

# Periodic Patterns Recovery for Multicamera Calibration

Lorenzo Sorgi and Andrey Bushnevskiy

Technicolor R&I, 30625, Karl-Wiechert Allee 74, Hannover, Germany

**Keywords:** Multicamera Calibration, Pattern Recovery, Partial Calibration Pattern Visibility, Calibration Pattern Occlusion, External Calibration, Calibration Pattern Offset Removal.

**Abstract:** Camera calibration is an essential step for most computer vision applications. This task usually requires the consistent detection of a 2D periodic pattern across multiple views and in practice one of the main difficulties is a correct localization of the pattern origin and its orientation in case of partial occlusion. To overcome this problem many calibration tools require a full visibility of the calibration pattern, which is not always possible, especially when a multicamera systems are used. This paper addresses the specific problem of consistent recovery of the calibration pattern, captured by a multicamera systems under the condition of partial occlusion of the calibration object in several (even all) calibration images. The proposed algorithm is structured in two sequential steps aimed at the removal of the rotational and the translational components of the pattern offset transformation, which is essential for a correct calibration. The paper focuses on two common calibration patterns, the checkerboard grid and the bundle of parallel lines; however, the technique can be easily rearranged in order to cope with other classes of periodic patterns. The algorithm effectiveness has been successfully proven on the simulated data and two real calibration datasets, captured using a fisheye stereo rig.

## 1 INTRODUCTION

Camera calibration aims at estimation of the internal and the external parameters of an imaging system. The first group collects the parameters describing the geometry of the image formation process of each camera, such as focal length and distortion coefficients; the second group of parameters instead provides a representation of the position and orientation of each camera in a common reference frame. The external calibration of course implies a multicamera systems use.

The calibration pipeline usually starts with the collection of a set of 2D/3D correspondences by selecting in each image the corners corresponding to a set of known points in 3D space. In order to simplify this task, most calibration tools use an object with an a-priori known geometry, which can be easily detected within an image, either in fully automatic way or with a minimal user interaction. Different objects, such as 2D or 3D grids and 1D dotted rod, have been used for calibration (Zhang, 2000; Faugeras, 1993; Zhang, 2002); in practice now the 2D black and white checkerboard has become almost a standard, due to its simplicity and the high accuracy of the corners detection algorithms.

The main difficulty in the grid detection phase is the correct localization of the grid origin and the axis orientation. This problem naturally arises in case of a partial occlusion of the calibration object, due to its periodic structure.

In case of a single camera calibration the wrong localization of the grid reference system is not an issue (Kassir and Peynot, 2010). However, this does not hold when a rigid multicamera systems must be calibrated. In this case indeed the additional external calibration step, that is aimed at the estimation of the relative geometry between the different cameras of the cluster, requires the establishment of a set of *spatially consistent* correspondences across the multiple views of the calibration object. In particular it will be shown in the next sections that the correct recovery of the correspondences is important only across the images, simultaneously captured by the cameras in the multicamera rig. This is an easy task in case of full visibility of the calibration object; however, with the increase in the number of cameras comprised in the multiview system, the full visibility constraint becomes more complicated to satisfy.

In this work we propose a more flexible solution, which decouples the object detection phase from the recovery of the consistent set of correspondences,

suitable for the external calibration. This is done by introducing an additional stage in the typical calibration pipeline between the internal and the external calibration steps, aimed at the grid offset removal between the multiple camera views.

It will be shown that the set of 2D calibration points can be corrected by enforcing two implicit geometrical constraints in the metric camera space: the rigidity of the camera cluster and the discretized nature of the space from which the grid coordinate offset is sampled. We will then show in practice how the algorithm can be applied to the offset correction of two common calibration patterns: a checkerboard grid and a bundle of parallel lines.

## 2 RELATED WORK

Most of the existing calibration tools require the full visibility of the calibration pattern in the each image (Tsai, 1986; Bouguet, 2013; OpenCV, 2014; Wedekind et al., 2013; Douskos et al., 2009; Heikkila, 2000; Datta et al., 2009; Vo et al., 2011). This solution noticeably reduces the complexity of the grid detection task at the price of a lower calibration accuracy. Indeed, the enforcement of the full visibility constraint implies, that the periphery of the image is only weakly spanned by the grid, whereas a strong calibration requires points broadly spread across the whole retinal plane.

Other tools use a pre-marked calibration object in order to establish the spatially consistent correspondences. (Vaish, 2006; Shu et al., 2003; CAMcal, 2014) for example use a grid with the integrated circle pattern, (Atcheson et al., 2010; Fiala and Shu., 2008) embed markers with a unique visual ID that can be used for resolving the multiview offset ambiguity. However, this approach increases the complexity of the 2D calibration pattern as well as the whole calibration pipeline, introducing an additional approach specific marker detection step.

(Agisoft Lens, 2014; Oyamada et al., 2012) calibration tools are designed in a more flexible way and capable of handling partially occluded views. However, those tools are aimed at the single camera calibration and can not be used with multicamera systems.

In our opinion, a robust calibration tool should be able to cope with partially occluded views of the calibration object, arbitrarily oriented with respect to the cameras in the acquisition cluster. For this reason we propose the offset removal stage between the internal and the external calibration, which allows the system to handle the partial occlusion of the calibration ob-

ject without any pre-marking. Despite the simplicity of the solution we claim that this provides a noticeable relief of the needed user interaction and greatly increases the usability of the calibration tool.

## 3 SYSTEM OUTLINE

Let us denote with  $\mathbf{M}$  a point in 3D space and with  $\mathcal{G}_{w,c} = (\mathcal{R}_{w,c}, \mathbf{T}_{w,c}) \in SE(3)$ , a rigid transformation between the reference coordinate system  $w$  and the one aligned with the  $c$ -th camera axis, where  $SE(3)$  is the space of the rigid Euclidean transformations.

If the coordinates of  $\mathbf{M}$  are expressed with respect to  $w$ , then projection from the 3D space onto the camera retinal plane is described by the equation

$$\mathbf{m} \sim \mathcal{K} \cdot nl(\mathcal{G}_{w,c} \circ \mathbf{M}; \theta_d), \quad (1)$$

where  $\mathcal{G} \circ \mathbf{M} = \mathcal{R}\mathbf{M} + \mathbf{T}$  is the coordinate transformation induced by  $\mathcal{G}$ ,  $\mathcal{K}$  is a camera matrix and  $nl(\mathbf{x}; \theta_d)$  is the non linearity, parameterized by the vector  $\theta_d$ , that models the lens distortion (Hartley and Zisserman, 2000).

The Internal Calibration is the process of estimating  $\mathcal{K}$  and  $\theta_d$  using a set of 2D/3D correspondences  $\{\mathbf{M}_i, \mathbf{m}_i\}_{i=1\dots N}$ . Usually the 2D points are extracted from one or multiple views of an ad-hoc designed calibration object, such as a checkerboard and the reference coordinate system  $w$  is implicitly defined by the calibration object pattern itself, for example assuming the origin in one of the corners of the grid and the axes are aligned with the grid directions.

A typical issue arises when the calibration object is partially occluded in some image. In this case the grid origin and the axis orientation cannot be unambiguously retrieved from the image content, and the correspondences with the 3D coordinates of the object can be defined only up to a common roto-translational offset. We denote this offset as  $\delta\mathcal{G} = (\delta\mathcal{R}, \delta\mathbf{M}) \in SE(3)$ , where  $\delta\mathcal{R}$  is a rotational component and  $\delta\mathbf{M}$  is a translational component of the offset. As usually the grid detection is run independently in each image, we assume that the extracted patterns are affected by a different offsets.

The space of feasible offsets affecting the point coordinates of an image is a discrete subspace of the Euclidean transformation space and depends on the specific symmetry of the pattern. For example, it is easy to see that for a 2D grid (Fig. 1(a)) the feasible offset can be defined as

$$\begin{cases} \delta\mathbf{M} = [\delta X \ \delta Y \ 0]^T : (\delta X, \delta Y) \in \mathbb{Z}^2 \\ \delta\mathcal{R} = \mathcal{R}_Z(\theta) : \theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\} \end{cases}, \quad (2)$$



Figure 1: Periodic calibration patterns, the checkerboard (a) and the parallel line bundle (b).

where  $\mathcal{R}_Z(\theta)$  denotes a 3D rotation about the Z axis. A different example is given by the line bundle pattern shown in Fig. 1(b). Its 1D symmetry implies  $\theta \in \{0, \pi\}$  and  $\delta X \in \mathbb{R}$ . In general the appropriate offset space (2) can be defined for every periodic pattern and the proposed grid recovery algorithm remains unchanged.

Let us denote with  $\hat{\mathbf{M}} = \delta \mathcal{G} \circ \mathbf{M}$  the point of the calibration pattern in 3D space, affected by the offset (2). By integrating the previous equation into the projection equation (1), one can see that the exploitation of the corrupted 3D grid points  $\hat{\mathbf{M}}$  leads to an incorrect camera pose estimation given by:

$$\hat{\mathcal{G}}_{w,c} = \mathcal{G}_{w,c} \circ \delta \mathcal{G}_c^{-1}, \quad (3)$$

where  $\delta \mathcal{G}_c$  is the offset for the  $c$ -th camera.

We can then infer that the improper identification of the pattern from a set of calibration images implies a wrong estimate of the camera pose corresponding to each view, but does not interfere with the estimation of the internal parameters. This is the reason why the full visibility constraint can be simply dropped for the purpose of the single camera calibration.

For a cluster of cameras rigidly mounted on a common chassis, the external calibration, comprised of the pose of each camera within the cluster, must be estimated as well. If we align the cluster reference system with the first camera, identified by the index 0, then the external calibration of the  $c$ -th camera can be computed by sequentially combining the pose transformations (3) of the two cameras with respect to the calibration object:

$$\hat{\mathcal{G}}_{0,c} = \mathcal{G}_{w,c} \circ \mathcal{G}_{w,0}^{-1}, \quad (4)$$

where  $\mathcal{G}_{w,0}$  is a transformation between reference coordinate system  $w$  and the coordinate system of the first camera 0. Using the offset-affected camera poses (3) to estimate  $\hat{\mathcal{G}}_{0,c}$  one obtains

$$\hat{\mathcal{G}}_{0,c} = \mathcal{G}_{w,c} \circ \delta \mathcal{G}_c^{-1} \circ \delta \mathcal{G}_0 \circ \mathcal{G}_{w,0}^{-1}, \quad (5)$$

where  $\delta \mathcal{G}_0$  is the offset of the first camera.

The equation (5) shows that for a reliable external calibration the offset affecting two views do not have

to be necessarily removed. It is sufficient to equalize the offsets to remove them from the eq. (5). In a practical calibration scenario, when several multiview captures are processed, the pattern origin and orientation extracted from the reference camera images can be considered offset-free and the offset in the views of the other cameras has to be accordingly corrected. This is a sufficient correction for a reliable external calibration. The residual offset affecting the reference view implies only a wrong estimate of the absolute pose of the cluster corresponding to each snapshot, which is anyway outside of the scope of the external calibration.

Following these two observations, we designed a multicamera calibration tool comprised of four phases: calibration object detection, internal calibration, object pose offset correction and external calibration. The first step extracts the calibration object independently from each image, without any multi-view consistency check. The second step estimates the internal parameters of each camera. The third step corrects the offset of the pattern coordinates, making it constant within each set of simultaneous captures and the last step estimates the cluster multiview geometry in a unique optimization framework.

This paper focuses specifically on the third step, which up to our experience has never been addressed. We claim that it plays an important role in the design of a user-friendly calibration tool as it allows a successful calibration of a multicamera system relying only on the knowledge of the pattern topology; the knowledge of the pattern size and the full visibility in each image are not required anymore.

In the next sections we show how the object offset can be corrected in a two consecutive phases, addressing the reduction of the rotational and the translation components of the offset from each calibration view.

### 3.1 Rotational Offset

Rewriting the rotational component of the  $c$ -th camera pose using the transfer via the reference camera (5) together with eq.(3) we obtain:

$$\mathcal{R}_{w,c} = \mathcal{R}_{0,c} \mathcal{R}_{w,0} = \hat{\mathcal{R}}_{w,c} \delta \mathcal{R}_c, \quad (6)$$

where  $\delta \mathcal{R}_c$  is the rotational component of the offset in the  $c$ -th camera view. By definition (2)  $\delta \mathcal{R}_c$  is a rotation around  $z$  axis, therefore one can remove  $\delta \mathcal{R}_c$  from the eq.(6) multiplying both sides by  $z$ :

$$\mathcal{R}_{0,c} \mathcal{R}_{w,0} z = \hat{\mathcal{R}}_{w,c} \delta \mathcal{R}_c z = \hat{\mathcal{R}}_{w,c} z. \quad (7)$$

Given a set of  $F$  multicamera captures with the corresponding camera pose estimates, for each camera of the cluster we can stack  $F$  equations in the form of

(7). Relaxing the constraint (2) an estimate of the rotational component of the external calibration can be computed as the solution of the following optimization problem:

$$\mathcal{R}_{0,c} = \min_{\mathcal{R} \in SO(3)} \sum_{f \in [0, F-1]} \left\| \mathcal{R} \cdot \mathcal{R}_{w,(0,f)} z - \hat{\mathcal{R}}_{w,(c,f)} z \right\|^2. \quad (8)$$

This is a typical rotation fitting problem (8), which can be solved in closed form via spectral matrix analysis if at least two non degenerate vector correspondences are given (Kanatani, 1994). The rotational offset components of each view are then extracted from (6) and projected onto the discrete offset space (2):

$$\theta_{(c,f)} = \arg \min_{\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}} \left\| \mathcal{R}_Z(\theta) - \hat{\mathcal{R}}_{w,(c,f)}^T \mathcal{R}_{0,c} \mathcal{R}_{w,(0,f)} \right\|$$

$$\delta \mathcal{R}_{(c,f)} = \mathcal{R}_Z(\theta_{(c,f)}) \quad (9)$$

We remark that if a calibration pattern other than the checkerboard is used a different search space should be considered in (9).

Once the rotational offsets of each image are computed, the corresponding grid coordinates are corrected and the problem is reduced to the case of a pure translational offset recovery, which is addressed in the next section.

### 3.2 Translational Offset

Rewriting the translational component of the  $c$ -th camera pose using the transfer via the reference camera (5) together with eq.(3) one obtains:

$$\mathbf{T}_{w,c} = \mathcal{R}_{0,c}^T (\mathbf{T}_{w,0} - \mathbf{T}_{0,c}) = \mathcal{R}_{w,c} \delta \mathbf{M}_c + \hat{\mathbf{T}}_{w,c}, \quad (10)$$

where  $\delta \mathbf{M}_c$  is a translational offset in the  $c$ -th camera view. Notice that in (10) we considered a null rotational component of the offset, as at this stage this is assumed to be already removed. Equation (10) can be rearranged in the matrix form

$$\mathcal{A}_c [\mathbf{T}_{0,c}^T | \delta \mathbf{M}_c^T]^T = \mathbf{b}_c, \quad (11)$$

where

$$\begin{cases} \mathcal{A}_c &= [\mathcal{R}_{0,c}^T | \mathcal{R}_{w,c}] \\ \mathbf{b}_c &= \mathcal{R}_{0,c}^T \mathbf{T}_{w,0} - \hat{\mathbf{T}}_{w,c} \end{cases} \quad (12)$$

Similarly to the rotational offset,  $F$  equations of form (11) obtained from  $F$  multicamera captures can be stacked for each camera in the cluster resulting in a over determined non-homogeneous system  $\mathcal{A} \mathbf{x} = \mathbf{b}$ , with  $(3 + 2F)$  unknown parameters in the vector  $\mathbf{x} = [\mathbf{T}_{0,c}^T | \delta \mathbf{M}_{(c,0)}^T | \dots | \delta \mathbf{M}_{(c,F-1)}^T]^T$ . It can be solved in a least square sense, provided that  $F \geq 3$ . The unconstrained solution of the translational offsets is projected onto discrete space (2) by means of a simple

iterative procedure. In each iteration the system is solved and  $\mathbf{T}_{0,c}^T$  is dropped from the solution vector  $x$ . The parameter of the remaining subvector with the smallest fractional part, denoted by  $x_j$ , is selected, truncated and stored in the final solution. Then the linear system is accordingly reduced by removing the corresponding column from the coefficient matrix  $\mathcal{A}$  and updating the vector  $\mathbf{b}$ :

$$\mathcal{A} \leftarrow \mathcal{A}_{(j)}$$

$$\mathbf{b} \leftarrow \mathbf{b} - x_j \cdot \mathbf{a}_j,$$

where  $\mathbf{a}_j$  is the  $j$ -th column of  $\mathcal{A}$  and the notation  $\mathcal{A}_{(j)}$  means the removal of the  $j$ -th column from the matrix  $\mathcal{A}$ . The process iterates until the size of the system matrix  $\mathcal{A}$  is reduced to  $3 \times 3$ , namely until all the translational offset components are estimated and the residual unknown vector is  $\mathbf{x} = \mathbf{T}_{0,c}$ .

## 4 RESULTS

The preliminary evaluation has been performed on the simulated data in order to verify the applicability of the pattern recovery algorithm to a typical calibration scenario. An acquisition system has been modeled as a trifocal cluster, with each camera represented as a 1024x768 pixel sensor and a 50° horizontal field of view lens. Each camera is randomly located on a sphere of the radius  $\rho$ , with the optical axis pointing towards sphere center, where also the calibration object is placed. The latter is captured  $F$  times by all cameras under different orientations, producing a calibration dataset of  $F \times 3$  images. The calibration dataset is then twofold corrupted. First the 3D coordinates of the calibration object corresponding to each view are corrupted by an offset randomly sampled from the feasible domain (2). Then the pose of each view is altered by a random Euclidean transformation of increasing intensity, denoted as  $\delta \hat{\mathcal{G}}_n = (\mathcal{R}_a(\theta_n), \mathbf{T}_n)$ , where  $a$  is a random unit norm 3D vector,  $\theta_n$  is an angle ranging in the interval  $[0; 10^\circ]$  and  $\mathbf{T}_n$  is a random 3D vector with norm in the interval  $[0; 0.1\rho]$ . These two uncorrelated noise sources are introduced in order to simulate the nature of a real calibration dataset: the transformation of the 3D calibration object represents the offset (2) affecting the 3D coordinates of the pattern recovered from each view, the camera pose transformation simulates the effect of the noisy image measurements on the internal calibration and consequently on the camera pose estimation. We remark that the intensity of this second transformation is consistently higher than the expected error of the camera pose estimation in a

real calibration scenario. Essentially this means that we are testing the worst-case scenario.

For each level of noise intensity a new scene is generated 100 times and the number of unsuccessful trials is counted. A single trial is assumed unsuccessful if any of the offset parameters is incorrectly estimated. In Fig. 2 the performance measure  $\rho_e$ , given by the ratio of unsuccessful trials over the total number of trials, is plotted against the noise intensity level. The results obtained from simulated data confirm that the proposed technique provides a reliable mean for the automatic grid recovery in a multicamera calibration tool. Indeed, in a real scenario a number of  $F \geq 10$  captures are typically collected and the error affecting the camera pose estimates is typically within the range of a fraction of degree. From the plot shown in Fig. 2 we can observe that the expected error ratio in this working conditions is essentially zero and the grid recovery can be considered fully reliable.

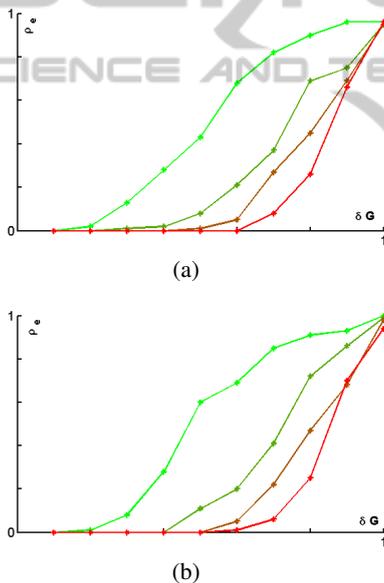


Figure 2: Results on the simulated data. The colors light green, dark green, brown and red identify the error ratio obtained for different size of the image set  $F = \{5, 10, 15, 20\}$ . The camera pose noise  $\delta G_n$  is expressed in normalized coordinates with respect to the intensity of the noise variables  $T_n$  and  $\theta_n$ .

We further tested the proposed algorithm as a part of our stereo calibration tool, which is working with two types of calibration objects: the standard checkerboard and a color-coded pattern embedding two bundles of orthogonal lines. The calibration pipeline is invoked according to the chosen pattern. The first one performs the grid extraction, then the internal calibration, based on the algorithm (Mei and Rives, 2007), the multiview grid recovery described in sections 3.1

and 3.2, and the external calibration, based on a multiview extension of (Lepetit et al., 2009). The second calibration pipeline performs the line features extraction, then the internal calibration, based on the algorithm (Kanatani, 2009), the multiview grid recovery and finally the external calibration, based on our unpublished algorithm.

For each calibration session, 10 stereo frames are captured using a GOPRO 3D HERO System, a stereo camera comprised of two GOPRO HERO 2 cameras equipped with a fisheye lens, rigidly coupled within a common chassis. This camera system turns out to be particularly hard to calibrate, mostly due to the difficulty in capturing a robust calibration dataset.

The calibration pattern is detected independently in each image by means of an automatic detector, that arbitrarily locates the origin of the axis orientation, without any multiview consistency control. This initial set of correspondences is used to compute the internal calibration of each camera. In the third calibration stage the grid offset is removed from each view. As it was pointed out in section 3, one can assume, that the grid extracted from the images of the reference camera is not affected by an offset, therefore the algorithm should only correct the offset in the images of *secondary* cameras of the multicamera system. In our test the left camera of the stereo rig is taken as a cluster reference and the grid of the right images undergoes the offset correction. The results obtained from this calibration stage are demonstrated in Fig. 3 and 4, presenting three stereopair samples from each calibration dataset. The detected grid is shown also in overlay for the left image and for the right image (before and after the grid offset removal).

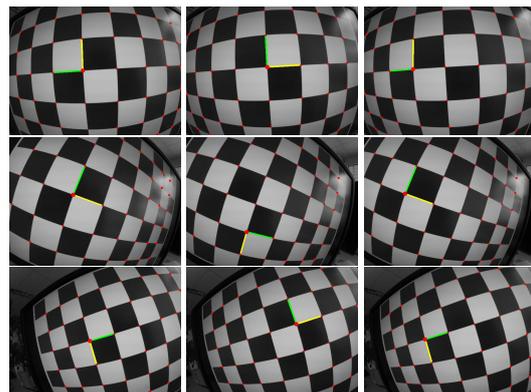


Figure 3: A sample of 3 stereopairs extracted from the  $2 \times 10$  image set processed according to the first calibration pipeline, (grid object and metric calibration). Each row shows the data corresponding to a stereopair. From left to right are shown the corners and the grid axes extracted from the left image, from the right image and the corrected grid of right image.

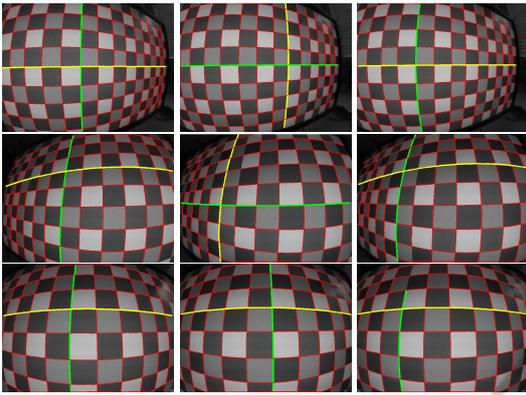


Figure 4: A sample of 3 stereopairs extracted from the 2x10 image set processed according to the second calibration pipeline, (line bundles object and non-metric calibration). Each row shows the data corresponding to a stereopair. From left to right are shown the grid axis extracted from the left image, from the right image and the corrected grid of right image.

The obtained results show that a periodic pattern can be successfully detected across multiple views only enforcing the rigidity constraint of the camera system and the discrete nature of the grid detection indeterminacy.

## 5 CONCLUSION

In this paper we derived a new technique, that simplifies the extraction of a symmetric 2D grid pattern across calibration pattern images obtained using multicamera systems, under the condition of partial visibility of the calibration object without the use of additional markers identifying the grid origin and the axis orientation.

The algorithm effectiveness has been proven on the simulated multicamera and real stereo camera datasets. We believe that this work provides a useful solution for a problem that inevitably arises in the context of multicamera calibration and allows for the design of a user-friendly camera calibration interface.

## REFERENCES

Agisoft Lens (2014). Agisoft lens automatic lens calibration software. <http://www.agisoft.ru/products/lens/>. [21 August 2014].

Atheson, B., Heide, F., and Heidrich, W. (2010). Caltag: High precision fiducial markers for camera calibration. In *Proc. VMV*, pages 41–48.

Bouguet, J. (2013). Camera calibration toolbox for matlab.

[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/). [21 August 2014].

CAMcal (2014). Camcal camera calibration program. <http://people.scs.carleton.ca/cshu/Research/Projects/CAMcal/>. [8 September 2014].

Datta, A., Kim, J., and Kanade, T. (2009). Accurate camera calibration using iterative resection of control points. In *ICCV Workshop on Visual Surveillance (VS)*.

Douskos, V., Grammatikopoulos, L., Kalisperakis, I., Karras, G., and Petsa, E. (2009). Faucal: An open source toolbox for fully automatic camera calibration. In *XXII CIPA Symposium*.

Faugeras, O. (1993). *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, ISBN: 9780262061582.

Fiala, M. and Shu, C. (2008). Self-identifying patterns for plane-based camera calibration. 19(4):209–216.

Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

Heikkila, J. (2000). Geometric camera calibration using circular control points. 22(10):1066–1077.

Kanatani, K. (1994). Analysis of 3-d rotation fitting. *IEEE T-PAMI*, 16(5):543–549.

Kanatani, K. (2009). Calibration of ultrawide fisheye lens cameras by eigenvalue minimization. *IEEE T-PAMI*, 35(4):813–822.

Kassir, A. and Peynot, T. (2010). Reliable automatic camera-laser calibration. In *Proc. of ACRA*.

Lepetit, V., F. Moreno-Noguer, and P. Fua (2009). Epnnp: An accurate o(n) solution to the pnp problem. *IJCV*, 81(2).

Mei, C. and Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids. In *Proc. IEEE ICRA*, pages 3945–3950.

OpenCV (2014). Open source computer vision library. <http://opencv.org/>. [8 September 2014].

Oyamada, Y., Fallavollita, P., and Navab, N. (2012). Single camera calibration using partially visible calibration objects based on random dots marker tracking algorithm.

Shu, C., Brunton, A., and Fiala, M. (2003). Automatic grid finding in calibration patterns using delaunay triangulation. Tech. Rep.

Tsai, Y. R. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. CVPR*.

Vaish, V. (2006). The stanford calibration grid detector. <http://graphics.stanford.edu/software/findgrid/>. [21 August 2014].

Vo, M., Wang, Z., Luu, L., and Ma, J. (2011). Advanced geometric camera calibration for machine vision. *Optical Engineering*, 50(11).

Wedekind, J., Penders, J., Howarth, M., Lockwood, A. J., and Sasada, K. (2013). Using generic image processing operations to detect a calibration grid. Tech. Rep.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhang, Z. (2002). Camera calibration with one-dimensional objects. volume 4, pages 161–174.