two strategies to speed up the detection scheme. Instead of constructing our detector over every pixel of the image, we consider those positions where sobel responses are relatively high. Anyway, the threshold should be low enough to not risk eliminating the accurate positions of the vertices when blurring exists. Down-sampling the image can notably accelerate the computation, but an additional step to find the accurate positions of the vertices in the original image should be added.

3.2 Mapping the Vertices to 3D Board

After the vertices are detected, we must find out their locations in the 3D board before camera calibration. As shown in **Fig. 9**, the vertices form 12 lines along the periphery of the marker and each 3 lines make up a group. Usually at most two groups are invisible because of the occlusion of the objects, so we have at least two intersecting groups to locate an ID in the corner of the board. Hough transform [14] is the most widely used method when dealing with line fitting problems, but it is at great disadvantage when there's remarkable camera distortion. Inspired by [6], we design a bidirectional growth algorithm to fit these lines.

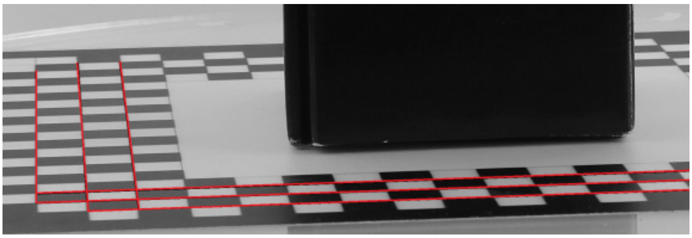


Fig. 9. 12 lines along the periphery

The algorithm is demonstrated in **Fig. 10**. For a vertex A, we find the closest vertex, named B, with which can form a path along the edge of the square (consider the sobel response). Put A and B in the inlier set and make growth along $\overrightarrow{AB}$: If another vertex and the last two inlier vertices lie along the edges of square, then put it into the set and keep looking for the next one . This process ends before the growth along the other direction $\overrightarrow{BA}$ starts. One-direction growth is enough when the vertices are ordered along the x or y axis.

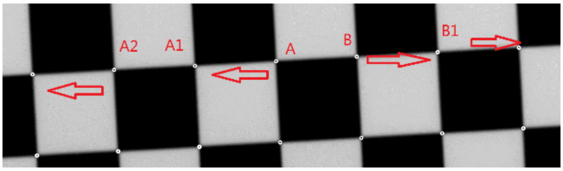


Fig. 10. Bidirectional growth algorithm

The lines successfully fit are illustrated in **Fig. 11.**

Fig. 11. Results of line fitting

The four ID codes in the corners of the grid are different from each other and have quite stable and distinctive appearances that are easy to be recognized. The 3D coordinates of the vertices detected are then obtained by using the method in [15].

4 Experimental Results

To show the contributions of our approach, we have performed the following two experiments: First, to evaluate the precision we compare the average reprojection error of the chess-board vertices detected by our detector with that by Harris and ChEss; Second, to evaluate the robustness we compare the chess-board recognition success rate of our method with that of J. Sun [16] and de la Escalera. We conduct an additional experiment over the images with different illumination, image sizes and camera poses to prove the validity of our method.

4.1 Precision

The dataset we use for this experiment consists of 20 groups (each group has about 20 images captured around the object on the chess-board) of images with good illumination.

These images can also be divided into 3 categories according to the angle between the image plane and the chess-board plane: high position (the angle is less than 30 degree), middle position (the angle is less than 60 degree and more than 30 degree), low position (the angle is more than 60 degree). As **stated** before, the camera focuses mainly on the objects to be reconstructed, so the **calibration pattern** is often captured with imperfect focus, which often cause blurriness in the condition of low position. We separately measure the average distance between the projected points of the 3D vertices and the vertices detected with Harris, ChEss and our detector. As shown in **Fig. 13**, our detector performs as well as ChEss, while much better than Harris, the average reprojection error is less than 1 pixel in all three conditions. We can also see that imperfect focus affects the accuracy, but our detector performs much more stable than Harris. ChEss is rather accurate but tends to produces much more false features and the sing-scale property also aggravates its limitation.

(a) **(b)** **(c)**

Fig. 12. Three categories divided by camera positions (a) high position (b) middle position (c) low position

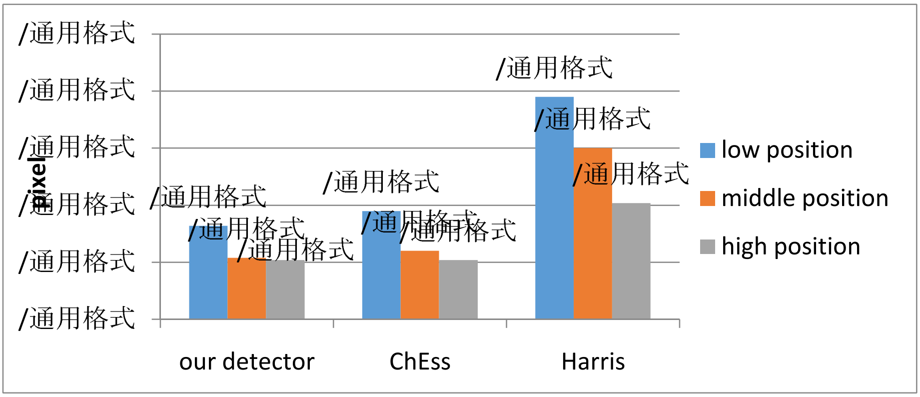


Fig. 13. Reprojection error of ChEss, Harris and our detector

Successful chess-board detection is to extract enough vertices, while discarding the false ones, and map them to 3D locations in the board to complete the camera calibration. There are generally three kinds of approaches in this field: J. Sun's method using LSC and line fitting scheme, Escalera method combining Harris with Hough transformation.

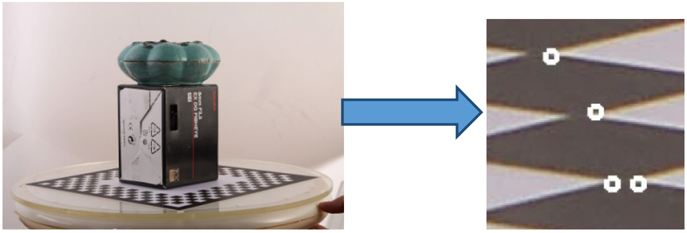


Fig. 14. An example where Harris corner deviants from true vertices

Our method differs with these two approaches. The comparison will be conducted in this experiment.

The dataset we use here consists of 40 groups of images which can also be divided into three categories like 4.1. We measure the chess-board recognition success rate and show some examples where these three methods would fail.

Fig. 15 shows that our method succeeds to calibrate the most images, which means it's the most robust among the four methods. The other three perform rather badly in the condition of low camera position. The line fitting scheme of de la Escalera is at a disadvantage when distortion or blurring exists. J. Sun's method has difficulty with discarding false vertices.

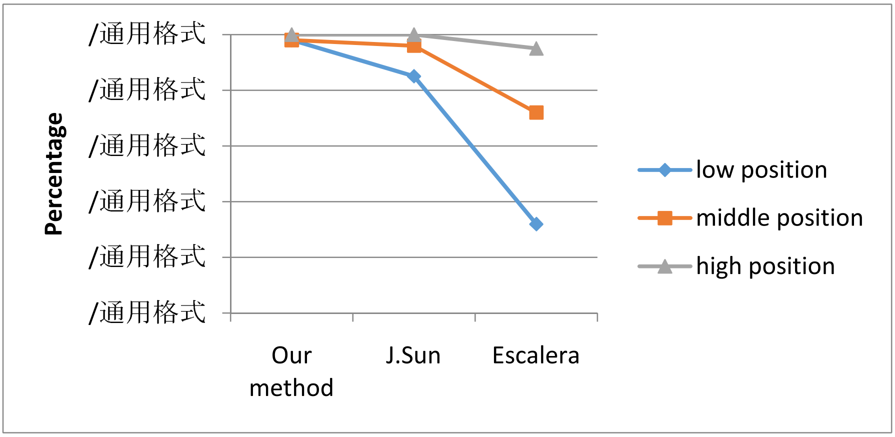


Fig. 15. Chess-board recognition success rates

4.2 Other Factors

We conduct this experiment on 200 images taken by us with different illumination, image sizes and camera poses and another 200 images taken by volunteers with little knowledge about our algorithm except the suggestion of capturing the object and the board at the same time. As shown in **Fig. 16**, we successfully complete calibration of almost all cameras of the first set, while we fail more over the second set due to some ill-captured images. We think this experiment is very necessary since our algorithm is to serve the users who might care about the performance only. It also helps us to find out the limitations and problems, or at least to make the rules of capturing clearer to users.

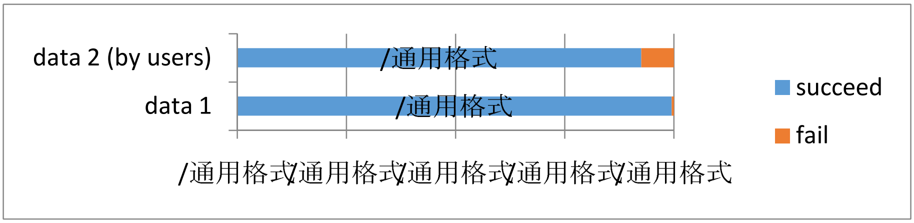


Fig. 16. Experiments on user-captured data

We show some results of chess-board vertices detection and the 3D models correspondingly reconstructed. With the help of our method, we can reconstruct quite accurate and vivid models. (We choose 4 images from each scenario)

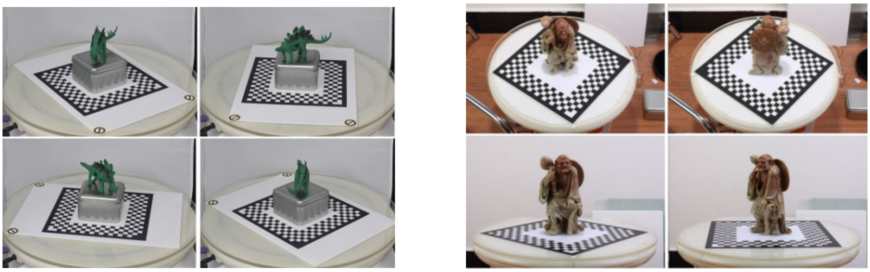


Fig. 17. Original images

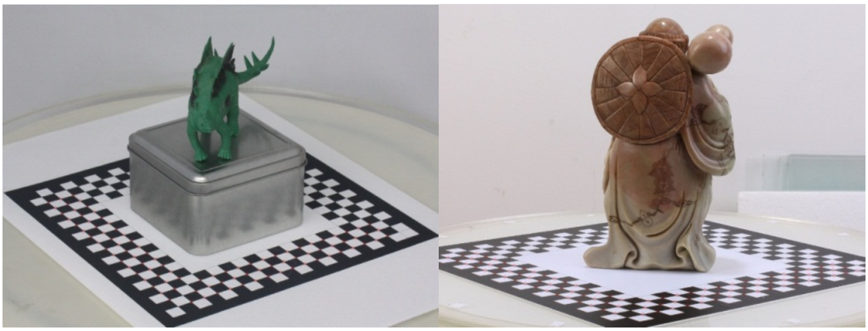


Fig. 18. Vertices detection results

Fig. 19. Reconstructed models

5 Discussion and Conclusion

In this paper we justified several properties of the chess-board vertices and designed a multi-scale detector with which we develop an accurate and robust chess-board corner detection algorithm that, according to experiments, performs well even in ill-illuminated and ill-focused conditions. The accuracy and robustness of our detector allows it to be employed in more applications related to chess-board detection. It can also help to refine the vertices detected by other methods.

Acknowledgements. This research was supported by Grant No 2010CB328002 from The National Basic Research Program of China(973 Program), Grant No. 61232014,61121002,61173080 from National Natural Science Foundation of China. Also was supported by Grant No. 2013BAK03B07 from The National Key Technology Research and Development Program of China.

References

1. Moravec, H.: Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Tech. Report CMU-RI-TR-3 (1980)
2. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proceedings of 4th Alvey Vision Conference, pp. 141–151 (1988)
3. Mokhtarian, F., Suomela, R.: Robust Image Corner Detection Through Curvature Scale Space. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1376–1381 (1998)
4. Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8, 679–698 (1986)
5. Rosten, E., Drummond, T.: Fusing Points and Lines for High Performance Tracking. IEEE International Conference on Computer Vision 2, 1508–1515 (2005)
6. Escalera, A., Armingol, J.M.: Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. Sensors 10(3), 2027–2044 (2010)
7. Yu, C., Peng, Q.: Robust recognition of checkerboard pattern for camera calibration. Optical Engineering 45(9), 093201 (2006)

8. Sun, W., Yang, X., Xiao, S., Hu, W.: Robust checkerboard recognition for efficient non-planar geometry registration in projector camera systems. In: Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems. PROCAMS 2008, pp. 1–7. ACM, New York (2008)
9. Cai, S., Zang, Z.: A Deformed Chessboard Pattern for Automatic Camera Calibration International Conference on Advanced ICT (2013)
10. Bennett, S., Lasenby, J.: ChESS – Quick and Robust Detection of Chess-board Features. CoRR (2013)
11. Vezhnevets, V.: OpenCV Calibration Object Detection, http://graphics.cs.msu.ru/en/research/calibration/opencv.html
12. Bouguet, J., Strobl, Y., Sepp, K., Paredes, W., Arbter, C.: Camera Calibration Toolbox for Matlab (2008), http://www.vision.caltech.edu/bouguetj
13. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)
14. Duda, R.O., Hart, P.E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. Artificial Intelligence Center (SRI International) (1971)
15. Chen, Y., Sun, J., Wang, G.: Minimizing geometric distance by iterative linear optimization. In: ICPR, pp. 1–4 (2010)
16. Sun, J.: Design and Optimization of a Calibration Point Extraction Algorithm in the Context of 3D Model Reconstruction (2012)