World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:4, 2016

X-Corner Detection for Camera Calibration Using Saddle Points

Abdulrahman S. Alturki, John S. Loomis

***Abstract*—This paper discusses a corner detection algorithm for camera calibration. Calibration is a necessary step in many computer vision and image processing applications. Robust corner detection for an image of a checkerboard is required to determine intrinsic and extrinsic parameters. In this paper, an algorithm for fully automatic and robust X-corner detection is presented. Checkerboard corner points are automatically found in each image without user interaction or any prior information regarding the number of rows or columns. The approach represents each X-corner with a quadratic fitting function. Using the fact that the X-corners are saddle points, the coefficients in the fitting function are used to identify each corner location. The automation of this process greatly simplifies calibration. Our method is robust against noise and different camera orientations. Experimental analysis shows the accuracy of our method using actual images acquired at different camera locations and orientations.**

***Keywords*—Camera Calibration, Corner Detector, Saddle Points, X-Corners.**

I. Introduction

CAMERA calibration has attained greater attention recently due to the increase of surveillance, security and gaming applications in the last decade. The focal length, lens quality and positioning of the camera are factors that must be known for these applications. To obtain these factors accurately, the intrinsic and extrinsic camera parameters must be determined. Intrinsic parameters describe the camera's focal length, along with any distortion it creates in imagery. Extrinsic parameters describe the camera's location and orientation with respect to an outside, or "world", coordinate system. In systems with multiple cameras, these parameters are necessary to determine spatial relationships between the cameras. Camera calibration can be roughly classified into two groups. Photogrammetric calibration uses images collected from targets with precisely known spatial features. Self calibration, on the other hand, uses images of the same static scene, and develops correspondences between the images to perform the calibration. This work examines a photogrammetric method, which is a well-developed and reliable method for estimating camera intrinsic parameters. Researchers have used many patterns (calibration targets) for camera calibration. Fig. 1 shows three different patterns.

Abdulrahman S. Alturki and John S. Loomis are with the Department of Electrical and Computer Engineering, University of Dayton, Dayton, OH, 45469 USA (e-mail: aalturki1@udayton.edu, jloomis1@udayton.edu).

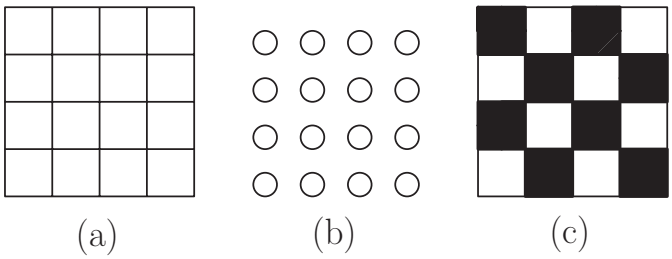

Fig. 1 (a) Cross lines (b) Circles (c) Checkerboard Patterns

The seminal work by Tsai [1] has influenced researchers to adopt a checkerboard target for calibration. The corners of the checkerboard can be precisely located with a variety of semi-automated procedures, because the white and black boxes result in strong corner features. Typically, the first step of a calibration procedure is to capture a set of images of the checkerboard at different positions relative to the camera. For each image in this set, the corner point locations are identified, and these are then used to determine the intrinsic and extrinsic parameters of the camera with numerical techniques.

The identification of corners in checkerboard images is of prime importance, because the quality of the camera calibration is sensitive to the accuracy of the corners. There are many approaches to using the corner points (or other features for other types of targets) for determining intrinsic and extrinsic parameters of the camera. For example, the plumb line method is used to account for the lens distortion that is a function of target position in the field of view. In [2], a novel model for calibration based on generalized projective mappings is presented. Two calibration planes in 3D are projected onto the image plane. Calibration approaches can be generally organized according to those that use direct nonlinear minimization, closed-form solutions, or two-step methods [3]. Direct nonlinear minimization involves the iterative computation of parameters, using equations that relate the corner point locations in space to the parameters. This approach is useful for characterizing many types of distortion. The closed-form approach is simpler since it does not require iteration, and the parameters are directly computed from the corner point positions. However, the closed-form approach does not model image distortion. The two-step approach computes most of the parameters in a closed-form fashion, and the remaining parameters are found with iterations. This allows one type of common distortion, radial distortion, to be modeled.

There are many techniques for performing calibration, but there is less research conducted on the necessary extraction of corner points from planar checkerboard imagery. This is often

performed manually, which ultimately is a time consuming and labor intensive process. For this reason, it is significant to search out a method which automatically extracts the corners from images.

There are several methods for automatic corner detection proposed in the literature. One approach is to find edges of the checkerboard pattern and fit lines to them [4]. The intersections between the fitted lines determine the corners of the checkerboard. However, the edges in the image are typically curved due to radial distortion, which creates problems for this technique. A second algorithm begins by prompting the user to identify the four outside corners in the image, along with the number of rows and columns in the pattern[5]. The algorithm then uses this information to automatically identify the interior corners in the pattern. However, the manual extraction of the four outside corners is time consuming, since calibration requires many target images. Zhongshi Wang [6] presented a methodology which can automatically detect all of the corners of the checkerboard pattern. It is based on the grid line architecture and local intensity characteristics of the planar checkerboard image. The approach consists of the detection of the corners on image, recognition of corners at the intersection of two grid lines, and the intersection of black and white squares.

Some approaches use vanishing points, which are an important geometric image parameter and can be helpful in calibrating cameras [7], [8]. A vanishing point exists at the convergence of a set of lines in the image plane, and the analysis of these lines provides information about invariant characteristics of the 3D scene, such as object dimension and depth.

Reference [9] presents the novel technique of finding checkerboards in blurred images regardless of their poses with high accuracy. The procedure involves two steps in which the detection of the checker board in the image is done along with the calculation of initial pixel coordinates for extracting corner as the first step. In the second step, corners are refined to sub pixel accuracy on the basis of Lucheese and Mitra technique [10].

In this paper, we present a fully automatic and robust algorithm for X-corner detection. This paper will be organized as follows. Section II reviews the details of related work on automatic corner detection. Section III covers the algorithm used in this work. Section IV presents the results and performance of this algorithm, as applied to a set of images. Section V concludes and discusses the implications of the results.

II. Related Work

Most of the camera calibration work in the literature that uses known geometrical targets focuses on algorithms that estimates the parameters from key points on the targets. The checkerboard target is used most frequently, which contain an array of X-corners. Although detecting the X-corners accurately is important, there is relatively little work on this key step. There are two commonly used techniques for finding X-corners. Bougets Matlab's toolbox [5] is a semi-automated X-corner detector, because it requires the user to manually click on the four extreme corners of the pattern. There is an OpenCV method based on Zhang's algorithm [4] with greater automation, although the number of rows and columns in the checkerboard and several thresholds must be entered by the user. The performance of the OpenCV algorithm is sensitive to the size and number of the squares in the image.

The idea of using X-corners in checkerboard images as points of interest was introduced by Moravec in 1977 [11]. The automated detection of these points is most often achieved with texture-based methods, which use the local intensity properties around the corner points. The Harris corner detector is a commonly used texture-based filter, and it is robust to rotation, changes in scale and image noise [12]. This detector applies an auto-correlation to small regions of the image, comparing the regions to shifted versions of themselves. This produces a matrix for each pixel in the image, and the eigenvalues and eigenvectors of the matrix provide information about the pixel. If the pixel is a corner, both eigenvalues are large. A threshold can therefore be used to identify corner points in the checkerboard image. Another well developed texture-based method is the SUSAN (Smallest Univalue Segment Assimilating Nucleus) detector, which applies a circular mask to detect corner strength for pixels [13]. Unlike the Harris filter, the SUSAN detector does not use derivatives. For each mask position over the image, it counts the number of image pixels with the same intensity as the center pixel. Comparing this to a threshold determines whether the center pixel is at a corner point in the image. Other texture-based techniques for locating checkerboard corners rely on wavelets [14], or blob detectors [15].

A second category of automated corner detection algorithms relies on the known geometry of the corners with respect to each other. For example, the corners occur at the intersections of lines in the image. Geometry-based techniques have difficulties when the image distortion is severe, causing lines to appear curved. Often, the geometrical relationships between corner points is used in combination with texture-based techniques, in order to eliminate spurious points.

Corner-detection algorithms most often return corner points as discrete pixel locations, with the same resolution as the original image. It is possible, however, to refine the location to sub-pixel precision by using a parametric model to represent the local region around corners. One example of this approach uses a Harris filter to first locate the corner pixels [16], [10]. The local image intensity around each corner point has a saddle shape, and can be represented by a hyperbolic function. The position of the saddle point is found by setting the partial derivatives of the hyperbolic function to zero. This location provides a sub-pixel approximation to the corner. Our proposed automated X-corner detection method also uses a second-order parametric model for the local intensity around corners, and treats the sub-pixel corner location as a saddle point.

A robust corner detector should satisfy the following requirements [17]:

- All correct corners should be detected .
- False corners should be eliminated ($Refine$).

- Correct corners should be precisley placed on thier true locations ($Localizations$).
- Detection should be robust regardless of scale and orientation of the pattern in the image.
- Detector should have minimum processing time (for real time applications)
- Detector should be robust against noise.

III. Saddle Points

Lines and edges are important features for many image processing and computer vision applications. An edge in an image corresponds to a significant local change in intensity level. Intensity can change due to various physical shape of the objects or because of different geometric affects (object boundary, surface boundary and occlusion boundary). Fig. 2 shows different types of edges.

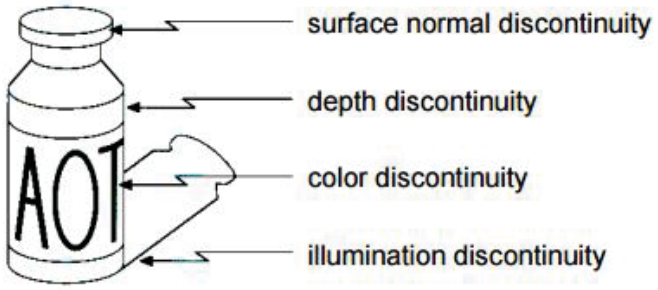

Fig. 2 Origin of Edges

Ideal edges are formed by a step function $E(x)$ defined by

$$E(x) = \begin{cases} 1 & \text{if } x > 0 \\ 1/2 & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1)$$

Real edges may be obtained from ideal edges by convolving with an imaging Point Spread Function (PSF) such as a Gaussian. An image of a real and ideal edge are shown in Fig. 3.

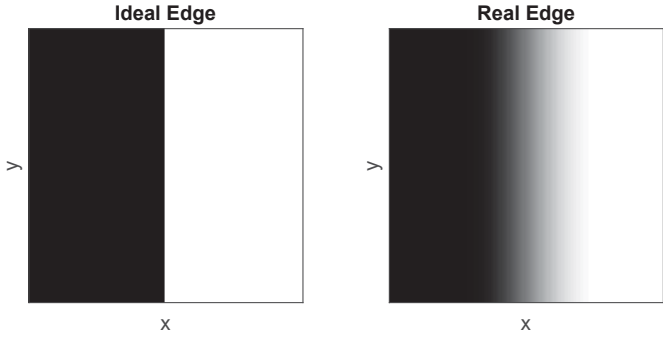

Fig. 3 Ideal vs real edge

The checkerboard pattern of interest in this work consists of an array of $x - shape$ corners, formed by the alternating black and white squares. The model for an ideal $x - shape$ corner is given in Fig. 4. It can be mathematically expressed by the function $E(xy)$.

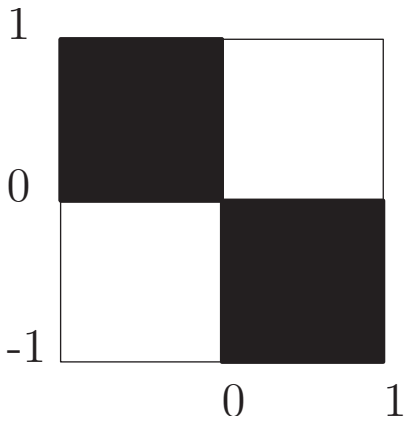

Fig. 4 Ideal X corner

The location of the corner in the ideal x-shape forms a natural saddle point. To illustrate this, Fig. 5 shows a model of a checkerboard-corner intensity $E(x^2 - y^2)$, blurred with a Gaussian to represent a realistic image. In the right of this figure, the intensity is plotted in three-dimensions, to show that the checkerboard corner forms a saddle point.

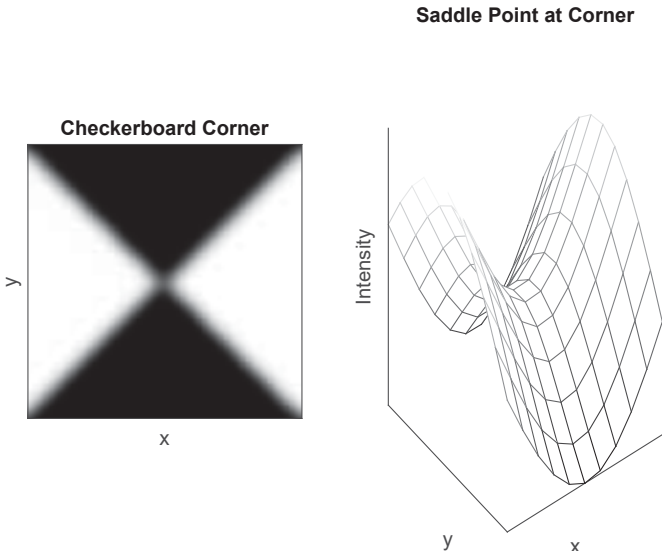

Fig. 5 Saddle point surface plot

The saddle point is located at the intersection of two lines along the checkerboard edges. We can estimate the saddle point location by fitting lines to the edges and finding the intersection of the lines [18]. We could also model a local X-corner, or saddle point, by considering two lines, given as:

$$\begin{aligned} l_1(x,y) &= a_1x + b_1y + c_1 \\ l_2(x,y) &= a_2x + b_2y + c_2 \end{aligned} \quad (2)$$

The X-corner is given by the product of the lines used as the argument of the edge function. The process is illustrated in Fig. 6. The top left image shows, $E(l_1(x,y))$, which could be color-coded as red, and the top right image shows, $E(l_2(x,y))$, perhaps color-coded as green. The bottom left image shows the four combinations of colors $(0G, RG, R0, 00)$. The bottom

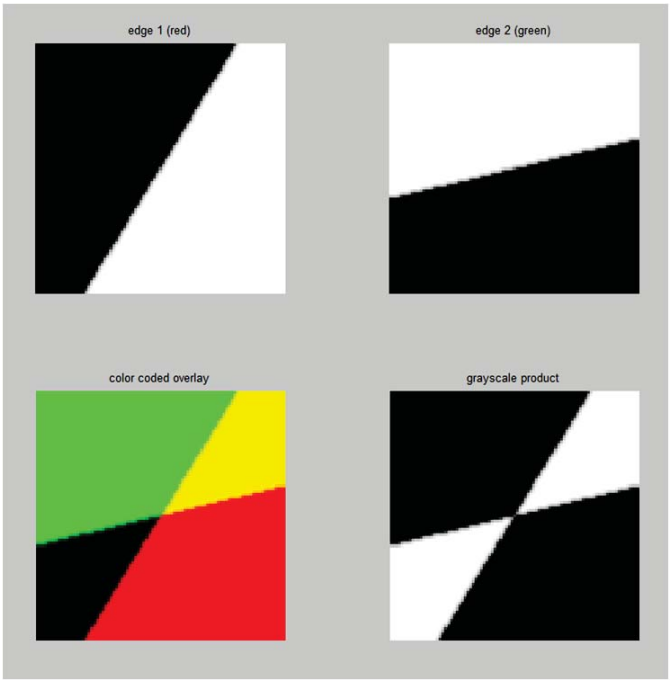

Fig. 6 Formation of X-corners from two edges

right image shows the X-corner The argument of the edge function given by the product of the lines can be reduced to a general quadratic as shown below.

$$\begin{aligned} p(x,y) &= l_1(x,y)l_2(x,y) \\ &= k_5x^2 + k_4y^2 + k_3 2xy + k_2y + k_1x + k_0 \end{aligned} \tag{3}$$

The corner point may be found from the inflection or saddle point of this polynomial. The solution is obtained by setting the derivatives in x and y equal to zero and solving the resulting set of linear equations, or more concisely by solving the following:

$$\begin{aligned} \nabla p(x,y) &= 0 \\ 2\mathbf{H}\begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{G} &= 0 \end{aligned} \tag{4}$$

where $\mathbf{H} = \begin{pmatrix} k_5 & k_3 \\ k_3 & k_4 \end{pmatrix}$ and $\mathbf{G} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}$

If a local region surrounding a pixel is examined, the polynomial may be interpreted as a Taylor series, **G** is the gradient vector and **H** is the Hessian matrix. If the eigenvalues of **H** are of opposite sign, the local region is a saddle point. If the eigenvalues have the same sign the local region is elliptical or hill-like, and if one eigenvalue is zero the local region is a ridge line. Hills and lines may be either bright on a dark background or dark on a bright background. X-corners may also be described in geometric terms by defining angles θ and ϕ as shown in Figure 7. The angle θ defines the orientation of the X-corner and ϕ defines the width of the X-corner.

The edge lines are given by

$$\begin{aligned} l_1(x,y) &= \sin(\theta+\phi)x - \cos(\theta+\phi)y \\ l_2(x,y) &= -\sin(\theta-\phi)x + \cos(\theta-\phi)y \end{aligned} \tag{5}$$

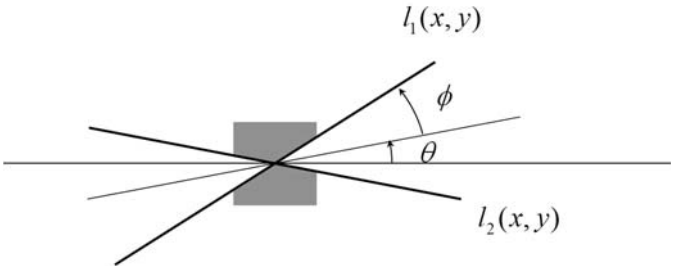

Fig. 7 Geometric interpretation of X-corners

After some trigonometric manipulation the corresponding quadratic obtained from the product of the two lines is given by:

$$\begin{aligned} e(x,y) = &-\cos(2\phi)(x^2+y^2) + \cos(2\theta) \\ &(x^2-y^2) + \sin(2\theta)(2xy) \end{aligned} \tag{6}$$

The general polynomial is given by:

$$\begin{aligned} p(x,y) = &c_5(x^2+y^2) + c_4(x^2-y^2) + c_3(2xy) \\ &+ c_2y + c_1x + c_0 \end{aligned} \tag{7}$$

where $c_5 = -\mathbf{K}\cos(2\phi)$, $c_4 = \mathbf{K}\cos(2\theta)$,$c_3 = \mathbf{K}\sin(2\theta)$ and **K** is a general constant which can be obtained from $\mathbf{K} = \sqrt{((c_4)^2 + (c_3)^2)}$

Fig. 8 shows the effect of varying c_5:$|c_5| < K$ corresponds to a saddle-point,$|c_5| = K$ corresponds to a ridge line and $|c_5| > K$ corresponds to an elliptical spot (hill).

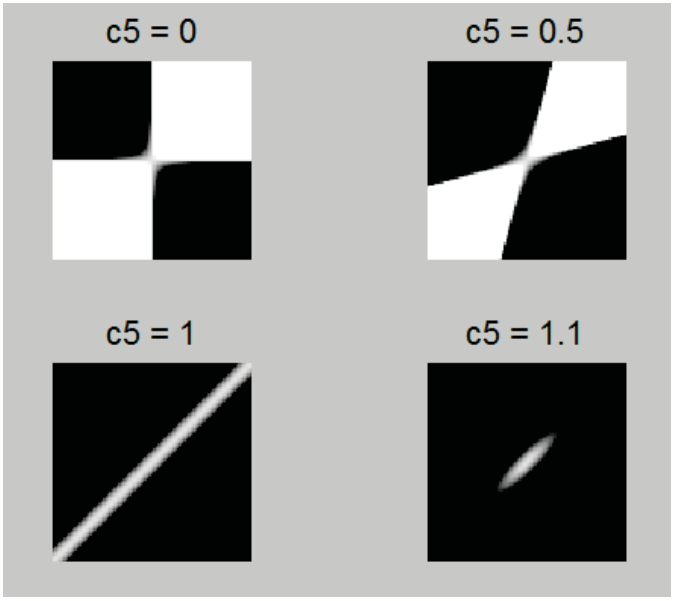

Fig. 8 Effect of varying c5, keeping c3 = 1 and c4 = 0

IV. Experimental Results

We can test the performance of the proposed algorithm using acquired images that vary in focal length, pose, and illumination. Fig. 9 is a test image with three checkerboard patterns at different orientations. All of the true x-corner points are indicated.

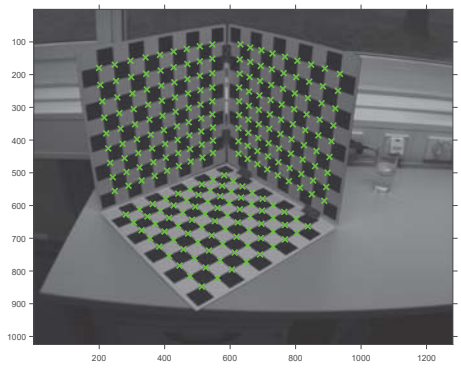

Fig. 9 Test image with detected x-corners.

The Harris corner detector is applied to the same image, and the results are indicated in Fig. 10. As implemented in Matlab, the Harris detector requires the user to provide the total number of X-corners, and the default value of 200 was used. As can be seen in the figure, the Harris detector does not find all of the true X-corners, and some spurious corners were found.

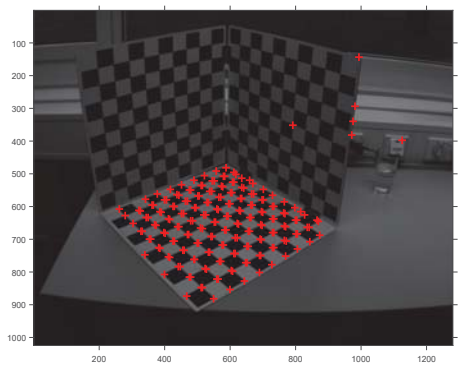

Fig. 10 Corner points located by the Harris detector

One particular checkerboard corner is magnified in Fig. 11, indicating that the algorithm accurately identifies the true corner. The corner is at the intersection of two lines along the checkerboard edges, as shown.

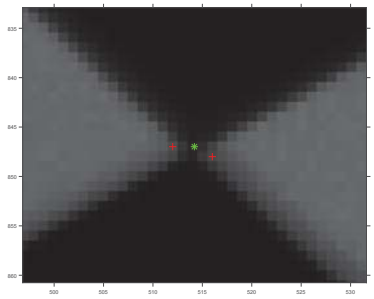

Fig. 11 One magnified x-corner with sub-pixel location. Green '*' indicates corner detected using proposed method, red '+' indicates corner detected using Harris detector

When one X-corner is magnified, as in Fig. 11, a common problem with the Harris detector occurs. This problem is that one X-corner is detected as two separate corners, formed from opposite white squares in the checkerboard pattern. The Harris detector operates as a general corner detector, not as an X-corner detector. We acquired another checkerboard image with different corner orientations. Fig. 12 shows the image with all the true x-corner indicated.

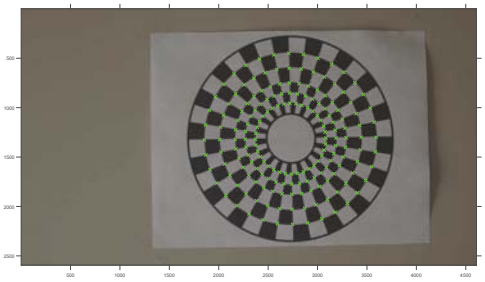

Fig. 12 X-corner of checkerboard at different orientations

Fig. 13 shows zoomed part of the checkerboard with all true x-corner indicated.

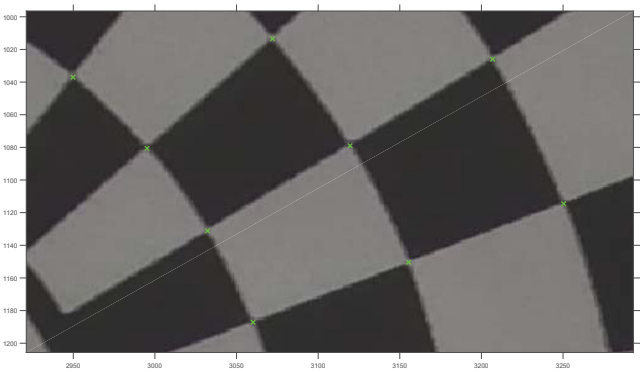

Fig. 13 Zoomed part of the checkerboard

Figs. 14, 15 indicate an array of checkerboard images taken at different orientations, with the algorithm's X-corners indicated in each. The algorithm successfully detects only the true X-corners, and does not return any spurious points.

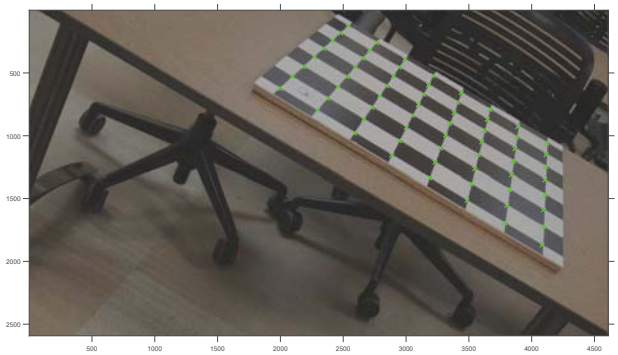

Fig. 14 Image of a checkerboard where X-corners are marked as '*'

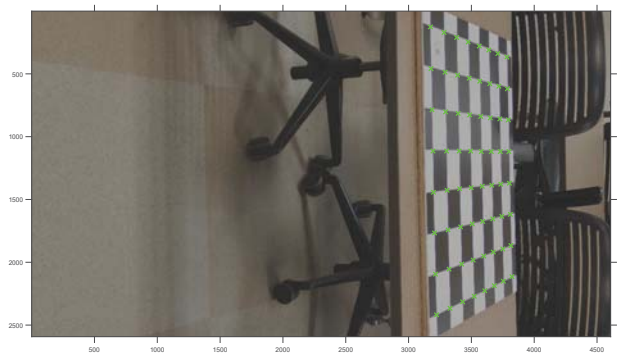

Fig. 15 Image of a checkerboard where X-corners are marked as '*'

V. Conclusions

This paper presented an automatic checkerboard corner detection, which uses a quadratic model for each X-corner and treats them as saddle points. The algorithm does not require the user to provide any information regarding the image, and a single threshold value can work for a wide array of images. Results indicate that it is robust to image noise, light levels, distance, and target orientation. The algorithm only uses texture information, and it can be combined with geometric properties of checkerboard patterns to easily eliminate any spuriously detected points.

References

[1] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986*, 1986.

[2] G.-Q. Wei and S. Ma, "A complete two-plane camera calibration method and experimental comparisons," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*. IEEE, 1993, pp. 439–446.

[3] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 965–980, 1992.

[4] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[5] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[6] Z. Wang, W. Wu, X. Xu, and D. Xue, "Recognition and location of the internal corners of planar checkerboard calibration pattern image," *Applied mathematics and computation*, vol. 185, no. 2, pp. 894–906, 2007.

[7] G. Wang, H.-T. Tsui, Z. Hu, and F. Wu, "Camera calibration and 3d reconstruction from a single view based on scene constraints," *Image and Vision Computing*, vol. 23, no. 3, pp. 311–323, 2005.

[8] G. Wang, H.-T. Tsui, and Q. J. Wu, "What can we learn about the scene structure from three orthogonal vanishing points in images," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 192–202, 2009.

[9] S. Placht, P. Fürsattel, E. A. Mengue, H. Hofmann, C. Schaller, M. Balda, and E. Angelopoulou, "Rochade: Robust checkerboard advanced detection for camera calibration," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 766–779.

[10] L. Lucchese and S. K. Mitra, "Using saddle points for subpixel feature detection in camera calibration targets," in *Circuits and Systems, 2002. APCCAS'02. 2002 Asia-Pacific Conference on*, vol. 2. IEEE, 2002, pp. 191–195.

[11] H. P. Moravec, "Towards automatic visual bbstacle avoidance," in *International Conference on Artificial Intelligence (5th: 1977: Massachusetts Institute of Technology)*, 1977.

[12] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of computer vision*, vol. 37, no. 2, pp. 151–172, 2000.

[13] S. M. Smith and J. M. Brady, "Susana new approach to low level image processing," *International journal of computer vision*, vol. 23, no. 1, pp. 45–78, 1997.

[14] X. Gao, F. Sattar, and R. Venkateswarlu, "Multiscale corner detection of gray level images based on log-gabor wavelet transform," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 7, pp. 868–875, 2007.

[15] A. Willis and Y. Sui, "An algebraic model for fast corner detection," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2296–2302.

[16] D. Chen and G. Zhang, "A new sub-pixel detector for x-corners in camera calibration targets," 2005.

[17] D. Parks and J.-P. Gravel, "Corner detection," *URL http://www. cim. mcgill. ca/˜ dparks/CornerDetector/harris. ht*, 2004.

[18] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-Hill New York, 1995, vol. 5.