

# Topological Detection of Chessboard Pattern for Camera Calibration

Gustavo Teodoro Laureano<sup>1</sup>, Maria Stela Veludo de Paiva<sup>1</sup> and Anderson Soares da Silva<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, University of São Paulo (USP/EESC), São Carlos, São Paulo, Brazil

<sup>2</sup>Institute of Informatics, Federal University of Goiás (UFG), Goiás, Brazil

**Abstract**—*Most works on camera calibration are directed to the stage of parameter estimation, while the phase matching is not always addressed. Most of applications assume that the correspondences are established in advance or require user intervention. Since the automatic applications require that the entire pattern is detected, which is difficult in most cases. This work aims to identify patterns of camera calibration automatically where the pattern can not be fully detected. Therefore, a corner detection and a topological filter are presented. The correspondence is done using neighboring properties on a geometric mesh and the sub-pixel location is threshold independent. The results show that the algorithm provides a robust detection even when the pattern is partially occluded.*

**Keywords:** Camera Calibration, Chessboard Detection, Topological Detection

## 1. Introduction

The camera calibration aims to determine the geometric parameters of the image formation process [1]. This is a crucial step in computer vision applications especially when metric information about the scene is required. In these applications the camera is generally modeled with a set of intrinsic parameters (focal length, principal point, skew of axis) and your orientation is expressed by extrinsic parameters (rotation and translation). Both intrinsic and extrinsic parameters are estimated by linear or non-linear methods using known points in the real world and their projections in the image plane [2]. These points are presented as a calibration pattern with known geometry, usually a flat chessboard.

Many studies have given attention to the camera calibration area, most of them are dedicated to the parameters estimation phase and refinement location of the calibration points [3], [4], [5], [6]. Tsai [7] and Zhang [8] are examples of the most cited papers related to this area. They propose closed form solutions for the estimation of intrinsic and extrinsic parameters using 3D and 2D calibration patterns respectively. Hamayed [9] and Salvi et al. [10] present reviews about some related works.

Camera calibration is a much discussed topic but the lack of robust algorithms for features detection difficults the construction of automatic calibration process. Calibration pattern recognition is a hard task, where the lighting problems and high level of ambiguities are the principal challenges. For this reason, the algorithms often require user intervention for a reliable detection of the calibration points. The hand tuning of points is tedious, imprecise and require user skill [11].

Some tools for automatic camera calibration are available. The Bouguet MatLab Toolbox [12] implements a semi-automatic calibration process. The application asks the user to define four extreme points that represent the area where an algorithm searches for the calibration points, given the number of rows and columns of the pattern. The OpenCV library [13] is a very popular computer vision library that offers an automatic way to detect chessboard patterns in images by the *findChessboardCorners()* function. The method performs successive morphological operators until a number of black and white contours be identified, subsequently the corners of the contours make up the calibration point set. The pattern is recognized only if all rectangles are identified. In an online system this restriction causes a considerable loss of image frames, since is not always possible to detect all the chessboard rectangles.

Fiala and Shu [14] use an array of fiducial markers, each one with a unique self-identifying pattern. The described methodology is robust to noise and it is not necessary to identify the entire calibration pattern. In the other hand, the markers are complex and require a special algorithm to recognize them.

Escalera and Armingol [15] identify the calibration points as the intersections of lines. The methodology uses a combined analysis of two consecutive Hough transforms to filter the collinear points inside the pattern. The assumption that all points of interest are collinear makes this algorithm very sensitive to distortions, limiting its use only to cameras with low radial distortion.

The system named CAMcal [16] uses the Harris corner detection and a topological sort of squares within a geometric mesh. Harris corner detection is time consuming, sensible to noise, needs an empirical threshold to select

interesting points and does not produce good results to the specific features of the chessboard image [17]. Furthermore, the system must to detect three circles to determine orientation of the pattern.

This work presents a system for automated detection of chessboard patterns for camera calibration. Initially a fast and specific x-shaped corner operator is performed to retrieve the interesting points. A geometric mesh is created from all the x-corners by Delaunay triangulation. A topological filter is proposed. Are taken as valid the triangles that match with the regularity of the pattern. The color and the neighborhood of the triangle are analyzed. Each remaining point defines a valid x-corner and a refinement location is performed locally.

The calibration process does not depend on full detection of the calibration pattern. When a minimum number of points is identified the calibration algorithm may be executed, in this case the Zhang’s algorithm [8].

## 2. X-Corner Detector

The first stage of the algorithm is the features detection. Corners x-shaped are identified analyzing the alternations of high contrast in the neighborhood of each pixel.

Considering  $V = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$  the neighborhood of a central pixel  $\mathbf{p}_c$ , defined by all the pixels in the border of a Breseham’s circle [18], the number of alternations of high contrast is computed by the Equation 1.

$$N_{alt} = \sum_{i=1}^n \begin{cases} 1, & I(\mathbf{p}_i) > T_h \ \& \ I(\mathbf{p}_{i-1}) < T_l \\ 1, & I(\mathbf{p}_i) < T_l \ \& \ I(\mathbf{p}_{i-1}) > T_h \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where  $\mathbf{p}_i \in V$ ,  $I(\mathbf{p}_i)$  represents the pixel intensity of  $\mathbf{p}_i$ ,  $T_l$  and  $T_h$  are the inferior and superior threshold respectively. Alternatively, both thresholds can be defined by:  $T_l = m - gate$  and  $T_h = m + gate$ , with  $m = \frac{1}{n} \sum_{i=1}^n I(\mathbf{p}_i)$ .

The pixel  $\mathbf{p}_c$  is classified as a x-corner if  $N_{alt} = 4$  and  $T_l < I(\mathbf{p}_c) < T_h$ . For the Equation 1, if  $i = 0$ ,  $i - 1 = n$  is assumed. The Figure 1 shows the considered area by this detector.

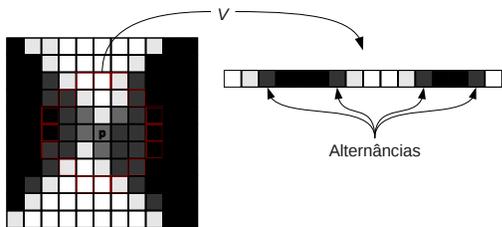


Fig. 1: Typical x-corner neighborhood.

The variable *gate* models the operator sensibility. Considering a previously blurred image, the number of

alternations imposes large part of the restriction required for a proper classification. Thus the variable *gate* has little effect on the final result. In this work *gate* is defined with 10 empirically.

This detector can be seen as a specification of the proposed detector in Rosten and Drummond [19], which is considered high performance. Since only a small portion of the neighborhood of the pixel is analyzed, the computational cost of this operation is reduced. Another similar detectors can be found in Zhao et al. [17] and Sun et al. [20]. The Figure 2 shows a typical result of this detector over the original image.

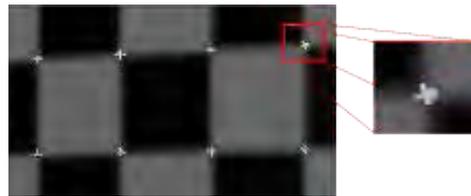


Fig. 2: X-corner operator response.

The formulation of this operator does not guarantee that only one pixel is classified as a x-corner in its neighborhood. To deal with this problem, the cost described by the Equation 2 is associated with each corner and a non-maximum suppression is performed [21].

$$\max \left( \sum_{\mathbf{p}_i \in \text{dark}} |I(\mathbf{p}_i) - m|, \sum_{\mathbf{p}_i \in \text{light}} |I(\mathbf{p}_i) - m| \right) \quad (2)$$

The classes *dark* and *light* contains the dark and light pixels respectively. The right corner is the one with the highest associated cost.

## 3. Topological Filter

The identification of valid corners is an important step because not all x-corners present in the image belong to the calibration pattern. In this work, the identification of valid x-corners is made considering the regularity neighborhood of the chessboard image. This problem can be extended to the problem of creating geometric meshes in computer graphics. In a mesh composed of basic components such as triangles, vertices are connected according to their neighborhood [22].

The Delaunay triangulation is a classic problem in computational geometry. Given a set of points in a plane, the only valid triangulation is one where the circumcircle of each triangle contains no other vertex [23]. This feature ensures that the triangles are formed by the more closely vertexes. Guibas et al. [24] present an algorithm for incremental triangulation that that runs in time  $O(n \log(n))$ .

With the creation of the mesh the neighborhood of each feature is defined. Figure 3 gives an example of this triangulation.

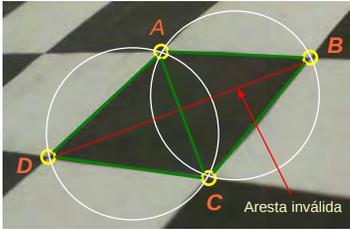


Fig. 3: Considering  $A$ ,  $B$ ,  $C$  and  $D$  four image corners, the valid triangulation is formed by the triangles  $\Delta(A, B, C)$  and  $\Delta(A, C, D)$ .

Using the geometric mesh, the vertices and triangles are submitted to a topological filter to exclude those not satisfying the regularity of the pattern. The corners (or vertexes) share internal triangles of different colors in a regular manner. Each square of the chessboard pattern is represented by two triangles of the same color. Each triangle has no more than two neighboring triangles that form two squares with different colors alike. The internal vertexes have in common a maximum of eight triangles. Valid triangles have its interior filled with a single color.

Even after the projected image plane, the neighborhood relationship between the corners is still maintained. This restriction allows us to evaluate if the corners really belong to the calibration pattern. Thus, they are considered valid:

- 1) those triangles that do not have color transitions in your interior;
- 2) only those triangles that have a neighbor with the same color;
- 3) those triangles that have only two neighbors of the same color and different color triangle taken as a reference;

This filter is applied to the grid until there are no more invalid triangles. In the end, the vertices that do not form any triangle are also removed.

To avoid the use of thresholds in the comparison of colors, this filter uses a binarized version of the image. This is an important step in validating points. If binarization fails, noisy points can be identified and actual points can be disregarded. To minimize these effects this work uses adaptive binarization described in the work of Bradley and Roth [25]. This algorithm handles well with large variations in illumination and runs in linear time for any window size.

The binarization phase can be influenced by problems from the acquisition of images due to lighting variations and also by the fluctuation of the intensities of the pixels. In the regions near to the edges a range of values may be

wrongly considered black or white pixels. This behavior can generate white triangles with black borders and black triangles with white edges. In practice, verification color transition is made in a region of the innermost triangle, ignoring the edges. Figure 4 shows the result of the topological filtering.

## 4. Point Correspondences

The next step of the algorithm associates each vertex to the real coordinates of the pattern. This is done by analyzing the relative position of each corner. First two neighboring triangles of the same color are arbitrarily selected:  $T_1$  and  $T_2$ . Three vertices make up the triangle  $T_1$ , the origin of the coordinate system is defined by the vertex that has  $T_2$  as its opposite triangle. For the remaining vertices are assigned the directions  $x$  and  $y$  of the Cartesian plane (Figure 5).

The propagation of coordinates consists in establishing the relative coordinates of the vertices neighbors. Given a triangle  $T$  whose vertices have already defined coordinates, where the origin is  $v_o$ ,  $v_x$  and  $v_y$  are the vertices with the  $x$  and  $y$  directions respectively.  $T_v$  is defined as a neighbor triangle of  $T$  with a different color. If  $T_v$  and  $T$  are neighbors then they share an edge  $e$  and  $T_v$  has a opposite vertex to the  $T$ , called  $v_v$ . The coordinates of the opposite vertex needs to be determined, thus:

- If  $v_x \in e$ , then  $v_v = [v_t^{(x)} \quad 2v_t^{(y)} - v_y^{(y)}]'$ ;
- If  $v_y \in e$ , then  $v_v = [2v_t^{(x)} - v_x^{(x)} \quad v_t^{(y)}]'$ ;

It is understood by  $v^{(\cdot)}$  the coordinate  $(\cdot)$  of vertex  $v$ . Similarly,  $T_{op}$  shares a border  $e_{op}$  with  $T_v$ , then  $v_{op} = [v_h^{(x)} \quad v_v^{(y)}]'$ , where  $v_h$  is the third vertex of  $T_v$  and  $v_{op}$  is the opposite vertex to  $e_{op}$ .

For each visited triangle, the vertexes coordinates of the current and opposite triangles are propagated. The algorithm performs recursively for each neighbor triangle to the pair  $T_v$  and  $T_{op}$ . It makes the algorithm  $O(\frac{n}{2})$ , where  $n$  is the number of triangles in the mesh.

## 5. Location Refinement

The use of the detector described in section 2 identifies the position of corners with low accuracy where the only information available is the position of discrete pixels. Since the quality of the calibration is directly dependent to the precision with which the position of features is found, there is a need for a technique refinement [1].

Traditional algorithms such as Harris and Stephens detector [26] and Shi and Tomasi [27], run throughout the image and use thresholds to select the features of interest. The sub-pixel precision is achieved by maximizing functions fitted to the square of the intensity profile of the local neighborhood of each pixel. The threshold has a direct impact on the quality of response of these

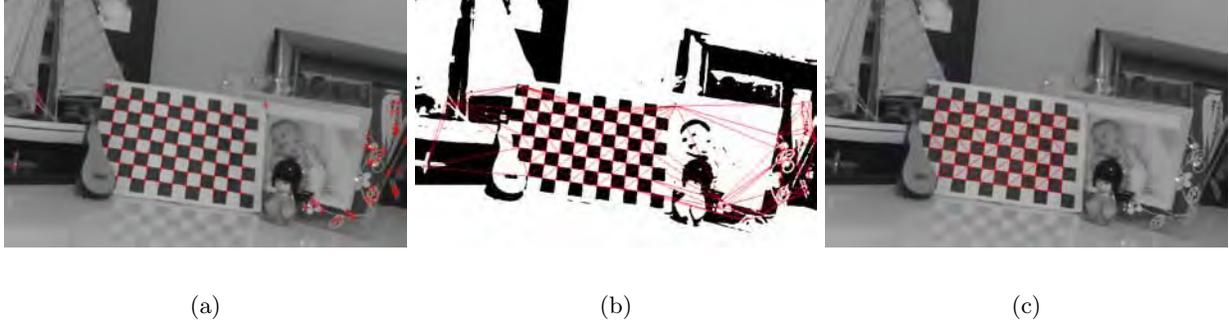


Fig. 4: Example of the detection and topological filter results. a) X-corners. b) Triangulation and binarized image. c) The valid triangles after the topological filter.

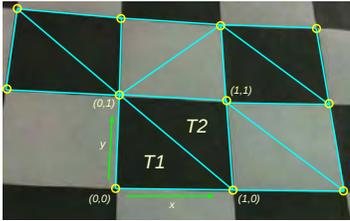


Fig. 5: The triangles  $T1$  and  $T2$  define the origin and the direction of coordinates.

detectors, so corners are usually classified as the  $N$  pixels with greater response to the operator.

Chen et al. [4] propose a new detector specially designed to fit corners of X-shape. Considering the neighborhood of a pixel as a surface, its Hessian matrix can be expressed as:

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (3)$$

where  $I_{xx}$ ,  $I_{xy}$  and  $I_{yy}$  are the second partial derivatives of pixel  $I(x, y)$ . Thus, the x-corner detector is described as:

$$S = \lambda_1 \cdot \lambda_2 = I_{xx}I_{yy} - I_{xy}^2. \quad (4)$$

where,  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of  $H$ .

In order to avoid unnecessary computation, this operator is only applied in regions defined by the valid vertices of triangle mesh. The x-corner can be detected by identifying the largest negative value of  $S$  and the refined coordinates  $(x_0 + s, t + y_0)$  is given by:

$$s = \frac{I_y I_{xy} - I_x I_{yy}}{I_{xx} I_{yy} - I_{xy}^2}, \quad t = \frac{I_x I_{xy} - I_x I_{xx}}{I_{xx} I_{yy} - I_{xy}^2} \quad (5)$$

## 6. Experimental Results

In this section, the detector response is evaluated considering an image database and by means of experimental tests with two different cameras.

The image database is provided by the toolbox for MatLab prepared by Bouguet [12]. This database consists of 20 images of a chessboard calibration pattern, accounting 156 x-corners arranged as  $12 \times 13$  matrix, being presented in different orientations. This set represents a common situation to most of systems where the pattern images are first captured and calibration is performed in an offline manner.

Figure 6 shows some examples of these images and Table 1 summarizes the results obtained for each one. The results are generated by applying the algorithm in each image and counting the number of corners identified. For Table 1, the vast majority of points is detected.

The mean accuracy of the algorithm is 85.38% however two images (Image 5 and 18) deserve attention by the low percentage of success. They represent situations where the calibration plane is very inclined to the camera. In this case it is expected that the corners are uncharacterized by high perspective distortion and lack of focus in the image. Another aspect to be considered is that the images in this database have low contrast, which complicates the identification of alternations of high contrast.

In general, the algorithm was able to find the most calibration points. If the two worst images are discarded, the accuracy of success rises to 90.88%, which reflects the efficiency of the methodology. The Figure 7 shows the worst and best results of the algorithm.

For the online experiments, we used two different cameras: (1) Philips Webcam SPC990NC e (2) Microsoft Webcam HD 5000. The calibration pattern used is formed by squares with 2.5cm of width and forms a matrix of  $11 \times 7$  x-corners. For each camera, were tested 14 real images of the calibration pattern in various orientations and distances.

The algorithm runs on a sequence of captured frames. The amount of detected points, presented in the second and fourth column of Table 2 corresponds to the average of the points detected in 10 frames for each position of

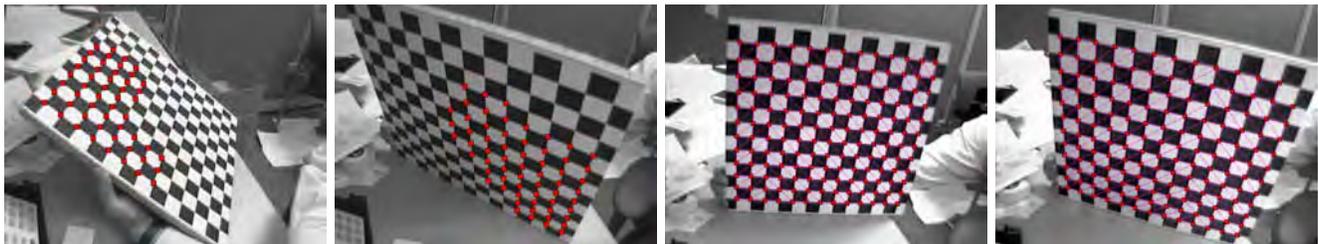


(a) Image 1      (b) Image 2      (c) Image 4      (d) Image 9      (e) Image 17      (f) Image 19

Fig. 6: Images of the Bouguet database.

Image	1	2	3	4	5	6	7	8	9	10
Corners	151	154	147	153	51	149	132	110	143	153
(%)	96.79	98.72	94.23	98.08	32.69	95.51	84.62	70.51	91.67	98.08
Image	11	12	13	14	15	16	17	18	19	20
Corners	152	155	156	138	121	154	155	61	111	118
(%)	97.44	99.36	100.00	88.46	77.56	98.72	99.36	39.10	71.15	75.64

Table 1: Results for the Bouguet database.



(a) Image 5      (b) Image 18      (c) Image 13      (d) Image 17

Fig. 7: Worst (Image 5 and 18) and best (Image 13 and 17) results in the Bouguet database.

	HD 5000		SPC900nc	
	x-corners	%	x-corners	%
Image 00	77	100	73	94.80
Image 01	77	100	77	100
Image 02	77	100	77	100
Image 03	73	94.80	76	98.70
Image 04	77	100	73	94.70
Image 05	77	100	77	100
Image 06	77	100	76	98.70
Image 07	77	100	77	100
Image 08	77	100	77	100
Image 09	77	100	76	98.70
Image 10	75	97.40	72	93.50
Image 11	70	90.90	76	98.70
Image 12	72	93.50	75	97.40
Image 13	70	90.90	77	100
Mean:		97.68		98.23

Table 2: Results for the online detection.

the calibration pattern.

The Figure 8 shows some of the images used in the second experiment. Once the pattern is completely visible, the algorithm has a high hit rate, while the missed corners tend to arise when there is a more accentuated

inclination of the plane in relation to the camera. In these images no false positives were identified, which confirms the robustness of the filter used.

To illustrate the efficiency of the filter topological and propagation of coordinates, Figure 9 illustrates the result of the algorithm using complex backgrounds and partial occlusion of the pattern. The occluded corners do not interfere in the propagation of correct coordinates. Thus, it is possible to use the maximum of features identified for the estimation of camera parameters. The last column shows the reprojection plan calibration calculated from the detected points.

## 7. Conclusions

This work proposes a methodology for detecting calibration patterns. The experimental results show that it is possible to detect these patterns in a robust and automatic without the use of thresholds. Furthermore, a low computational cost is achieved, since the refinement of X-corners is directed to specific regions of the image. The filtering of topological corners-X allows handle cameras with radial distortion and high immunity to noise. A

partial identification of the pattern allows the calibration process is considering giving maximum points detected. In conditions where few points are detected, most picture frames are utilized.

## References

- [1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [2] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- [3] A. Dutta, A. Kar, and B. N. Chatterji, "A novel window-based corner detection algorithm for gray-scale images," in *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, ser. ICVGIP '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 650–656.
- [4] D. Chen and G. Zhang, "A new sub-pixel detector for x-corners in camera calibration targets." in *WSCG (Short Papers)'05*, 2005, pp. 97–100.
- [5] X. Hu, P. Du, and Y. Zhou, "Automatic corner detection of chess board for medical endoscopy camera calibration," in *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, ser. VRCAI '11. New York, NY, USA: ACM, 2011, pp. 431–434.
- [6] L. Krüger and C. Wöhler, "Accurate checkerboard corner localisation for camera calibration," *Pattern Recogn. Lett.*, vol. 32, no. 10, pp. 1428–1435, July 2011.
- [7] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1986, pp. 364 – 374.
- [8] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330 – 1334, nov 2000.
- [9] E. Hemayed, "A survey of camera self-calibration," in *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, july 2003, pp. 351 – 357.
- [10] J. Salvi, X. ArmanguÃ, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognition*, vol. 35, no. 7, pp. 1617 – 1635, 2002.
- [11] S. Bennett and J. Lasenby, "Chess - quick and robust detection of chess-board features," *CoRR*, vol. abs/1301.5491, 2013.
- [12] J.-Y. Bouguet, "Camera calibration toolbox for matlab," [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).
- [13] OpenCV, "Open source computer vision library," <http://www.opencv.willowgarage.com/>.
- [14] M. Fiala and C. Shu, "Self-identifying patterns for plane-based camera calibration," *Machine Vision and Applications*, vol. 19, pp. 209–216, 2008, 10.1007/s00138-007-0093-z.
- [15] A. de la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration." *Sensors (Basel, Switzerland)*, vol. 10, no. 3, pp. 2027–44, Jan. 2010.
- [16] C. Shu, A. Brunton, and M. Fiala, "A topological approach to finding grids in calibration patterns," *Machine Vision and Applications*, vol. 21, pp. 949–957, 2010, 10.1007/s00138-009-0202-2.
- [17] F. Zhao, C. Wei, J. Wang, and J. Tang, "An automated x-corner detection algorithm(axda)," *JSW*, vol. 6, no. 5, pp. 791–797, 2011.
- [18] J. Bresenham, "A linear algorithm for incremental digital display of circular arcs," *Commun. ACM*, vol. 20, no. 2, pp. 100–106, Feb. 1977.
- [19] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the 9th European conference on Computer Vision - Volume Part I*, ser. ECCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 430–443.
- [20] W. Sun, X. Yang, S. Xiao, and W. Hu, "Robust checkerboard recognition for efficient nonplanar geometry registration in projector-camera systems," in *Proceedings of the 5th ACM/IEEE International Workshop on Projector camera systems*, ser. PROCAMS '08. New York, NY, USA: ACM, 2008, pp. 2:1–2:7.
- [21] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2nd ed. Academic Press ISBN: 978-0-12-372538-7, 2008.
- [22] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Computing in Euclidean geometry*, vol. 1, no. 1, pp. 23–90, 1992.
- [23] M. V. K. Mark De Berg, Otfried Cheong, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag ISBN: 978-3-540-77973-5, 2008.
- [24] L. Guibas, D. Knuth, and M. Sharir, "Randomized incremental construction of delaunay and voronoi diagrams," *Algorithmica*, vol. 7, pp. 381–413, 1992, 10.1007/BF01758770.
- [25] D. Bradley and G. Roth, "Adaptive thresholding using the integral image." *Journal of Graphics Tools*, pp. 13–21, 2007.
- [26] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.
- [27] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, jun 1994, pp. 593 –600.

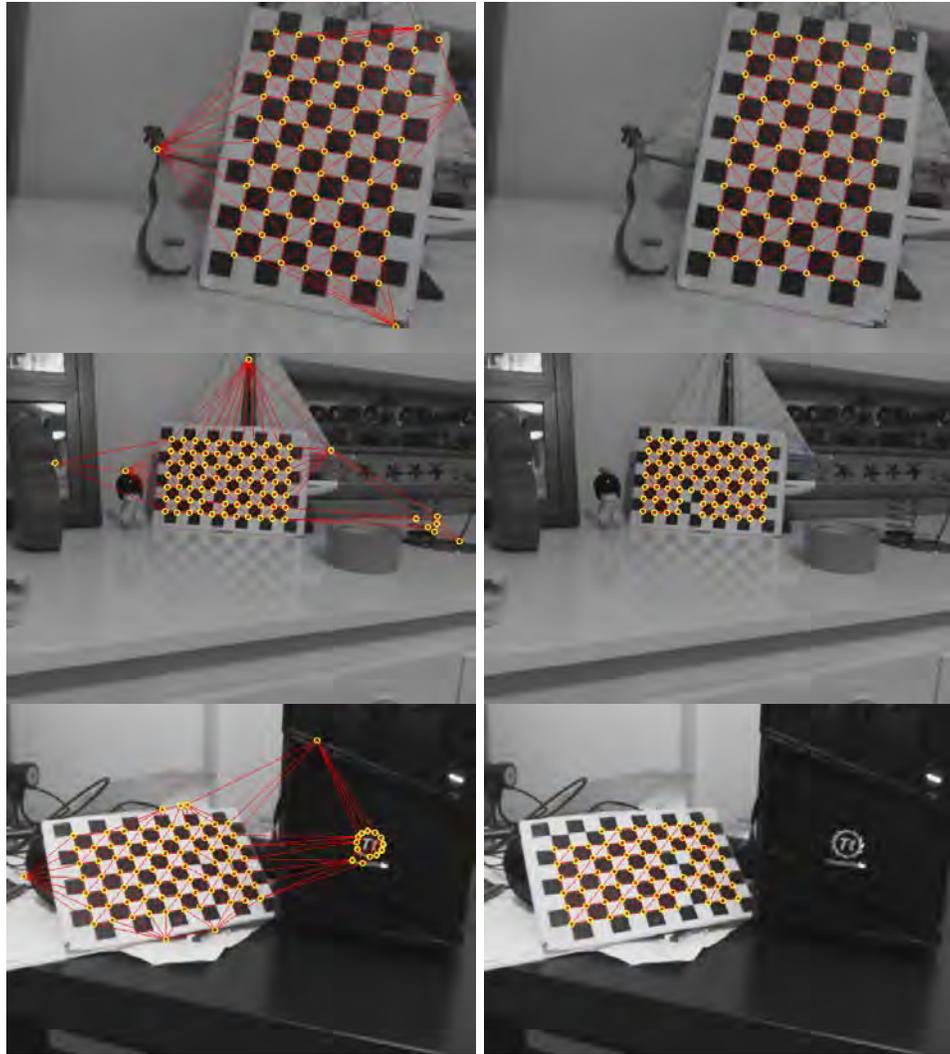


Fig. 8: Example of images used in the second test. The left column shows all x-cornes and the triangulation. The right column shows the topological filter result.

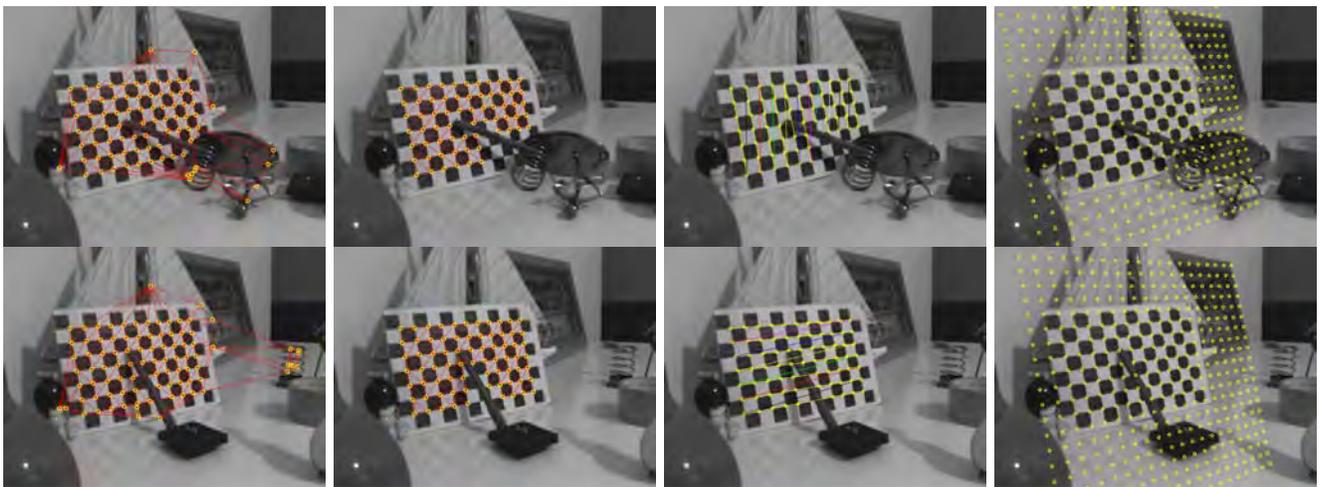


Fig. 9: Results with complex backgrounds and partial occlusion.