Check2D software

Background and Case Study

Technical Report – C2D1

Dr Simon Goodwill

Sheffield Hallam University, UK

Email: s.r.goodwill@shu.ac.uk

January 2013

Abstract

Check2D is a camera calibration software package that has been developed by the authors for use in 2D kinematic analysis, when using camera images that exhibit lens distortion. The motivation for this study was to illustrate the need for accounting for lens distortions when using consumer level cameras with a zoom lens on wide setting or a wide angle lens. Distortions in the image, due to lens distortion and de-centering, influence the accuracy of reconstructed coordinates.

The planar calibration method was used to obtain the intrinsic and extrinsic camera parameters required for 2D kinematic analysis. A case study is described to illustrate the practical considerations which are involved in using this calibration technique. The accuracy of the method is quantified using a scale model of the swimming pool, and compared to the 2D direct linear transformation method (DLT). The root mean square error values were 19 mm and 83 mm for the *Check2D* and 2D-DLT methods respectively (in the direction of the swimmers motion). The better accuracy achieved using the *Check2D* software demonstrates the suitability of the planar calibration method for 2D kinematic analyses.

1. Introduction

Two-dimensional (2D) kinematic analyses of sporting activities are commonly performed using a single camera. A typical analysis aims to identify the position of an object on either an orthogonal plane (to the camera) or a perspective projected plane. This paper describes the use of a new camera calibration toolbox (*Check2D*) to illustrate the need for accounting for image distortions when wide fields of view are used in 2D kinematic analysis.

In 2D kinematic analyses, a wide field of view (FOV) is often used to achieve the required field-of-view when spatial constraints restrict the options for the position of the camera[1]. This may be in a laboratory environment or in a competitive sporting environment where stadia layout and/or access restrictions limit the choice of filming location. The effects of out-of-plane and perspective error, as well as methods for their correction, have been well documented[2,3]. However, the effects of lens distortion in images (for 2D kinematic analyses) have not been well documented. The impact of neglecting lens distortions has been demonstrated by previous authors[4], who reported reconstruction errors of 0.35 ± 0.27 m when filming calibration markers on a football pitch with a standard lens.

In this study, we focus on 2D kinematic examples where the object (athlete) moves across a planar surface. In general, this gives a perspective projection of the plane due to practical restrictions on where the camera can be placed. The Direct Linear Transformation (DLT) method[5] is used extensively in sport biomechanics to reconstruct position data from perspective projected images. However, a wide FOV (on consumer level cameras) typically induces distortions in the image[6]. A planar modification of DLT, termed 2D-DLT, calculates eight DLT coefficients necessary to reconstruct the 2D position of a point on a plane[7,8]. The accuracy of 2D-DLT reconstruction is dependent on a number of factors. Increasing the number of calibration points from a minimum of four (required to calculate coefficients) has been shown to reduce 2D-DLT reconstruction error[9]. Brewin and Kerwin[3] showed that reconstruction error was higher for points located outside of the volume enclosed by the calibration points. The effect of lens distortion can be calculated using 2D-DLT[10]. However, this method requires a relatively large number of calibration points to calculate the coefficients, and this is one possible reason why this method is rarely adopted in 2D kinematic analyses.

Another method to handle lens distortion in 2D kinematic analysis involves the segregation of the control volume plane into a grid, and the calculation of local DLT coefficients for each cell in the grid. A method based on this principle is used in the SIMI Motion software (SIMI Reality Motion Systems™). However, this procedure has two major practical issues which are (1) calibration points evenly distributed across the control volume are rarely available and (2) the time cost involved in accurately measuring these points makes this method inefficient.

An alternative to 2D-DLT calibration involves use of a non-linear camera calibration technique[11] to determine the intrinsic properties of the camera (focal length, optical distortion and de-centring distortion). This method uses multiple images of a planar pattern with known geometry. This can be a regular pattern of black dots on a white background, for example, but most commonly involves using a checkerboard pattern[1,6,12,13]. Silvatti et al.[12] concluded that the planar calibration method was a highly accurate alternative to nonlinear DLT for underwater 3D analysis of swimmers. The MATLAB Camera Calibration Toolbox[13] has been used to show that the planar calibration method gives a more accurate reconstruction of a tennis court geometry, when compared to the 2D-DLT method[1]. In that paper, a standard HD camcorder (JVC™ Everio GZ-HD40EK) was used on the widest field-of-view setting. Dunn[1] calculated lens distortions of up to 30 pixels, and demonstrated that the planar calibration method gave a more accurate reconstruction compared to the 2D-DLT method. This paper describes the practical application of the planar calibration method using the freely available *Check2D* software.

2. Planar calibration method

The most basic camera model is an ideal pinhole model[14]. In this type of model, the camera aperture is assumed to be a point and no lenses are used to focus the light on the image sensor. This means that a pinhole camera model can only be used as a first order approximation of the transformation from a real-world coordinate (3D) to an image plane coordinate (2D).

As such, a unique scale factor is all that is required to transform between real-world and image coordinates. This scale factor is typically obtained by placing an object of a known length into the plane of movement, and this plane must be orthogonal to the optical axis of the camera. The length of this object is measured (in image pixels) and this allows the digitised image coordinates to be translated into real-world coordinates. This method is commonly used in commercial performance analysis software such as Dartfish™. This calibration method assumes a linear pin-hole camera model. This assumption is valid if there are negligible optical distortions in the image. This would occur if (1) a telephoto lens is being used or (2) a high quality aspherical lens is used. However, in cases where a wide FOV is used and image distortions occur, a non-linear camera calibration should be performed for 2D kinematic analysis.

The camera model used in this paper is from the *OpenCV* library described by Bradski and Kaehler[6], and it is derived from Heikkilä and Silvén[15]. *OpenCV* is a library of computer vision programming functions developed by Intel. The projection of a point in the camera coordinate system to the camera image plane is given by,

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad [1]$$

where (X, Y, Z) are the coordinates of a 3D point in the camera coordinate system, (x, y) are the coordinates of the projected point in pixels, (c_x, c_y) is the principal point, f_x and f_y are the focal lengths expressed in pixel-related units, and s is a scale factor.

The radial and tangential lens distortion models used in *OpenCV* are well documented[6]. The radial distortion model is described by,

$$x_{model} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)x \qquad [2]$$

$$y_{model} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)y \qquad [3]$$

and the tangential distortion model is described by,

$$x_{model} = x + \left(2k_4\, y + k_5(r^2 + 2x^2)\right) \qquad [4]$$

$$y_{model} = y + (k_4\,(r^2 + 2y^2) + 2k_5 x) \qquad [5]$$

where,

$$r^2 = \mathrm{x}^2 + \mathrm{y}^2 \qquad [6]$$

and (x,y) is the location of the distorted point in the image,(x_{model},y_{model}) is the new location calculated by the correction model, and $k_{1\text{-}5}$ are the coefficients which are solved in the calibration optimisation routine. The group of parameters (f_x, f_y, c_x, c_y, $k_{1\text{-}5}$) are collectively called the camera intrinsic parameters.

The extrinsic camera parameters define the translation (T) and rotation (R) of the camera with respect to the global origin system. These parameters are required to reconstruct the projected points on the image to the global origin system. The reconstruction method used in this study is the same as that described by Dunn et al.[1].

The work in this paper uses a software package called *Check2D* which runs on the Microsoft™ Windows operating system. *Check2D* was developed by the authors and uses an implementation of the planar calibration method from *OpenCV*. The description of the method given here focuses on the practical considerations which are involved in the use of the camera calibration technique, rather than the mathematics of the optimisation routine.

3. Application of the planar calibration method

3.1 Practical considerations

Check2D uses the *OpenCV* library to compute the intrinsic camera parameters. This library uses multiple views of a planar object which has many individual and identifiable points of known geometry. This object should be moved across the full range of the camera view and be held at various angles. In general, the object is a printed checkerboard which has been securely attached to a flat, rigid object. The checkerboard needs a white (or black) border (Figure 1). The calibration procedure in *OpenCV* only requires two distinct views of a checkerboard to be able to calculate a solution for the intrinsic camera parameters[6,11]. However, this is analogous to the constraint that only two data points are required to calculate the parameters of a linear trend line ($y = mx + c$). Clearly, a larger data set will improve the accuracy of the trend line parameters for data points which contain noise.

Figure 1. A typical image of a checkerboard showing the white space around the squares. This space is used to locate the board in the image. Each corner has been detected, and the square search area is superimposed on the image.

The algorithm that *OpenCV* uses to solve for the focal lengths and principal points is based on Zhang[11], but *OpenCV* uses a different method[16] to solve for the distortion parameters. When using Zhang's algorithm, the position of the checkerboard must be moved significantly between views. Otherwise, the matrices of points used to solve for calibration parameters may form an ill-conditioned (rank deficient) matrix and this will either lead to an incorrect solution or no solution. This is because two planes which are parallel to each other provide the same information as a single plane, albeit there are more detected checkerboard corners[11,17].   Furthermore, *OpenCV* uses the vanishing point present in perspective views of the checkerboard to obtain an initial estimate of the intrinsic camera parameters. Therefore, the solution will either be incorrect or slow to convergent when the image set mainly contains checkerboards with plane normals that are parallel to optical axis of the camera. Bradski and

Kaehler[6] qualitatively describe the collection of checkerboard images needed for calibration as “a rich set of views”.

It is very difficult to prescribe the optimum number of checkerboard views required for an accurate camera calibration due to the large number of variables involved (image resolution, number and size of checkerboard squares, extrinsic parameters of checkerboard, amount of lens distortion, etc.). Bradski & Kaehler[6] advise a minimum of ten images of a 7-by-8 (or larger) checkerboard, and reinforce that this advice is only valid when the checkerboard is moved sufficiently between images.

The first stage in the calibration process is to automatically extract the image coordinates of the squares on the planar checkerboard, from a series of calibration images. This extraction method is multi-threaded in *Check2D* to reduce the execution time on a multi-core CPU. The extraction method searches for each checkerboard corner across a window (Figure 1) of size win_t . The size is defined by Bouguet[13] using,

$$win_t = \max\left(\frac{\mathrm{w}}{128}, \frac{\mathrm{h}}{96}\right) \qquad [7]$$

where w and h are the image width and height respectively.

It is important that the entire image is covered with points to maximise the accuracy of the camera model, especially when calculating the lens distortion parameters. *Check2D* enables the user to do a visual check of the coverage by plotting all extracted corners on a single image. This allows the user to make a qualitative assessment of the level of evenness in the distribution. If the points are not evenly distributed across the image, then the validity of the camera model cannot be quantified for the entire image. This is analogous to the task of calculating the parameters of a polynomial trend line $(y = ax^2 + bx + c)$ in the range

$0 < x < 100$. For the coefficients to be correctly calculated, the regression analysis must use data points that are evenly distributed from x=0 to x=100.

3.2 Model assumptions

The default assumptions for the camera calibration model are to calculate the principal point and both radial and tangential distortions. However, there are cases when the principal point should be assumed to be at the centre of the image plane, and/or that there are no lens distortions. These assumptions are discussed below.

Typically, the principal point should be calculated which means that the model will estimate the point (c_x, c_y). The principal point is the intersection of the optical axis and the sensor plane. This location will only be at the centre of the image plane if the lens and sensor are perfectly aligned. If the location of the principal point (c_x, c_y) has been incorrectly calculated by the model, the user should override the model, and force the assumption that the principal point is at the image centre. In these cases, it is at the discretion of the user to decide if the principal point has been calculated correctly. This decision should be based on the consideration as to whether the lens and sensor are likely to be physically misaligned by the same magnitude as calculated by the model.

The radial and tangential distortions should initially be calculated by the model, as defined by equations 2 to 5. *Check2D* reports the magnitude of the two different distortion components, as calculated by these equations. Two examples of lens distortion are given (Figure 2) for two different cameras, and two different fields-of-view. The maximum radial distortion for the AXIS P1346 camera (Axis™ Communications, 1920x1080 pixels, progressive scan, 55° viewing angle) is approximately 60 pixels and the maximum tangential distortion is approximately 1 pixel (Figure 2 (a)). It is well documented that the radial distortion is the dominant distortion for most consumer level cameras[6,11], and in this case study it would be advisable to rerun the solver using the assumption that only radial distortions are assumed. The maximum radial distortion for the Sony HDR HC9 (Sony™ Corporation, 1920x1080, interlaced, 35° viewing angle) is approximately 5 pixels and the maximum tangential distortion is approximately 3 pixels (Figure 2 (b)). Conventionally, the radial distortion would either be pincushion distortion or barrel distortion[11], and the magnitude would increase with increasing distance from the principal point. In this case, this does not occur, and is an example where a high order model has been incorrectly used and has led to numerical instability[11]. In this case it would be advisable to rerun the camera calibration solver assuming that there are no radial or tangential distortions.

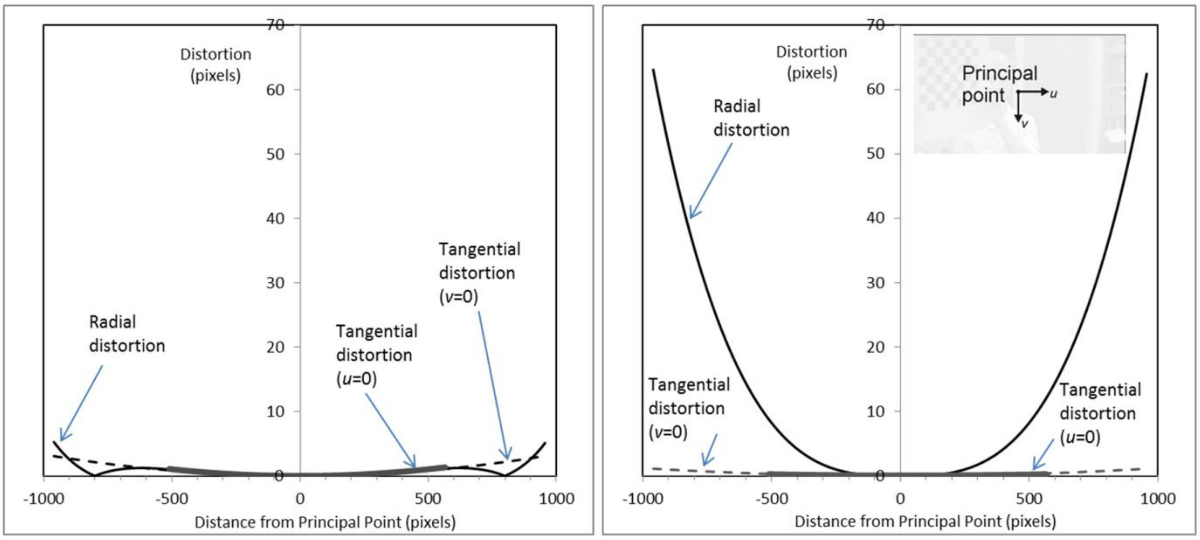

Figure 2. The magnitudes of the components of lens distortion for points along the *u* and *v* axis, for (a) Axis P1346 camera (viewing angle 55°) and (b) Sony HDR HC9 camera (viewing angle 35°). Radial distortion is symmetrical around the principal point and therefore can be represented by a single curve.

3.3 Evaluation of model

A Levenberg-Marquardt optimization algorithm is used to minimise the root mean square error (RMSE) in the reprojection. This is calculated from the distances between the extracted checkerboard points and the projected points (calculated from the board extrinsic parameters and the converged solution for the camera model). The optimization algorithm will terminate when either the maximum number of iterations has been reached or the epsilon (ε) value has reached a value less than the termination criteria. The value of ε represents the magnitude of the change in the calibration parameters between iterations and this value should converge towards zero in the optimisation algorithm. This indicates that a stable solution has been attained.

It is well documented that the accuracy of the calibration model is highly dependent on the board being rotated through a wide range of angles[11,17] (Figure 3). A board held orthogonal to the optical axis has a plane normal vector of [0, 0, -1] in the camera coordinate system. In practice, the board is predominantly rotated about the x and y axes, and these angles are defined as α and β, when the plane is projected in the XZ and YZ plane respectively. The value of these two angles can be exported from *Check2D* and (when plotted appropriately) can be used by the user to qualitatively conclude whether the angle variation is sufficient. In a typical calibration data set, there will be insufficient data points to apply a valid statistical method, such as the Chi-squared distribution test, to quantify the evenness of the distribution of α and β.

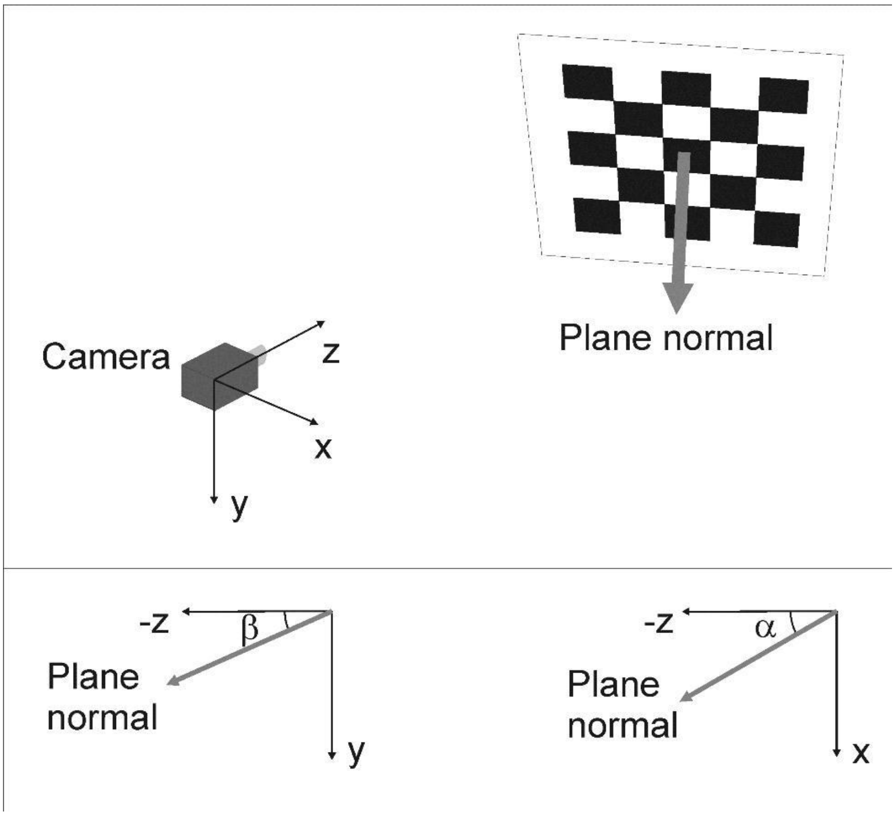

Figure 3. Definition of the camera coordinate system and the projected angles (α is the angle projected in the YZ plane and β is angle project in the XZ plane).

The maximum and minimum values of α and β are limited by (1) size of the square, (2) distance between board and camera, and (3) focal length of camera. In Figure 1, the search window (of size (win_t)) is plotted on the checkerboard at each corner. The board must not be held in a position where the search windows overlap as this may lead to errors in the automated detection procedure.

4. Case study

The benefit of the planar calibration method used in *Check2D* can be demonstrated by considering a typical usage example. In this example, the objective was to measure the horizontal velocity of a backstroke swimmer in a swimming pool. It was assumed that the leading edge of the swimmer's head was a valid discrete digitisation point to approximate the centre-of-mass (horizontal) motion. The swimmer's head was assumed to move in the plane of the surface of the swimming pool.

The testing was conducted at Ponds Forge International Swimming Pool in Sheffield (UK). The camera was placed on a tripod in the spectator gallery. An AXIS P1346 camera was used (Axis™ Communications, 1920x1080 pixels, progressive scan). The camera (horizontal) angle of view was 54° and parts of the image were masked for privacy reasons. The swimmers were only in lanes 5 to 9, and the control volume was 20 m x 12.5 m (Figure 4). The camera was placed on a tripod at the highest location in the spectators viewing gallery, but still required the widest setting on the zoom lens to be used to achieve the required field-of-view. *Check2D* was used to calculate the intrinsic parameters of the camera (including lens distortion).

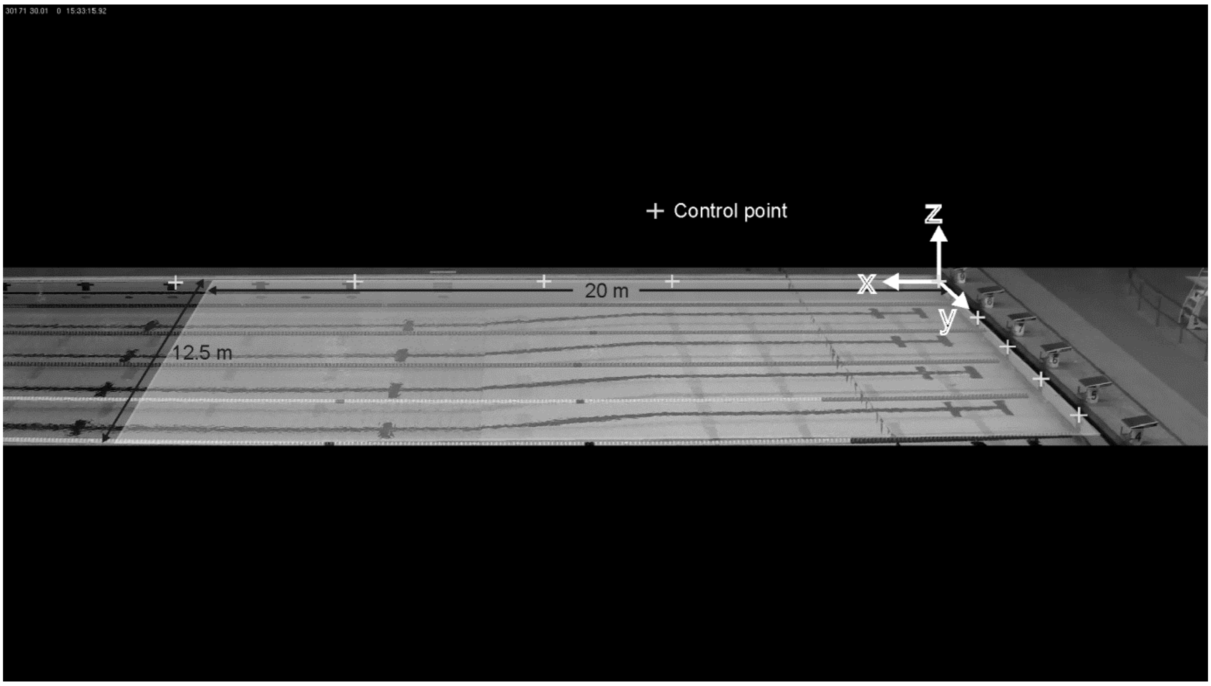

Figure 4. A typical view of the swimming pool showing the world coordinate system of the 20 m x 12.5 m control volume. The nine (known) control points are denoted by crosses.

The camera was focused on the lane rope at the centre of the image, and manually locked to prevent any changes. This camera has a very large depth of field which meant that checkerboards held two metres away from the camera are in also in focus. The camera focus and zoom were locked after filming, which allowed the camera to be calibrated when back in the office. The camera was calibrated using 30 images of an 8-by-8 checkerboard with square size of 30 mm (Figure 5). The checkerboard corners which were extracted from all images are superimposed on the first image to illustrate that the entire sensor was covered. The intrinsic parameters for the camera were calculated for the swimming pool (Table 1). The tangential distortion was not calculated as it was negligible compared to the radial distortion. The model gave an RMSE reprojection error of 0.296 pixels.

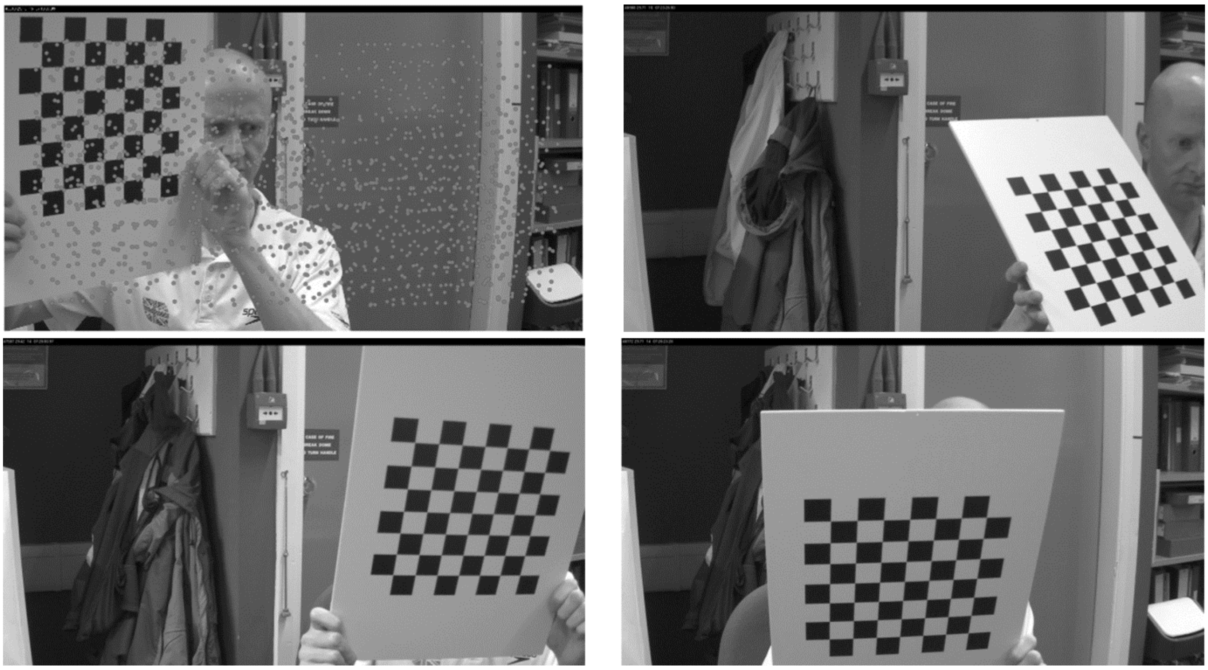

Figure 5. Typical views of a checkerboard moved around the view. The first image shows the extracted checkerboard corners from all images.

The *Check2D* software requires the location of at least 4 control points to determine the translation (T) and rotation (R) of the camera with respect to the world origin system (Figure 4). The location and orientation of the camera (T and R) are referred to as the extrinsic camera parameters. In the current study, nine control points were used to determine the extrinsic camera parameters. These control points are located on the swimming pool wall (at water level) as this is the only static object in the control volume plane, and are superimposed on the image (Figure 4).

Table 1. Intrinsic and extrinsic camera parameters for swimming pool and scale model experiments

		Swimming pool	Scale model
Focal length (pixels)	f_x	2421.2	2378.3
	f_y	2410.4	2367.8
Principal point (pixels)	c_x	918.7	969.8
	c_y	487.3	507.8
Radial distortion	k_1	-0.365	-0.371
	k_2	0.182	0.203
	k_3	0	0
RMSE reprojection (pixels)		0.296	0.434
Translation (mm)	T	[10038 -800 41631]’	[91 -12 412]’
Rotation	R	[0.0468 2.472 -1.877]’	[0.0591 2.468 -1.865]’
RMSE reprojection (mm)		25.2	0.4

The main purpose of the *Check2D* software is to transform the projected points on an image to the world coordinate system. However, the accuracy of this transformation cannot easily be validated in this case study because (1) the control volume plane is very large and (2) it is impractical to place markers at known locations on the water surface.

The validity of the method can be tested using a scale model of the experiment, as done by Dunn et al.[1]. A 1:100 scale model of the swimming pool is represented by a checkerboard consisting of 22 by 12 squares with a square size of 10 mm (Figure 6). The camera (horizontal) angle of view was set at 54° (as for the swimming pool study) and the lens was manually focused. In the scale model environment, the camera was positioned using an iterative method, to achieve an equivalent perspective view of the control volume. This was done by overlaying the (scaled) positions of the nine control points onto a live image feed from the camera. The camera was then translated and rotated until the points were aligned correctly on the image.

The intrinsic and extrinsic camera parameters for the scale model environment are shown in Table 1. The rotation R is represented by a Rodrigues vector[6]. The intrinsic parameters are similar and the different focus setting is the likely cause of the differences. The maximum difference in the scaled translation components was less than 10 mm which illustrates the efficacy of this approach.

The accuracy of the *Check2D* method was compared to the 2D-DLT method. The nine control points (that were used to obtain the extrinsic camera parameters in the swimming pool experiment) were used to obtain the DLT coefficients. The RMSE reconstruction for the nine control points was 0.9 mm. For comparison, an additional set of DLT parameters were also calculated using known positions in the scale model experiment, which would be on the water surface in the swimming pool experiment. In this case, a total of 16 DLT control points were used and the locations of the additional seven points are also superimposed on the image (Figure 6). The RMSE reconstruction for the nine control points was 1.3 mm. Clearly this would have been impractical for the swimming pool study, but adheres more closely to the principles on which the 2D-DLT method is based where by the control points encompass the control volume.

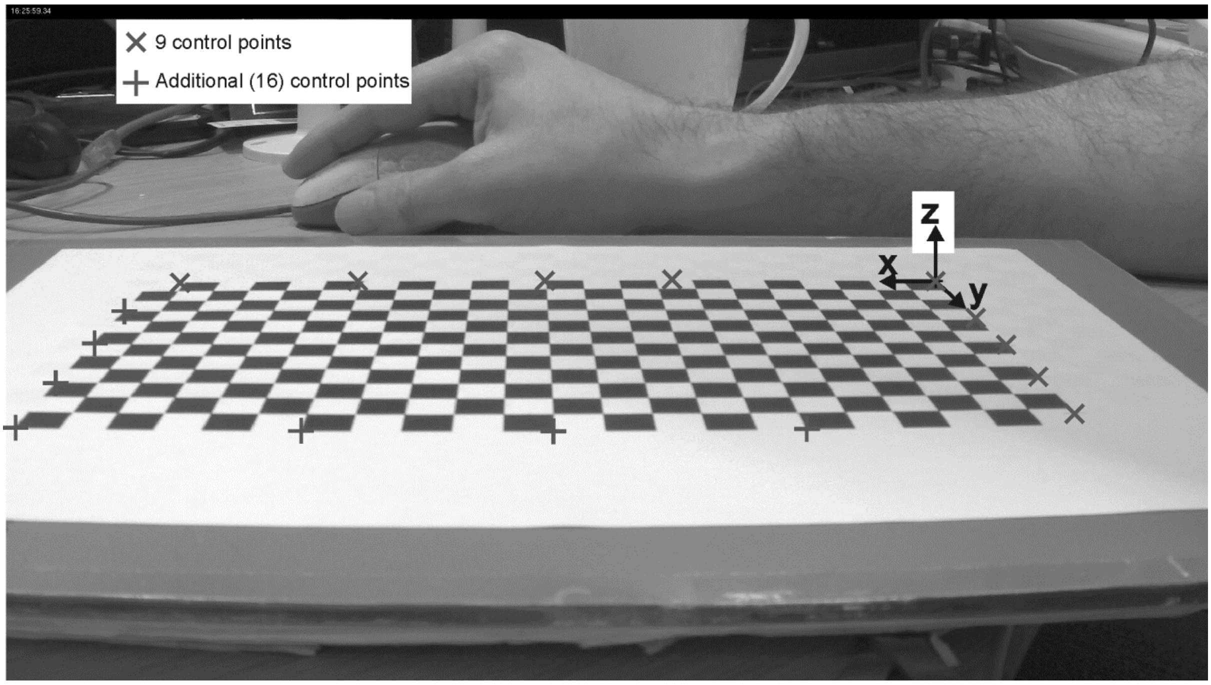

Figure 6. 1:100 scale model of swimming pool control volume, showing the world coordinate system. Each square is 10 mm. The control points (for the DLT method) are superimposed as crosses on the image.

For the scale model experiment, each checkerboard corner was automatically extracted. This gave a sample of reconstruction points (N=231). These points were reconstructed using standard 2D-DLT routines[18] and using *Check2D*.

The RMSE between reconstructed and real world coordinates was calculated separately for the *x* and *y* coordinates denoted as $RMSE_x$ and $RMSE_y$ respectively. These were calculated using,

$$RMSE_x = \sqrt{\sum_{i=1}^{N} \left(P_{i(x)} - p_{i(x)}\right)^2 / N} \qquad [8]$$

$$RMSE_y = \sqrt{\sum_{i=1}^{N} \left(P_{i(y)} - p_{i(y)}\right)^2 / N} \qquad [9]$$

where P_i is the real world location, p_i is the reconstructed location and N is the number of points used. The overall resultant *RMSE* was also calculated using,

$$RMSE = \sqrt{\left(RMSE_x^2 + RMSE_y^2\right)} \quad [10]$$

The resultant RMSE for *Check2D* is the lowest of the three reconstruction methods (Table 2). The RMSE is lower for the DLT method when more control points are used.

Table 2. Root mean square error, maximum error, mean error and standard deviation for reconstruction points (N=231) in scale model.

	Check2D	DLT (9 points)	DLT (16 points)
$RMSE_x$ (mm)	**0.19**	1.21	0.83
$RMSE_y$ (mm)	0.36	1.01	**0.31**
RMSE (mm)	**0.41**	1.57	0.88
Maximum error - *x* (mm)	**0.20**	2.88	1.28
Maximum error - *y* (mm)	**0.56**	2.47	1.03
Mean error - x (mm)	**-0.15**	0.27	-0.18
StDev -x (mm)	**0.12**	1.18	0.81
Mean error - y (mm)	-0.19	0.79	**0.12**
StDev -y (mm)	0.31	0.64	**0.29**

The objective of the swimming pool experiment was to measure the velocity of the swimmer. The $RMSE_y$ value was marginally lower for DLT (16 control points) than planar calibration in the *y* direction. However, swimmers move predominately along the x-axis and therefore it can be concluded that the $RMSE_x$ is the more important error term. The $RMSE_x$ is 0.19 mm and 0.83 mm for *Check2D* and DLT (16 control points) respectively, emphasising the increased accuracy obtained using the planar calibration method for this camera.

The maximum error (for both x and y axis) is lowest for the *Check2D* method (Table 2). A comparison of the maximum errors for the *Check2D* and DLT (nine control points) methods, 0.2 mm and 2.88 mm respectively, emphasises the high inaccuracy associated with using the DLT with these nine control points.

The mean errors (in the x axis) are similar for both *Check2D* and DLT (16 control points). However, the standard deviation of the errors is considerably higher for the DLT (16) method compared to *Check2D,* being 0.81 and 0.12 respectively. The high standard deviation value signifies that the errors have a large range for the DLT (16) method. This illustrates that, even when using control points that surround the control volume, the DLT method has significantly higher errors than *Check2D.*

It is assumed that the error values obtained for the scale model experiment can be applied to the swimming pool experiment by scaling appropriately. For example, $RMSE_x$ would be 19 mm for the *Check2D* method in the swimming pool experiment. Furthermore, the scaled maximum error for DLT (9 points) would be 288 mm (in the x axis).

5. Conclusions

It has been shown that the intrinsic and extrinsic parameters of a consumer level camera can easily be obtained using the planar calibration method. The calibration object is a checkerboard that can be printed on an office laser printer and attached to a rigid flat surface. The *Check2D* software application has been developed to facilitate this task. The user must decide what level of complexity is assumed for the model, and can use the diagnostic output from *Check2D* (radial/tangential distortion components, etc.) to inform that decision. The errors associated with using this method for a 2D kinematic analysis case study have been shown to be significantly smaller than those calculated using the direct linear transformation (DLT) technique.

References

1. Dunn M, Wheat J, Goodwill S, Haake S. Reconstructing 2D planar coordinates using linear and non-linear techniques. In: *Proceedings of the 30th International Conference on Biomechanics in Sports*. Melbourne, Australia; 2012:380 – 383.

2. Siha BL, Hubbard M, Williams KR. Correcting out-of-plane errors in two-dimensional imaging using nonimage-related information. *Journal of Biomechanics*. 2001;34(2):257–260.

3. Brewin MA, Kerwin DG. Accuracy of Scaling and DLT Reconstruction Techniques for Planar Motion Analyses. *Journal of Applied Biomechanics*. 2003;19(1):79–88. Available at: http://journals.humankinetics.com/jab .

4. Alcock A, Hunter A, Brown N. Determination of football pitch locations from video footage and official pitch markings. *Sports Biomechanics*. 2009;8(2):129–140. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19705764 [Accessed February 13, 2013].

5. Abdel-Aziz YI, Karara HM. Direct linear transformation from comparator coordinates into object space coordinates in close range photogrammetry. In: *Proceedings of the Symposium on Close-Range Photogrammetry*. Falls Church, VA: American Society of Photogrammetry; 1971:1–18.

6. Bradski G, Kaehler A. Camera Models and Calibration. In: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media; 2008:370–404.

7. Walton JS. Close-range cine-photogrammetry: a generalized technique for quantifying gross human motion. 1981.

8. Kwon Y-H. Object plane deformation due to refraction in two-dimensional underwater motion analysis. *Journal of Applied Biomechanics*. 1999;15:396–403. Available at: http://journals.humankinetics.com/jab .

9. Mclean SP, Vint PF, Hinrichs RN, et al. Factors affecting the accuracy of 2D-DLT calibration. In: *28 Annual Meeting of the American Society of Biomechanics*. Portland, OR, USA; 1981:246–247.

10. Wenhao F, Li J, Yan L. Creation of distortion model for digital camera (dmdc) based on 2d Dlt. In: *20th International Society for Photogrammetry and Remote Sensing*. Istanbul, Turkey; 2004:979–984.

11. Zhang Z. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In: Vol 00.; 1999:666–673.

12. Silvatti AP, Dias FAS, Cerveri P, Barros RML. Comparison of different camera calibration approaches for underwater applications. *Journal of biomechanics*. 2012;45(6):1112–1116. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22284990 [Accessed February 12, 2013].

13. Bouguet J. Camera Calibration Toolbox for Matlab. Available at: www.vision.caltech.edu/bouguetj/calib_doc/ [Accessed January 1, 2013].

14. Hartley R, Zisserman A. Camera Models. In: *Multiple View Geometry in computer vision*. Cambridge University Press, UK; 2003:154.

15. Heikkilä J, Silvén O. A Four-step Camera Calibration Procedure with Implicit Image Correction. In: *Computer Vision and Pattern Recognition*. San Juan, Puerto Rico; 1997.

16. Brown D. Close-range camera calibration. *Photogrammetric Engineering*. 1971;37(8):855–866.

17. Sturm PF, Maybank SJ. On plane-based camera calibration: A general algorithm, singularities, applications. In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Fort Collins, Colorado, USA: IEEE Comput. Soc; 1999:432–437. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=786974.

18. Meershoek L. Matlab routines for 2-D camera calibration and point reconstruction using the DLT for 2-D analysis with non-perpendicular camera angle. 1997. Available at: http://isbweb.org/software/movanal.html.