Semi-automatic Camera Calibration Using Coplanar Control Points

Rune H. Bakken, Bjørn G. Eilertsen, Gustavo U. Matus, Jan H. Nilsen
Sør-Trøndelag University College

Abstract

Camera calibration is a prerequisite for computer vision and photogrammetry applications that infer knowledge about an object's shape and location from images. Many implementations of camera calibration methods are freely available on the Internet. We have reviewed eleven existing camera calibration tools within a feature oriented framework for comparison, and our findings show that achieving the required accuracy can be quite cumbersome. Automatic detection of control points and live capture of calibration images are two key features that would simplify the process. We have developed a new camera calibration tool, with automatic control point extraction and live capture. Our semi-automatic application achieves the same level of accuracy as one of the most widely used camera calibration tools in computer vision applications.

1 Introduction

In many computer vision and photogrammetry applications it is necessary to infer knowledge about an object's shape and location from image evidence. The process of capturing an image is a mapping from a three-dimensional space to a two-dimensional plane, so some information is lost. In order to recover that lost information intimate knowledge about the mapping is needed. The mapping is represented by a model of the camera, and the process of estimating the parameters of the chosen camera model is called camera calibration.

It is common to divide the parameters into two groups, intrinsic and extrinsic. The intrinsic parameters include focal length and principal point, while the extrinsic parameters describe the camera's orientation and location in the world coordinate system. Both groups of parameters can be estimated through linear transformations. Additionally, there are non-linear distortions in the lens and from the camera manufacturing process that must be estimated to have a complete model of the camera.

Extensive research have been done [6, 11] to find accurate and robust methods for camera calibration. One class of techniques use known control points on a 2D or 3D calibration object as input for the estimation. The term self-calibration is used for approaches that do not rely on known control points.

Many camera calibration methods have implementations freely available on the Internet. While some progress has been made in terms of ease-of-use and automation,

---

This paper was presented at the NIK-2009 conference; see http://www.nik.no/.

camera calibration is still a cumbersome process using existing implementations. Many solutions require that the control points have been extracted in advance, either a separate application for automatic extraction is needed or the points must be manually identified by the user. Implementations with automatic control point extraction suffer from the problem that the algorithms fail to find points in some image configurations. This may result in poor accuracy because of a lack of input points, and the user has to go back and capture new calibration images to compensate. The solution to these problems would be to have a semi-automatic camera calibration application that given a known calibration object can capture images, detect control points and give the user immediate feedback on the quality of the images.

In this paper we present Calvin, a new camera calibration tool with automatic control point extraction and live capture of calibration images. Calvin is based on an existing calibration method using coplanar control points. We have compared Calvin with one of the most widely used calibration tools in computer vision applications, and while they yield similar accuracy, Calvin can produce results semi-automatically.

The paper is structured as follows: section 2 explores the theoretical background of camera calibration and discusses different types of calibrations, as well as proposed methods for calibrating cameras. In section 3 we discuss existing implementations that are freely available and compare them. Our new calibration implementation is presented in section 4 and in section 5 we outline some experiments that we have conducted with our implementation. In section 6 we sum up the paper by discussing the results of our experiments and point out some aspects of our implementation that needs further work.

2 Background

As mentioned in section 1 camera calibration is the process of estimating the parameters of a camera model. We describe in more detail some camera models that are used and discuss some proposed methods for calibrating cameras.

Camera Models

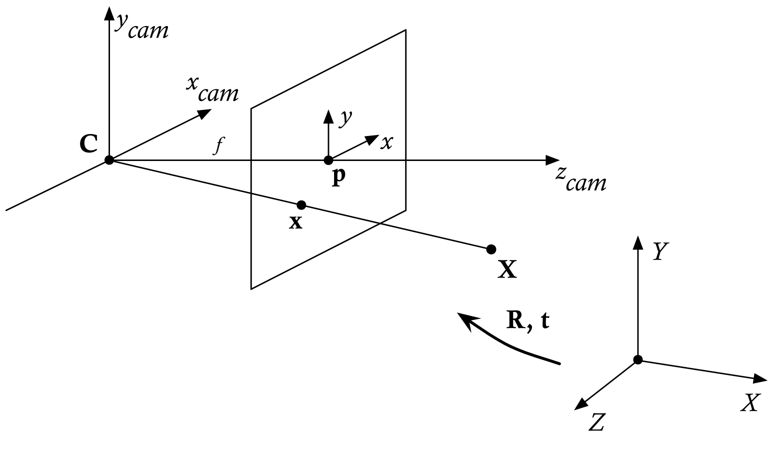
We start by briefly describing the camera models commonly in use. A more thorough explanation can be found in Hartley and Zisserman [7].

A camera is a mapping from a three-dimensional space onto a two-dimensional image plane. Perspective, or central, projection is the process whereby a point in Euclidean 3-space is mapped to the intersection of a line from the point to a central focal point, the camera centre, with the image plane. The most basic model of perspective projection is the pinhole camera model. The camera centre **C** is placed at the origin, with the principal axis of the camera pointing along the z axis. The point **p** where the principal axis intersects the image plane is called the principal point.

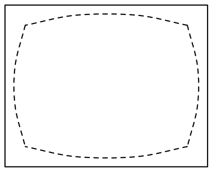
Given the orthogonal distance f, the focal length, from the camera centre to the image plane, the point $(X,Y,Z)^T$ is mapped to the point $(fX/Z, fY/Z)^T$. This mapping can be written as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \mapsto \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix}. \tag{1}$$

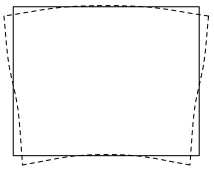
In this mapping it is assumed that the origin of the image plane coincides with the principal point. In reality this may not be the case, and we therefore add an offset to



(a) The linear mapping from world coordinates to the image plane of a camera.



(b) Radial distortion.



(c) Tangential distortion.

Figure 1: Camera models.

the mapped image coordinates. Using homogeneous coordinates we can express the projection in matrix form as

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2}$$

Writing

$$\mathtt{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{3}$$

we get the compact form

$$\mathbf{x} = \mathtt{K}[\mathtt{I}|\mathbf{0}]\mathbf{X}_{cam}. \tag{4}$$

The matrix $\mathtt{K}$ is called the camera calibration matrix, and its non-zero elements are called the intrinsic camera orientation.

The object points in equation 4 are given in the camera coordinate frame with the camera centre at the origin. In general it is desirable to have both the camera and the object points placed in some world coordinate system. Given a rotation matrix $\mathtt{R}$ and the camera centre coordinates in the world frame $\tilde{\mathbf{C}}$ we can express the transformation from the camera frame to the world frame as

$$\mathbf{X}_{cam} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X} \tag{5}$$

We can replace the camera centre coordinates with a translation vector $\mathbf{t} = -\mathtt{R}\tilde{\mathbf{C}}$, and combining equations 4 and 5 we get

$$\mathbf{x} = \mathtt{K}[\mathtt{R}|\mathbf{t}]\mathbf{X} \tag{6}$$

The rotation matrix $\mathtt{R}$ and the translation vector $\mathbf{t}$ are called the extrinsic camera orientation. The linear mapping from the world coordinate frame to image coordinates is shown in figure 1a.

In modern CCD cameras the pixels in the image sensor are usually not exactly square. This can be modelled in terms of pixels per unit distance in image coordinates, m_x and m_y, as

$$\mathtt{K} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{7}$$

where $\alpha_x = fm_x$, $\alpha_y = fm_y$, $x_0 = m_x p_x$ and $y_0 = m_y p_y$.

So far we have considered a linear mapping from object points to image points. This is a simplification that is sufficient for some applications, however, this mapping is in reality non-linear. This is called distortion and there are several ways to model it. The most commonly used model divides distortion into radial and tangential components.

When the focal length gets small, the light rays passing through the lens are subjected to a radial distortion. The distortion is 0 at the (optical) image centre and increases toward the edges of the image. The effect on the image coordinates is shown in figure 1b. Undistorted image coordinates $(x_u, y_u)^T$ can be found with a Taylor series expansion, given the distorted coordinates $(x_d, y_d)^T$, shown here with three terms

$$x_u = x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \qquad y_u = y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{8}$$

If the image sensor in the camera is not exactly parallel with the lens, there will be a tangential distortion of the image, as shown in figure 1c. Tangential distortion can be corrected with

$$x_u = x_d + (2p_1 y_d + p_2(r^2 + 2x_d^2)), \qquad y_u = y_d + (p_1(r^2 + 2y_d^2) + 2p_2 x_d) \tag{9}$$

Equations 8 and 9 are derived from Brown's models of distortion [3]. Two terms is usually sufficient for modelling radial distortion. For highly distorted fish-eye lenses the third term can be used. Tangential distortion is rarely a problem with modern CCD cameras and can usually be omitted from the camera model.

Camera Calibration Methods

The computer vision and photogrammetry communities have come up with a wide range of different methods for performing this estimation. In computer vision many approaches rely on using a calibration object with known control points. Both planar and special-purpose 3D objects have been used.

Tsai [17] presented a two-stage approach relying on $n > 8$ known feature points in each image. Some parameters are assumed to be constant and given by the manufacturer of the cameras. Tsai argues that tangential distortion is not required for machine vision applications and including it would cause numerical instability, hence only radial distortion is considered, with one term. The first step of the approach is to compute the extrinsic orientation using given camera information. In step two, the intrinsic orientation is refined using the results from step one. The method can be used both with planar grids of control points and 3D calibration objects.

The method presented by Heikkilä and Silvén [8] is a four-step procedure. Firstly, a linear estimation of the camera parameters is performed, disregarding distortions. Secondly, a non-linear optimisation scheme is used to refine the initial estimates and compute radial and tangential distortions. Step three corrects assymetric projections of shapes that cover more than one pixel. Finally, in step four, the undistorted image coordinates are computed. The method is demostrated on a 3D calibration object with circular control points, but can also be used with planar grids.

Zhang's [18] method is based on viewing a planar grid of control points from at least two different orientations. An initial estimate of the camera parameters is obtained with a closed-form solution. Next, maximum likelihood estimation is used to refine the initial solution. Radial distortion is modelled with two terms. Finally, non-linear optimisation is used to refine all parameters. This approach is quite similar to Trigg's [16].

Sturm and Maybank [13] presented a method that can handle varying intrinsic parameters, but they disregard any lens distortions. Thus a set of linear equations can be solved based on one or more views of a planar calibration grid. The authors also identify a number of singularities where plane-based approaches in general will yield unreliable results.

A simple, linear approach was presented by Bakstein [1]. A combination of real calibration images and a virtual 3D calibration object was used to improve the initial estimates of the calibration parameters. Personnaz and Sturm's [10] approach calibrated a stereo vision system, based on the motion of special-purpose 3D calibration object. A non-linear optimisation scheme is used to estimate the calibration parameters. Strobl and Hirzinger [12] presented a calibration method targeted for hand-eye calibration in robotics, but the algorithms can also be used for general camera calibration. Their approach include a parameterisation of the calibration pattern that compensates for inaccuracies in its measurement.

A comparison of four different linear algorithms for coplanar camera calibration was presented by Chatterjee and Roychowdhury [5]. They also presented a novel non-linear method specifically tailored for calibration with coplanar control points, using constrained optimisation. Results were compared with a photogrammetric calibration method, and the constrained non-linear algorithm was found to be on par with the photogrammetric method in terms of accuracy. Sun and Cooperstock [14] presented another comparison of methods, notably those of Tsai and Zhang, and studied the difference in accuracy of a casual versus an elaborate calibration setup.

A separate class of methods termed self-calibration in the computer vision literature does not rely on any known calibration object, but rather apply a number of constraints to infer calibration parameters from an unknown scene. Three types of constraints are used: scene constraints, camera motion constraints and constraints on the intrinsic orientation of the camera. A review of self-calibration methods is given by Hemayed [9].

Svoboda et al. [15] presented a self-calibration method designed for large camera arrays. They used a laser pointer to produce easily detected corresponding points in all images. Geometric constraints were imposed on the recovered control points. Their algorithm first produced a linear estimate of calibration parameters, and this was used as input for a post-processing step that determined non-linear distortion parameters.

Close-range photogrammetry is a field of research that has matured for over 150 years and accurate camera calibration has been an important aspect of photogrammetric applications for much of that period. Clarke and Fryer [6] give an overview of the main contributions from the last fifty years. The predominant method for photogrammetric camera calibration is the bundle adjustment approach [4]. This technique produces a simultaneous determination of all intrinsic and extrinsic parameters. Bundle adjustment can be used both with known control points and for self-calibration.

Remondino and Fraser [11] presented a comparison of different calibration methods for digital cameras, both from the computer vision and photogrammetric points of view. They compared experimental results of six implementations, with bundle adjustment methods yielding superior results.

3 Existing Implementations

Several implementations of the calibration methods presented in section 2 are freely available on the Internet. We are interested in determining how easily eleven freely available calibration tools can produce results for a typical computer vision camera calibration task, and we have compared their feature sets with this in mind. For our review of existing calibration applications we have formulated a feature oriented framework of comparison, consisting of the following categories: *calibration method*, *calibration pattern*, *automatic pattern detection*, *rectification of distorted images*, *extrinsic calibration*, *minimum number of cameras*, *live capture*, *platform requirements*, and *latest update*.

TsaiCode[1] This is the reference implementation of Tsai's [17] calibration method described in section 2. It is the oldest of the implementations in this comparison, the latest version being over a decade old. It has only a rudimentary command line interface and it requires that the control points have been extracted in advance.

Microsoft Easy Camera Calibration Tool[2] The reference implementation of Zhang's [18] method is also quite dated. As with TsaiCode it requires the coplanar control points to be extracted in advance.

Matlab Camera Calibration Toolbox[3] This toolbox is perhaps the most widely used of the freely available calibration solutions. The implementation is inspired by Zhang's [18] method, but uses Heikkilä and Silvén's [8] model of distortion. The toolbox has many useful features, such as the ability to use an earlier calibration result as the initial values for a new run of the algorithm, advanced error analysis and the possibility to undistort images, however, it lacks automatic extraction of control points from the images.

Camera calibration toolbox for Matlab (Heikkilä)[4] The reference implementation of Heikkilä and Silvén's [8] method is also a Matlab toolbox, however, it lacks the GUI features of the previously described method. Image coordinates of the control points must be extracted in advance.

tclcalib[5] This implementation is a collection of small stand-alone programs that together makes Heikkilä's easier to use. This includes automatic extraction of control

---

[1] http://www.cs.cmu.edu/~rgw/TsaiCode.html (last checked 09-07-2009)

[2] http://research.microsoft.com/en-us/um/people/zhang/calib/ (last checked 09-07-2009)

[3] http://www.vision.caltech.edu/bouguetj/calib_doc/ (last checked 09-07-2009)

[4] http://www.ee.oulu.fi/~jth/calibr/ (last checked 09-07-2009)

[5] http://users.soe.ucsc.edu/~davis/projects/tclcalib/ (last checked 09-07-2009)

points, extrinsic calibration given an existing intrinsic calibration and a gui tie it all together.

Camera calibration toolbox for Matlab (Bakstein)[6] This Matlab toolbox implements Bakstein's [1] calibration method. The tool is quite simple, control points must be extracted in advance.

BlueCCal[7] This is a Matlab toolbox implementing Svoboda et al.'s [15] method. It does not have a GUI, but has automatic control point extraction. One drawback is that it requires at least three cameras for the calibration to work. It also does not place the origin of the world coordinate system in a known location, but rather at the centre of the extracted point cloud.

GML calibration tools[8] This calibration tool is a stand-alone application that mimics the functionality of Bouguet's Matlab toolbox. The calibration routines used are from the OpenCV Library [2], which is a reimplementation of Bouget's Matlab code. GML sports automatic control point extraction and correction of distorted images. The program requires a .Net runtime.

Camera Calibration Tools[9] This is another stand-alone application inspired by Bouguet's Matlab toolbox, that relies on OpenCV's [2] calibration routines. It has relatively limited functionality, as it only supports intrinsic calibration, but it has one interesting feature: the ability to capture images live within the program.

Tele2[10] This stand-alone application implements Personnaz and Sturm's [10] method. Tele2 requires a special-purpose calibration object that complicates its use. It is designed for stereo calibration, but also works with single cameras. One advantage of this application is that it is implemented in Java, hence it is platform independent.

CalDe and CalLab[11] These two applications implement Strobl and Hirzinger's [12] calibration method. This implementation supports calibration of a single or a stereo pair of cameras, offers several estimation methods and has advanced features for analysing the calibration results. The program requires an IDL runtime.

The findings of our review are summed up in table 1. As can be seen some of the offerings have not been updated for quite some time, making them cumbersome to use with current versions of their respective platforms. About half of the impementations require Matlab, an expensive software package for scientific computation. While this might not necessarily be a problem, it can be an issue if funds are limited. Some of the newer tools offer automatic extraction of control points. This saves a lot of time in generating input for the calibration algorithm, however, it must be robust to have a meaningful impact on the time used. The point detection algorithms are unable to find the correct pattern in some image configurations. With fewer control points the accuracy of the calibration results are potentially reduced, and the user is forced to go back and aquire more images to compensate.

---

[6] http://terezka.ufa.cas.cz/hynek/toolbox.html (last checked 09-07-2009)

[7] http://cmp.felk.cvut.cz/ svoboda/SelfCal/index.html (last checked 09-07-2009)

[8] http://research.graphicon.ru/calibration/2.html (last checked 09-07-2009)

[9] http://www.doc.ic.ac.uk/ dvs/calib/main.html (last checked 09-07-2009)

[10] http://perception.inrialpes.fr/Soft/calibration/index.html (last checked 09-07-2009)

[11] http://www.dlr.de/rm/desktopdefault.aspx/tabid-4853/ (last checked 09-07-2009)

	Calibration method	Calibration pattern	Automatic pattern detection	Undistort images	Extrinsic calibration	Minimum number of cameras	Live capture	Platform requirements	Last updated
TsaiCode	Tsai [17]	Known control points (2D or 3D)	No	No	Yes	1	No	Unix/Dos	28-10-1995
Microsoft Easy Camera Calibration Tool	Zhang [18]	Coplanar control points	No	No	Yes	1	No	Windows	04-06-2001
Matlab Camera Calibration Toolbox (Bouguet)	Zhang [18]	Checkerboard	No	Yes	Yes	1	No	Matlab	02-06-2008
Camera calibration toolbox for Matlab (Heikkilä)	Heikkilä and Silvén [8]	Grid of circular control points	No	No	Yes	1	No	Matlab	17-10-2000
tclcalib	Heikkilä and Silvén [8]	Grid of circular control points	Yes	No	Yes	1	No	Irix/Windows	14-08-2002
Camera calibration toolbox for Matlab (Bakstein)	Bakstein [1]	Line grid	No	No	Yes	1	No	Matlab	10-06-1999
BlueCCal	Svoboda et al. [15]	Laser pointer	Yes	No	Yes, but not relative to a known coordinate frame	3	No	Matlab	24-05-2005
GML C++ Camera Calibration Toolbox	Zhang [18]	Checkerboard	Yes	Yes	No	1	No	.Net 1.1	06-02-2006
Camera Calibration Tools	Zhang [18]	Checkerboard	No	Yes	No	1	Yes	Windows	16-02-2007
Tele2	Personnaz and Sturm [10]	Special-purpose calibration object	Yes	No	Stereo	1	Yes	Java	20-03-2002
CalDe and CalLab	Strobl and Hirzinger [12]	Checkerboard with special pattern	Yes	No	Yes	1	No	IDL	30-01-2008

Table 1: Comparison chart of freely available camera calibration tools.

Live capture combined with automatic control point extraction gives the user immediate feedback on the quality of a given calibration image, but only two of the studied solutions have this feature. *Tele2* requires a special-purpose 3D calibration object which makes this method difficult to use, while the *Camera Calibration Tools* has very limited features other than live capture.

4 Calvin – A Semi-automatic Camera Calibration Tool Using Coplanar Control Points

As was mentioned in section 3, there are a lot of existing implementations with many useful features freely available, but none that include all the features necessary for semi-automatic camera calibration. By semi-automatic calibration we mean an application that can capture a set of images of a calibration object and estimate the camera's intrinsic, extrinsic and distortion parameters without any further manual processing. We have developed Calvin, a new stand-alone camera calibration application that includes all the features in table 1, and is capable of semi-automatic calibration with a planar calibration object.

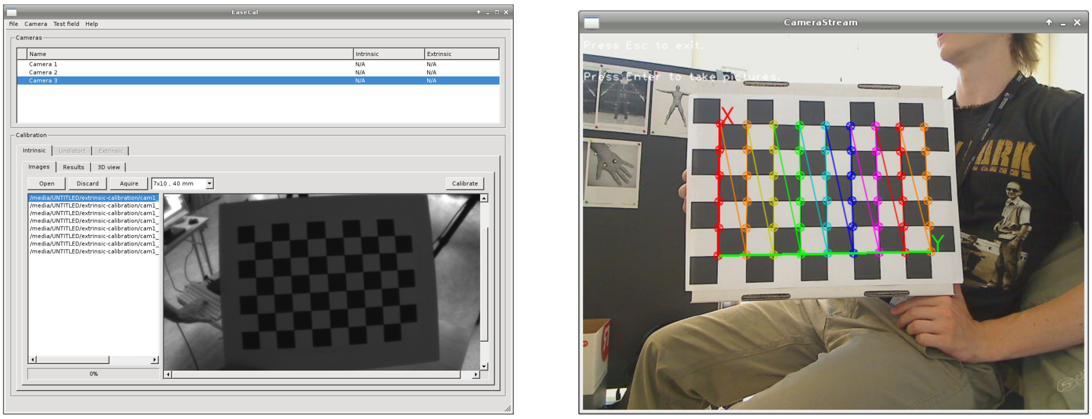
Calvin is built around the camera calibration routines in OpenCV [2]. Calibration in OpenCV is based on a combination of Zhang's method [18] and the distortion terms from Brown [3]. The Qt library is used to create a simple and user-friendly GUI. Both libraries are open-source and platform independent, so Calvin can be built for a multitude of operating systems.

Intrinsic and extrinsic calibration is done in two separate steps. The main window of the application, shown in figure 2a, is organised in tabs to reflect this functional subdivision. A list of available checkerboard patterns can be maintained for use in different situations, for instance a smaller hand-held pattern for intrinsic calibration, and a larger pattern placed on the floor for extrinsic calibration.

For a given camera, the user can choose to load a set of calibration images from disk, or use live capture to aquire the images. When using live capture the automatically detected control points are highlighted, thus ensuring that the images are useable in the calibration steps. Currently, only ordinary USB cameras are supported for live capture, but support for other types of cameras will be added in the future. Figure 2b shows live capture of a checkerboard pattern with detected control points overlayed. Once an adequate set of images (ten or more) has been loaded or aquired the intrinsic calibration can be performed.

Given the intrinsic camera parameters, extrinsic calibration can be performed using a single image loaded from disk or captured live. Cameras are handled independently by the calibration routines, so there are no lower or upper limits on the number of cameras. Provided that images of the same calibration pattern can be aquired by a set of cameras, the extrinsic parameters for those cameras can be combined to place them in the same world coordinate frame. The placement of the cameras can be reviewed in a 3D view. Figures 2c and 2d shows a laboratory camera setup and the results of its calibration.

Additionally, Calvin can undistort severly distorted images. Provided that the intrinsic calibration has been done, images from for instance a camera with a fish-eye lens can be rectified.



Name	Intrinsic	Extrinsic
Camera 1	N/A	N/A
Camera 2	N/A	N/A
Camera 3	N/A	N/A

7x10 , 40 mm

/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_
/media/UNTITLED/extrinsic-calibration/cam1_

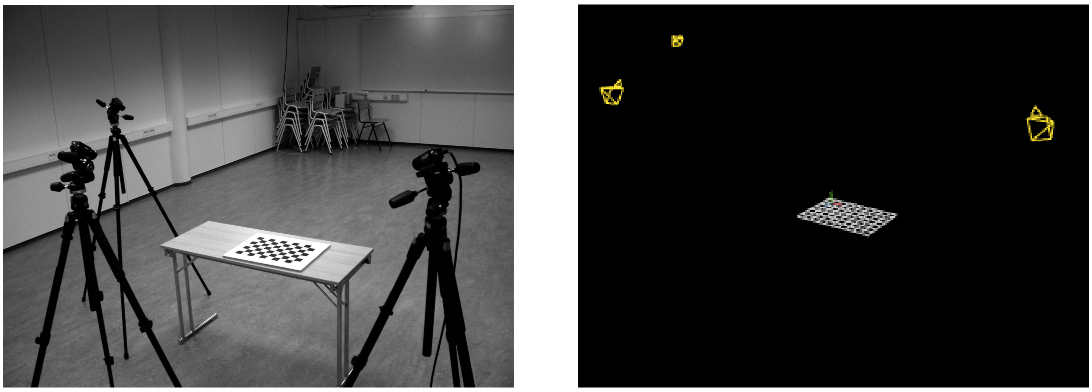
0%

Press Esc to exit.

Press Enter to take pictures.

(a) Main window with the list of calibration images.

(b) Live capture with automatic pattern detection.



(c) Laboratory environment with three cameras.

(d) 3D view of the calibrated cameras in the laboratory environment.

Figure 2: The main functionality of the Calvin calibration tool.

5 Experiments

We wanted to determine if Calvin's semi-automatic operation affects the accuracy of the calibration results. Since both Calvin and Bouguet's calibration toolbox are based on the same calibration method, we compared their respective output with the same input images. We captured ten image sequences, each consisting of 25 images. Each sequence contained images of the checkerboard with 90 degree rotations to avoid singularities. The camera used was an AVT Marlin with a 6.5 mm fixed focal length lens.

Bouguet's toolbox allows the checkerboard corners to be recomputed using the previous calibration result for use as initial values in a new run of the estimation algorithm. Hence, we tried single, and five consecutive runs to see if this greatly affects accuracy. The results can be seen in table 2, and they show little difference in accuracy between the three trials. The estimates are consistent, and the standard deviations indicate that they are reproducable. It should be noted that processing the 250 images using Calvin took about ten minutes, while several hours were required to do the manual corner extraction using Bouguet's toolbox.

	Calvin	Bouguet's toolbox (1 run)	Bouguet's toolbox (5 runs)
α_x	1424.32220 (2.73598)	1424.35075 (2.91961)	1425.20787 (2.04359)
α_y	1424.05504 (2.72395)	1424.08673 (2.94896)	1425.00161 (2.38663)
x_0	302.71328 (3.66743)	303.20818 (3.97749)	303.81051 (4.22332)
y_0	240.94768 (2.88067)	240.38488 (2.86339)	239.56113 (2.65305)
k_1	-0.24282 (0.01242)	-0.24323 (0.01308)	-0.24823 (0.01137)
k_2	0.39128 (0.25483)	0.40439 (0.26352)	0.51467 (0.26808)
p_1	0.00035 (0.00036)	0.00031 (0.00036)	0.00029 (0.00033)
p_2	-0.00025 (0.00056)	-0.00017 (0.00057)	-0.00021 (0.00054)

Table 2: Comparison of intrinsic calibration with Calvin and Bouguet's Matlab Calibration Toolbox. The results are based on ten image sets, each consisting of 25 images. Mean values for all intrinsic parameters are given, with standard deviations in parentheses.

6 Conclusion

We have presented a new semi-automatic camera calibration tool based on coplanar control points in the form of a checkerboard pattern. The stand-alone application, Calvin, is platform independent and automatically extracts control points from captured images. The results of our experiments show that Calvin's accuracy is on par with one of the most widely used calibration tools in computer vision, and should be usable in many of the same application areas.

Using live capture to aquire the calibration images improves Calvin's robustness, but extrinsic calibration with automatic detection of a checkerboard pattern can still yield poor results if the angle between the principal axis of the camera and the control point plane is too acute. We plan to complement the automatic pattern detection algorithm with the ability to manually select control points for extrinsic calibration.

Compared to some of the other calibration tools we have studied, the functionality for error analysis in Calvin is fairly rudimentary. We plan to add more advanced error analysis tools in the future.

References

[1] H. Bakstein. A complete DLT-based camera calibration with virtual 3D calibration object. *Diploma thesis*, 1999.

[2] G. Bradski and A. Kaehler. Learning OpenCV. Computer Vision with the OpenCV Library. *O'Reilly*, 2008.

[3] D. Brown. Close-range camera calibration. *Photogrammetric engineering*, 1971.

[4] D. Brown. The bundle adjustment—progress and prospects. *International Archives of Photogrammetry*, 1976.

[5] C. Chatterjee and V. Roychowdhury. Algorithms for coplanar camera calibration. *Machine Vision and Applications*, 2000.

[6] T. Clarke and J. Fryer. The development of camera calibration methods and models. *Photogrammetric Record*, 1998.

[7] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2003.

[8] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[9] E. Hemayed. A survey of camera self-calibration. *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003.

[10] M. Personnaz and P. Sturm. Calibration of a stereo-vision system by the non-linear optimization of the motion of a calibration object. *INRIA Technical Report*, 2002.

[11] F. Remondino and C. Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006.

[12] K. Strobl and G. Hirzinger. More accurate camera and hand-eye calibrations with unknown grid pattern dimensions. *IEEE International Conference on Robotics and Automation*, 2008.

[13] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[14] W. Sun and J. Cooperstock. Requirements for camera calibration: Must accuracy come with a high price? *Seventh IEEE Workshop on Application of Computer Vision*, 2005.

[15] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & Virtual Environments*, 2005.

[16] B. Triggs. Autocalibration from planar scenes. *Lecture Notes in Computer Science*, 1998.

[17] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 1987.

[18] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *IEEE International Conference on Computer Vision*, 1999.