Machine Vision and Applications (2008) 19:209–216
DOI 10.1007/s00138-007-0093-z

ORIGINAL PAPER

Self-identifying patterns for plane-based camera calibration

Mark Fiala · Chang Shu

Received: 18 July 2006 / Accepted: 27 March 2007 / Published online: 3 August 2007

Abstract Determining camera calibration parameters is a time-consuming task despite the availability of calibration algorithms and software. A set of correspondences between points on the calibration target and the camera image(s) must be found, usually a manual or manually guided process. Most calibration tools assume that the correspondences are already found. We present a system which allows a camera to be calibrated merely by passing it in front of a panel of self-identifying patterns. This calibration scheme uses an array of fiducial markers which are detected with a high degree of confidence, each detected marker provides one or four correspondence points. Experiments were performed calibrating several cameras in a short period of time with no manual intervention. This marker-based calibration system was compared to one using the OpenCV chessboard grid finder which also finds correspondences automatically. We show how our new marker-based system more robustly finds the calibration pattern and how it provides more accurate intrinsic camera parameters.

Keywords Calibration · Fiducial marker systems · ARTag

1 Introduction

The process of camera calibration determines the intrinsic and/or extrinsic parameters of the camera from correspondences between points in 3D and their projection points in one or more images [11,12]. People usually use easily identifiable objects, such as a chessboard, as calibration targets to establish these correspondences. Although it is possible to calibrate cameras from pure natural images, called *autocalibration* [6], when accuracy is required, calibration targets are usually employed.

It is more convenient to construct and use a planar target object than a 3D one. A typical planar target can be printed on a standard letter-sized page from a laser printer and mounted on a rigid surface. Flexible calibration algorithms [10,14] have been designed to calibrate a camera from correspondences on a planar object captured from a few different views.

While the problem of calibration has been thoroughly studied, the issue of correspondence has received little attention. The performance of the target detection systems, for example robustness and accuracy, has seldom been investigated. This is perhaps because calibration was traditionally an off-line, once-for-all process, and one could afford to create the correspondences manually to have a small number of cameras calibrated. However, as cameras are increasingly becoming a commodity, more and more applications make use of large numbers of cameras in often changing configurations. Examples of these applications are security surveillance, robot navigation, special effects in movie production, and augmented reality. These applications demand rapid calibration of cameras.

OpenCV [1], a popular open source library for computer vision, provides a function for automatically finding grids from chessboard patterns. However, it requires the user to provide the dimensions (number of rows and columns) of the chessboard. It attempts to order corners in a grid to find correspondences. One disadvantage of this method is its lack of robustness in that all points must be fitted into the grid

M. Fiala · C. Shu (✉)
Institute for Information Technology,
National Research Council Canada, 1200 Montreal Road,
Building M-50, Ottawa, ON, Canada K1A 0R6
e-mail: chang.shu@nrc-cnrc.gc.ca

M. Fiala
e-mail: mark.fiala@nrc-cnrc.gc.ca

for any correspondences to be reported, and so many image frames are unusable. It does not work well for cameras with highly distorted lenses.

Soh et al. [7,9] use a pattern of 16 squares which are found with thresholding and blob analysis. The centroids of the squares are arranged to fit a regular grid using attributed relational graph matching. The use of structural knowledge of the grid could potentially improve the robustness of the pattern detection algorithm, though little detail was given in the paper, nor is the software based on their method freely available.

Shu et al. [8] have a system called *CAMcal* which uses corner detectors and topological operators to find chessboard squares to order the corner points. It, however, can fail when false corners get detected or when the pattern is viewed from an oblique angle such that the triangulation step does not correctly link corners. CAMcal is quite sensitive to imperfections in the image. However, it does not require that an entire pattern is in view.

This paper introduces a way to use the recently developed *ARTag* fiducial marker system [3] as a more robust method to find the correspondences. An array of self-identifying markers is presented to the camera-under-test in various poses and the correspondences will be generated automatically.

The benefit of this system is that a camera can be calibrated in a matter of minutes since the correspondences are determined robustly and automatically. The camera is simply moved around to several views in front of a planar array of self-identifying markers and the calibration is performed completely automatically. The whole process for a given camera takes under 5 min. Typically about 10–15 images are needed from different angles to accurately calibrate the camera (Fig. 1).

The rest of the paper is organized as follows. Section 2 describes the ARTag patterns. Section 3 discusses the calibration system. Section 4 reports experimental results of the system running under different conditions and comparisons with the publicly available OpenCV grid finder and semi-manual calibration using conventional dot arrays. Finally, in Sect. 5, we draw conclusions and offer discussions.

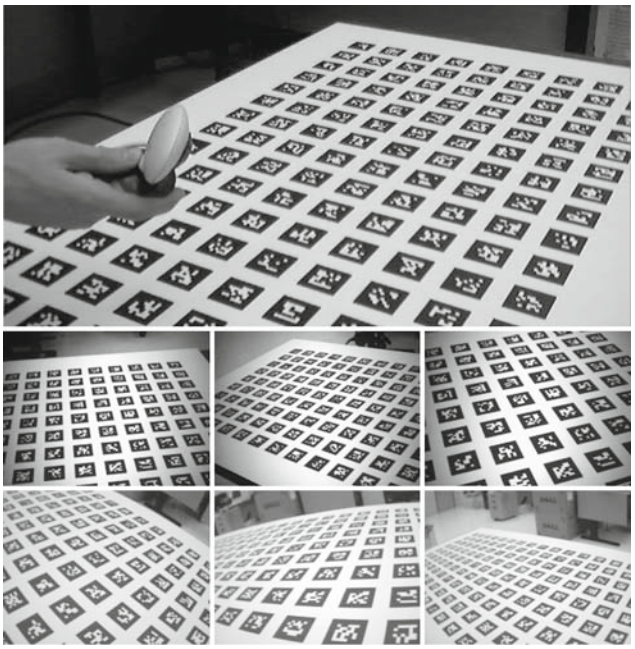


Fig. 1 (*Left*) Camera-under-test is moved around to different views of self-identifying pattern. Examples of captured images used for calibration (*right*). Typically 10–15 images are needed

2 Self-identifying patterns

Self-identifying patterns are special marker patterns that can be placed in the environment and automatically detected in the images. Also known as *fiducial marker systems*, a library of these patterns and the algorithms to detect them help to solve the correspondence problem. Self-identifying marker systems are typically used for applications such as calculating camera pose for augmented reality and robot navigation.

The ARTag system is detailed in [3]. Here, we give a brief description. The marker system is bi-tonal, containing 2,002 planar markers, each consisting of a square border and an interior region filled with a 6×6 grid of black or white cells. A total of 1,001 markers have a black square border on a white background, and vice versa for the other 1,001.

The algorithm for detecting the markers in an image first locates quadrilaterals which may be perspective views of the marker border, then the interior is sampled into 36 binary '1' or '0' symbols. Further processing is in the digital domain providing a non-linear response giving very low *false positive* and *inter-marker confusion* rates. With this marker system, the probability of falsely identifying one marker for another, or a piece of the background as a marker, is vanishingly low [2]. Figure 2 shows the markers being detected in an image.

The correspondences are the centers of each marker or the four corners of each marker located in the image (Fig. 2). Both the usage of the four corners, and just the marker center were investigated with the conclusion being that the best accuracy is obtained using the marker center.

Since each marker is unique, the 3D coordinates associated with the marker can be found once the marker is identified. Therefore, there is no need for ordering the markers.

While the points themselves are reliable as far as identification, the accuracy of their reported image locations will affect the accuracy of calibration. The location of each marker corner is determined by finding the intersection of the line of the two adjoining sides. The line equations themselves are found by fitting a line to points of locally maximum derivative

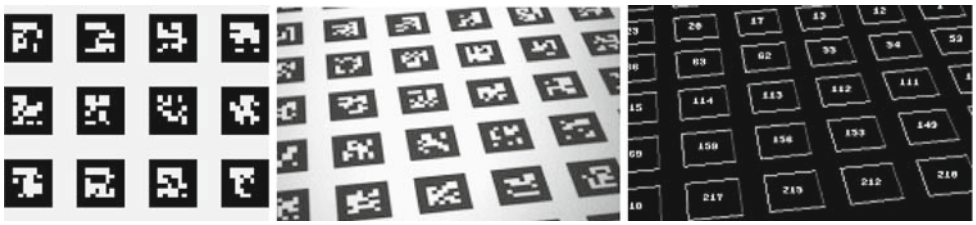


Fig. 2 Stages of marker detection. Markers (*left*) are searched in the input image (*middle*) and their corners or centers and the matching world coordinate point used as correspondences (*right*)

perpendicular to the line. This differentiates ARTag from the basic greyscale thresholding technique used in many calibration and fiducial systems. A measure of accuracy of the image location is the jitter, with full room lighting and good focus, an ARTag marker corner's coordinate has a standard deviation of about 0.03 pixels when a marker is viewed with a 640 × 480 greyscale Dragonfly IEEE-1394 camera (surprisingly this experiment yields the same result regardless of marker size).

3 Calibration

Camera calibration typically involves finding parameters for a camera model given world-point correspondences. Photogrammetry's *bundle adjustment* is highly accurate and has complex models with dozens of parameters. At the other end of the spectrum of complexity is the two parameter model of focal length and scale, or horizontal and vertical focal lengths (F_x, F_y). The horizontal F_x and the vertical F_y focal lengths in pixels can be simply calculated by measuring the image width and height of a known object at a known distance. This assumes a pinhole projection model with the center of projection (u_o, v_o) as being the image center, this is sufficient for many projects.

The next step in complexity is to calculate the entire camera matrix $\mathbf{K}$ containing F_x, F_y, u_o, v_o and possibly the skew factor s. Suppose $\mathbf{X} = [\,X\ Y\ Z\ 1\,]^\top$ is a 3D point and $\mathbf{x} = [\,x\ y\ 1\,]^\top$ is its projection in the image plane, both in homogeneous coordinate. The perspective projection can be modeled by

$$\lambda \mathbf{x} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\mathbf{X},$$

where λ is an arbitrary constant, $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector, together they relate the world coordinate frame to the camera coordinate frame, and $\mathbf{K}$ is an upper triangular matrix defined as

$$\mathbf{K} = \begin{bmatrix} F_x & s & u_0 \\ 0 & F_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Most cameras, especially low-cost ones with lenses of large curvature, require going beyond this pinhole model to address the "barrel" or "pincushion" effect of radial distortion. Radial and thin prism distortion is typically modeled by polynomials with 1 or 2 term coefficients. Zhang's model [13,14] finds these, which this paper labels as F_x, F_y, u_o, v_o for the focal lengths and image center, k_1, k_2 for the radial distortion, and the thin prism parameters p_1, p_2.

To calibrate a camera, one needs to have a list of corresponding image points and world points (either co-planar 2D points, or 3D points). Gathering these correspondences has usually been a manual task, which is slow and tedious and prone to errors that adversely affect the results. If the camera is moved or jarred, its focus or zoom changed, it must be re-calibrated. Ways to do it automatically would find much use.

Our system was created by combining the marker detection with OpenCV's *cvCalibrateCamera()* function, which implements Zhang's algorithm [14]. An array of 19 × 9 markers was printed and mounted on a table top size 60 in. × 30 in. meddite panel and was used for most of the experiments in this paper. Two smaller marker arrays patterns were printed on 8.5 in. × 11 in. pages for the experiments in Sect. 4.2.

Figure 1 shows a webcam being calibrated by capturing views of the meddite panel. Figure 3 shows the recovered camera positions relative to the panel.

The quality of each calibration is evaluated by computing the reprojection error, which is the Euclidean distance between the identified marker point and the projection of its corresponding pattern point onto the image.

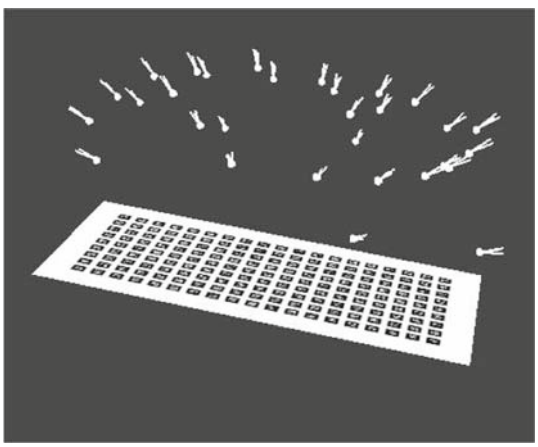


Fig. 3 Automatically recovered camera positions relative to the marker array

Table 1 Calibration using Zhang's algorithm with points extracted automatically using the pattern array with two methods: using the marker center as a correspondence point, or using all four corners

Sequence		**K** matrix				Distortion parameters				Re-projection error std. Dev/maximum	
# Frms	Avg. Pts/Frm	F_x	F_y	u_o	v_o	k_1	k_2	p_1	p_2	Center set	Corner points
Powershot S60 short focal length (640 × 480 pixels)											
63	48	531.70	531.46	322.45	243.19	−0.1448	0.1090	0.0005	0.0006	0.14/1.64	0.25/4.64
Point grey research dragonfly IEEE-1394 camera with 8 mm lens #0 (640 × 480 pixels)											
44	55	1097.63	1100.41	327.69	256.40	−0.0523	−0.0883	−0.0023	−0.0019	0.13/1.62	0.35/7.28
Creative webcam live ultra for laptops USB 2.0 Webcam (640 × 480 pixels)											
35	62	520.90	521.24	332.92	277.33	−0.2968	0.0903	−0.0006	−0.0020	0.16/1.97	0.25/6.66
Logitech quickcam Pro 4000 USB 2.0 webcam (640 × 480 pixels)											
25	37	786.26	789.44	373.26	271.22	0.1227	−0.0660	0.0028	0.0049	0.29/2.62	0.63/5.8
SONY 999 NTSC camera captured with ATI PCI framegrabber (640 × 480 pixels)											
40	59	633.87	620.20	304.13	240.31	−0.2034	0.1060	−0.0002	−0.0003	0.13/1.25	0.27/5.06
Single-board X10 greyscale NTSC camera captured with ATI PCI framegrabber (640 × 480 pixels)											
27	31	632.60	616.31	346.58	239.30	−0.4231	0.1764	−0.0002	0.0000	0.19/1.21	0.39/4.72

Calibration parameters are shown for the center method

A variety of cameras or camera/frame-grabber configurations were calibrated ranging from low-cost webcams with resolutions 320 × 240 pixels, NTSC cameras, to high resolution 1,280 × 1,024 digital cameras. Table 1 shows the results of automatic calibration of some of these cameras using the marker array.

4 Performance evaluation

4.1 Comparison with OpenCV grid finder

The OpenCV grid finder function *cvFindChessBoardCornerGuesses()* can be used to automatically find correspondences between the image points and the points in the calibration plane. The function starts with converting the input image into a greyscale image followed by thresholding to obtain a binary image. It then finds contours from the binary image and extracts all the contours with exactly four sides. The corners of those four-side contours are considered to be the corners of the chessboard. They are ordered into a grid based on their geometric proximity.

The function often does not recognize all the points as discussed earlier. The images must have low radial distortion, the grids must be completely visible, and the lighting has to be good for the thresholding step to succeed. In general, the detection can be unreliable and we found many images had to be captured to provide enough successfully detected grids could be found for calibration. Table 2 shows some sample results of how often grids are located, images are taken with five cameras of a 8 × 10 pattern.

This low detection rate can be compared to our system where correspondences are found in all images and there is no requirement to see the full pattern. With our marker system, every image taken with reasonable lighting, focus, and lack of motion blur produces at least some correspondences. The ARTag marker system has a very low false detection rate when compared to the many false corners detected by the OpenCV grid finder, especially if the scene around the pattern is cluttered. Also, markers can be detected to almost 85° from the normal whereas we could not get the OpenCV grid finder to find markers past 60° from the normal.

The main advantage of the ARTag pattern over the chessboard pattern is that each marker is unique, therefore its coordinates in the pattern plane can be recovered as soon as the marker is identified. For the chessboard, these coordinates have to be obtained through ordering the quadrilaterals based on their geometric proximity. In the OpenCV implementation, the ordering requires the entire grid to be visible. Although methods can be designed to detect and order partial grids in the chessboard [8], it is generally difficult to order the grids robustly, especially when the lens distortion is high.

Zhang's method requires multiple views with different perspectives and it is difficult in OpenCV to get many views where the points stretch well into the image corners, where the distortion is greatest and in most need of calibration. This limits the potential accuracy of the calibration. In contrast, our method allows the pattern to be larger than the field of view so that any view has pattern points seen throughout the image. Our method is ideal for providing the

Table 2 Comparison of calibration using marker corner, marker center, circular blobs, and chessboard methods

Camera	Marker corners			Marker centers			Circular blobs			Chessboards		
	% det.	avg. pts	Reproj. error	% det.	avg. pts	Reproj. error	% det.	avg. pts	Reproj. error	% det.	avg. pts	Reproj. error
Powershot S60	100	76	0.55/3.55	96	37	0.43/3.23	100	80	0.49/3.81	13	63	0.72/5.86
Dragonfly 8mm	100	79	0.18/2.00	100	68	0.32/2.43	100	80	0.27/1.99	6	69	46.1/477
HICOL Dragonfly	100	80	0.78/9.45	100	63	1.44/9.56	100	80	1.19/8.37	40	60	1.65/13.7
Creative Ultra	100	79	0.45/3.55	100	48	0.50/5.78	100	80	0.62/5.93	58	78	0.99/10.7
Intel Pro	100	78	0.46/3.54	100	45	0.51/3.92	100	80	0.47/3.65	14	60	20.2/301

An array was made for each method and used to calibrate each camera. The '% det.' column is the percentage of frames for which the calibration grid could be extracted, the 'avg. pts' column is the average number of points used in each successfully detected grid (maximum 80), and the reprojection error is shown as standard deviation/maximum in the 'error' column

correspondences for correcting the radial distortion as in the method suggested by Hartley and Kang [5].

4.2 Comparing accuracy with marker corners, marker centers, OpenCV grid finder, and circular dots

Another experiment was performed to directly compare the marker corner and center methods, the OpenCV grid finder, and the traditional circular dot array used in calibrating cameras. The planar array of dots was used as a benchmark to compare the marker and the OpenCV grid finder methods. This allows comparison against what is likely the most common camera calibration method.

To fairly compare the four methods, identical poses were captured for each method. The cameras were mounted on tripods and the four calibration patterns were hinged together so they could be changed without changing the pose. The same number of images were taken for each method, for each of five cameras. The marker points are in the same positions in each image locations, all have eight rows and ten columns of points. Each point corresponds to a marker corner in the marker corner method, to a marker center in the marker center method, to a dot center in the circular dot method, and to an inner corner in the chessboard pattern for detection by OpenCV grid finder. The experiment setup is shown in Fig. 4.

The results are shown in Table 2. The circular dots were all found manually and so have a 100% detection rate, and the full 80 points found in each image. The markers were detected in all the frames with varying number of correspondence points. With all the four array types, the reprojection error was calculated by projecting onto a larger set of points. The marker system achieved equal or better accuracy than the circular dot method but was performed in seconds as opposed to several hours for manually locating and verifying the dot centroid locations. The OpenCV grid finder provided inferior calibration results, this could be due to the less number of calibration grids that were found, and/or a possible intrinsic weakness in using intersections corners.

4.3 Image set size

A practical question asked by one wishing to calibrate a camera is how many images are necessary. We performed an experiment on several camera configurations where we calibrated using image set sizes varying from 3 to 30 image frames per set. The accuracy of the intrinsic parameters extracted was evaluated by measuring the reprojection error when projected back to the full set of all frames. We randomly chose a subset of images from the full set, calibrated the camera using the marker center method, and then evaluated these intrinsic parameters. The intrinsic parameters were evaluated using the full set of images, calculating only the extrinsic parameters, and observing the reprojection error. The standard deviation and maximum reprojection error was averaged over all the calibration runs. Ten iterations were performed for each set size. Starting with set size of 2, we incremented the set size to 3, 4, 5, etc. up to a set size of 30.

Figure 5 shows the result for one of the cameras (Powershot S60); plots for more cameras can be found in [4]. The x-axis is the number of frames in a set, the top plot is the average standard deviation error, the bottom is the average maximum error. We observed that between 10 and 15 images are needed to calibrate a camera.

4.4 Sensitivity to lighting and focus

With all these methods each correspondence's location is determined by the position of borders or corners, the position is susceptible to the effects of the non-ideal properties of real cameras and lenses with respect to lighting change and blur. With an ideal pinhole a bright point in the scene maps to a single image plane point, however, with a finite aperture size and defocus the image plane point expands. With increasing irradiance and binary threshold decisions to find lines, corners and circular blobs, light/dark edges tend to move toward the darker side, corners on dark ARTag markers move inward, and dark circular blobs shrink. Lighting and

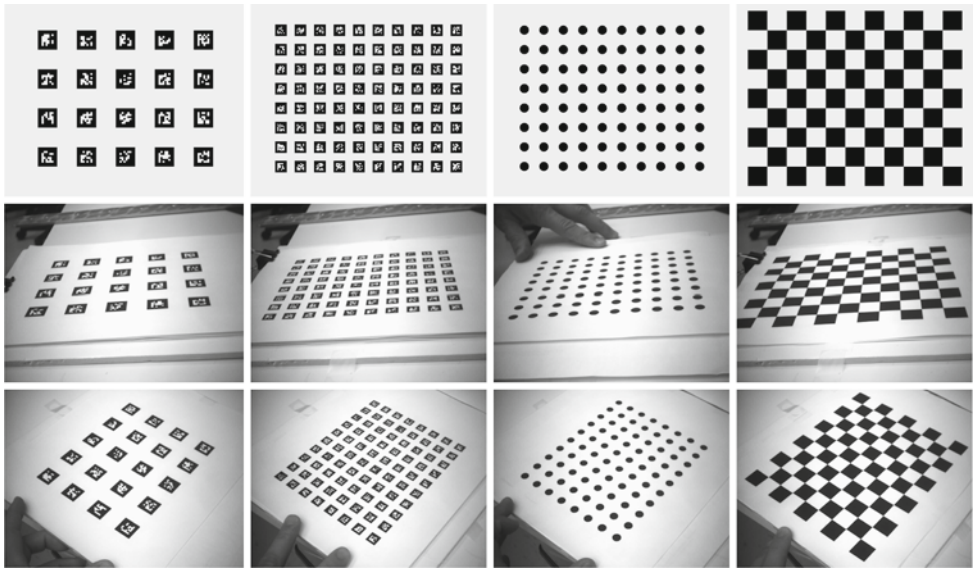


Fig. 4 Comparison of calibration using marker corner, marker center, circular blobs, and chessboard method using identical poses and locations of correspondence points

focus thus affect the image coordinates in the correspondence points used for calibration. With the greyscale Dragonfly camera ARTag marker corners moved by up to 1.5 pixels between light and dark exposures. The centroid of circular dots and the centers of ARTag markers move less than the marker corners due to the canceling effects of opposing changes from opposite sides. Also, the intersection point at the checkerboard corners will turn into two corners as either the opposite white or black squares merge due to bright or dim lighting. The OpenCV grid finder averages the distance between these two nearby corners found as endpoints of sides of closed quads, however, it is not clear how the following sub-pixel corner find operation will affect the reported position.

Therefore we would expect to see the calibration results to be affected by increasing light and lens/aperture non-ideal properties. We expect to see the ARTag corner positions move more than the centers, and we are unsure of how the OpenCV grid finder will respond. This was examined in an experiment where calibrations were performed at different exposure settings. With the Dragonfly IEEE-1394 camera it is possible to turn off the automatic shutter control and override it with a manual setting from 0.25 to 60 ms. We took 16 different exposure settings in this range at each of 20 camera-pattern poses and separated the images into 16 separate calibration runs. This experiment was performed both with the meddite ARTag array and a 4×5 checkerboard image. We also took 35 images with the normal auto-shutter of the ARTag meddite array and used the marker centers as the "full set" for evaluating the 16 calibration runs with each system. Figure 6 shows some images from the ARTag and checkerboard image sets. Four images from a single pose are shown for both the ARTag and chessboard array, some close-ups of a corner as it changes with shutter setting are shown.

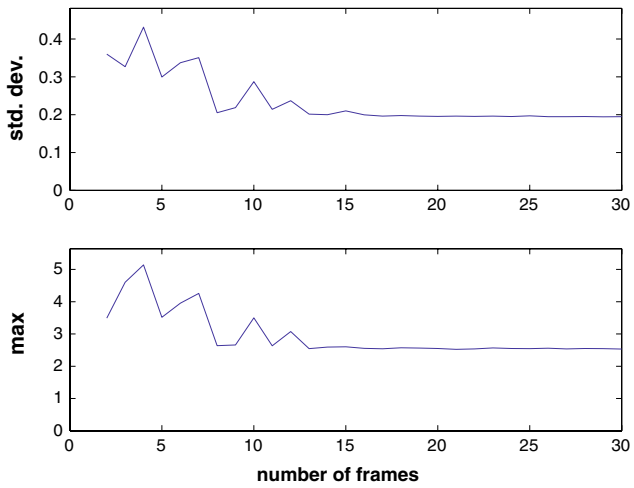


Fig. 5 Calibration accuracy vs set size

This provided us with two groups of images, one for the ARTag array and one for the checkerboard array. Each group contains 16 sets of 20 images, each set being from a different shutter setting. Correspondences in the checkerboard group are found with the OpenCV *CheckerBoardGuesses()* function. For the ARTag array, correspondences are found for both the center and corner methods. The intrinsic parameters are found for each of the three methods for each shutter setting and the accuracy evaluated by reprojecting to the "full set" of points.

Figure 6 shows a plot of the results, the ARTag center method shows a flatter, i.e., less sensitive response to lighting than the ARTag corner approach. The bottom of the ARTag corner curve is in the region of the 18 ms which the automatic shutter chosen with the lighting reflected off

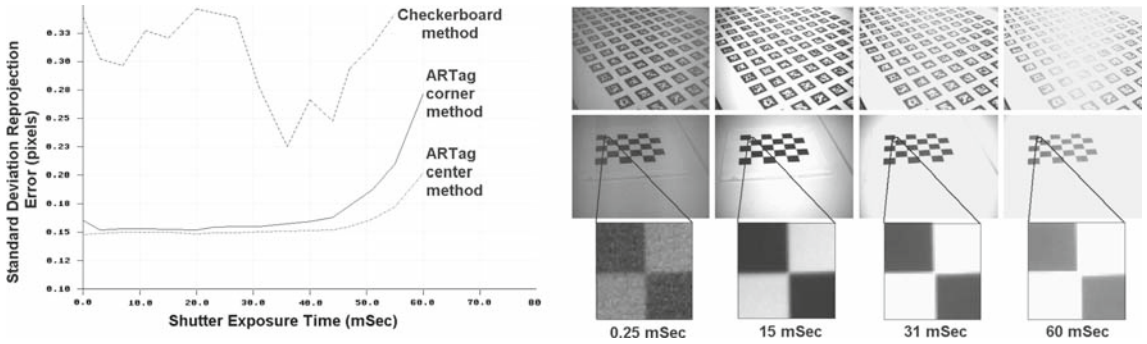


Fig. 6 Sensitivity of calibration accuracy due to lighting and non-ideal lens and aperture qualities. The camera shutter was manually controlled in 16 settings from 0.25 ms (*dark exposure*) to 60 ms (*bright exposure*) at each of 20 pose positions for both the ARTag and checkerboard array. Due to the point spread function corners move with increased lighting and defocus, the results show that the ARTag center method is less sensitive to the lighting (shutter setting) than either the ARTag corner or OpenCV grid finder

the ARTag array when auto-shutter was turned back on. The OpenCV grid finder provides a more erratic response, possibly due to only about 50% of images having the grid extracted, but the accuracy does seem to be higher closer to the 30 ms shutter setting chose by the auto-shutter mode. Our hypothesis of the ARTag center method being less sensitive to the ARTag corner method was verified, and both were found to be less sensitive than the checkerboard method.

5 Conclusions

Calibrating cameras is a time-consuming task despite the availability of calibration algorithms and software. This paper introduced a system where a camera can be calibrated fully automatically by simply taking images aimed at a self-identifying pattern. Experiments were performed calibrating several cameras in a short period of time with no manual intervention.

It was found that when using self-identifying markers, the best calibration results are achieved using the marker center instead of each corner; this reduces the number of correspondence per marker but decreases the sensitivity to lighting and focus. This is assumed to be analogous to using the centroid of a circular dot, the center will move less with blooming and defocus than will a marker or dot's edges.

It was found that the OpenCV grid finder is not as suitable for automatic accurate calibration since it functions well only when the array is small compared to the size of the image, and cannot fill the full extent of the image as is needed for accurate calibration as that the non-linear radial and thin prism distortion effects manifest themselves with greatest magnitude away from the image center. It was shown that the grid was often not detected when the pattern filled the image, and the grid had to be small relative to the image for repeatable enough detection to allow calibration.

A summary of the comparison between the OpenCV grid finder and the new self-identifying marker based system is that the marker system provides more accurate intrinsic parameters, verified by several experiments. We hypothesize that the increased accuracy is due to two reasons; more calibration points are found due to a more robust pattern detection, and these points extend further into the image corners. The system can use a pattern that extends beyond the camera view so that correspondences more completely fill the image, and utilizes all the input images (with reasonable lighting and focus/blur conditions). The OpenCV grid finder's "all or nothing" approach where all points in the calibration grid must be recognized causes many images to be unusable with zero correspondences provided.

Regarding how many frames are necessary for a good calibration, it was determined experimentally that reasonable calibration results can be obtained with as 10 frames for most cameras, but that a recommended number of frames is 15 or 20.

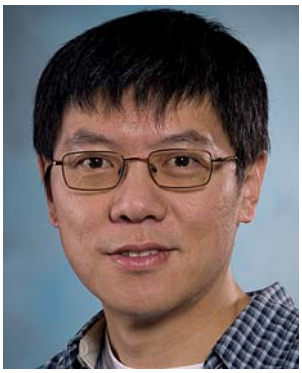
References

1. Open source comptuer vision library. In: http://www.intel.com/research/mrl/research/opencv
2. Fiala, M.: ARTag revision 1: A fiducial marker system using digital techniques. Technical Report NRC-47419/ERB-1117, National Research Council of Canada (2004)
3. Fiala, M.: ARTag: a fiducial marker system using digital techniques. In: CVPR'05, vol. 1, pp. 590 – 596 (2005)
4. Fiala, M., Shu, C.: Fully automatic camera calibration using self-identifying calibration targets. Technical Report NRC-48306/ERB-1130, National Research Council of Canada (2005)
5. Hartley, R.I., Kang, S.B.: Parameter-free radial distortion correction with centre of distortion estimation. In: IEEE International Conference on Computer Vision (ICCV'05), vol. 2, pp. 1834–1841 (2005)
6. Luong, Q., Faugeras, O.: Self-calibration of a moving camera from point correspondences and fundamental matrices. Int. J. Comput. Vis. **22**(3), 261–289 (1997)

7. Min, S.L., Kittler, J., Matas, J.: Empirical evaluation of a calibration chart detetor. Mach. Vis. Appl. **12**, 305–325 (2001)
8. Shu, C., Brunton, A., Fiala, M.: Automatic grid finding in calibration patterns using Delaunay triangulation. Tech. Rep. NRC-46497/ERB-1104, National Research Council of Canada (2003)
9. Soh, L., Matas, J., Kittler, J.: Robust recognition of calibration charts. In: IEE 6th International Conference on Image Processing and Its Applications, vol. 2, pp. 487–491 (1997)
10. Sturm, P.F., Maybank, S.J.: On plane-based camera calibration: A general algorithm, singularities, applications. In: IEEE Conference on Pattern Recognition and Computer Vision (CVPR99), vol. 1, pp. 432–437 (1999)
11. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE J. Robot. Automat. **3**(4), 323–344 (1987)
12. Willson, R.: Tsai Camera Calibration Software. http://www-2.cs.cmu.edu/People/rgw/TsaiCode.html (1995)
13. Zhang, R. Deriche, O.F., Luong, Q.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artif. Intell. **78**(1–2), 87–119 (1995)
14. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intelli. **22**(11), 1330–1334 (2000)

Author biographies

Dr. Mark Fiala is a computer vision researcher at Canada's National Research Council (NRC), where he works in the Computational Video Group centered in Ottawa, Ontario. His work includes fiducial marker systems, panoramic vision, and general computer vision topics such as image segmentation and camera calibration. He graduated from his Ph.D. in Electrical Engineering in 2002 in the field of panoramic computer vision. He also holds an Electrical Engineering B.Sc. and has spent over 5 years in industry in hardware design for panoramic imaging and telecom applications.



Chang Shu received Ph.D. in computer science from Queen Mary College, University of London, UK, in 1992, and B.Sc. from Harbin Institute of Technology, China, in 1985. He is currently a senior research scientist at the Institute for Information Technology, National Research Council of Canada. He is also an adjunct research professor at the School of Computer Science, Carleton University, Ottawa, Canada. From 1992 to 1996, he was a research associate in the Department of Mechanical and Aerospace Engineering at the Carleton University, Ottawa, Canada. His research has mostly been in developing geometric methods for solving problems in computer vision, computer graphics, robotics, and scientific computing. His current research interests include surface reconstruction, multiview stereo, and human shape modeling. He is a member of the IEEE.