Checkerboard Pose and Tracking

For use as a simple fiducial marker and navigational aid

Nissanka Weerekoon
Department of Mechatronics Engineering
University of Canterbury
Christchurch, New Zealand
Email: naw52@uclive.ac.nz

Richard Green
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
Email: Richard.green@canterbury.ac.nz

***Abstract* — In recent times pose estimation and tracking of fiducial markers has involved complex pattern libraries and recognition algorithms for multiple markers in a given scene. This paper proposes a simplified method for single checkerboard recognition and pose estimation. This provides the basic functionality of a fiducial marker and could be used as a navigational aid for robots without any rotational ambiguity. The proposed method involves: adaptive thresholding; detection of internal checkerboard corner locations with sub-pixel accuracy; and estimation of the extrinsic parameters to give rotational and translational information in real world coordinates. The method was tested with two different image resolutions and found to be invariant of background lighting within 2m of the camera however was affected by overhead lighting conditions.**

Keywords- checkerboard; extrinsic parameters; pose; fiducial markers

I. INTRODUCTION

The motivation for this research came from the need of an autonomous go-kart to track a target object. The simple task of identifying a target and steering towards it is one basic element of autonomous control. Pose determination in six degrees of freedom (6-DOF) would allow the go-kart control system to judge horizontal position, distance away and rotation of the target. Alternatively, the go-kart could determine its current roll, pitch and yaw.

Fiducial markers are used in augmented reality as well as object tracking applications. These are often complex patterns which can be individually identified using algorithms specially designed for the particular type of pattern.

The goal of this study is to propose a simple method for pose determination of a single target object in 6-DOF which does not require the use of complex patterns and algorithm libraries. This could be used as an initial navigational aid for an autonomous go-kart as well as a simple fiducial marker.

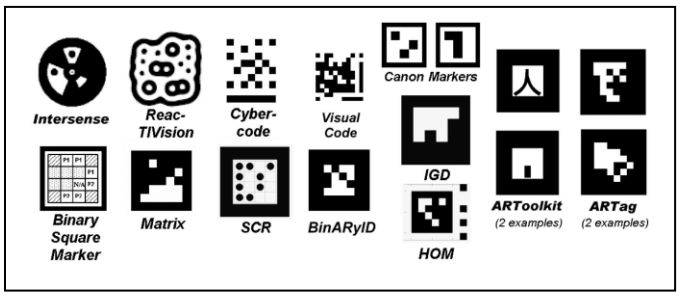

Figure 1: Common fiducial marker types

II. BACKGROUND

A. Fiducial Markers

Fiducial markers are artificial patterns used to aid recognition in augmented reality (AR), pose determination, robot navigation and tracking. Reliable markers provide easily detectable points for correspondence between images, irrelevant of the orientation [1]. In recent times this has led to the formation of complex marker patterns which are uniquely identifiable. These require detection algorithms tuned to a collection of markers with common features. Reference [1] also explains that detection methods typically have two stages, hypothesis generation followed by verification/identification. As seen in Fig. 1 markers can have almost any shape and style. A distinct border is commonly included for ease of background separation.

One fiducial marker library is ARTag. This utilises a method by M. Fiala involving digital encoding to identify as many as 2002 possible marker patterns [2]. ARTag focusses on very low false positive detection and inter-marker confusion rates. This would be particularly beneficial for multiple point correlations in AR and tracking multiple objects or multiple points on a single object. ARTag was an improvement on ARToolkit developed by Hirokazu Kato in [3] in which fiducial markers were used especially for AR in applications such as video conferences. Some of the main improvements of ARTag are the use of an edge-based quadrilateral finder and the removal grayscale thresholding. The ARTag system is also able to maintain marker recognition if a large portion of the boarder is occluded, whereas the ARToolkit is not.

While most methods appear to use geometric point search based pattern identification, some such as ARToolkit use template matching to identify specific user ID markers [3]. This introduces the need for predefined templates to be added to the system each time a new marker is added, which would require further understanding and implementation by the user. The template matching in ARToolkit is done on image regions where an identified contour comprises of four vertices [3]. However a pattern such as a checkerboard with far more detection points may provide less false positive detections during early processing steps.

In contrast to the marker systems mentioned, fiducial markers have also been developed for applications other than AR. CALTag is a fiducial marker type designed for use in camera calibration [4]. It is intended to support fully automatic camera calibration, especially involving multiple cameras, where occlusion of the calibration pattern often occurs. The single marker pattern used by CALTag is based on a black and white checkerboard. The algorithm employs an OpenCV [5] function to find saddle points at approximate locations of square edges, and thereby acquiring sub-pixel accuracy of calibration point locations [4].

All the fiducial marker methods explained thus far require patterns to be generated to conform to a particular collection. Complex algorithms are then used to individually identify each marker. Knowledge of the associated libraries and setup procedure is therefore required of the user. A similar amount of work would be required by advanced users wishing to track multiple objects compared with novice users wanting single object information. In addition, the mentioned marker types incorporate high levels of detail within the pattern which would limit the maximum distance from which one could be recognised. These markers would therefore be less useful for robot navigation type applications where larger distances are required compared to AR. The method proposed in this paper involves a simple checkerboard marker, though limited in uniqueness, form part of a simple single target tracking and pose estimation platform.

B. *Pose Estimation*

Object pose is usually calculated as a rotation and translation in 3D relative to a defined coordinate space. This is most commonly and easily done for planar objects using the concept of planar homography between the camera CCD and target object planes [6]. Calibration of the camera provides the intrinsic parameters which can be used along with the target object projection on the image plane (CCD) to compute the extrinsic parameters, as explained in [6]. The resulting calculation of object pose is often referred to as estimation; most probably due to the dependence on image resolution and processing approximations. The pose of non-planar objects can also be estimated as in [7] using SIFT or SURF algorithms to detect and match feature points with a database.

According to [8] marker shapes also influence the accuracy of pose estimation. Square and circular markers are found to be most common, both having their own merits: The pose of a square can be calculated from its four vertices and is most often unique. The pose of a circular maker, though calculated from its entire surrounding contour thereby minimising occlusion error, is ambiguous when only one marker is present.

A further technique used for pose estimation in [9] incorporates the use of a self-encoded marker. This allows the pose to be determined without the need to search for corresponding feature points in the image. This would increase processing speed; however increase in marker detail would limit operating distance as previously mentioned. It should be noted that the marker used in [9] is not fully planar and the application is such that the marker consumes most of the image. This almost eliminates the problem of identifying the marker as would be encountered in robot navigation.

C. *Camera Calibration*

In order to obtain reliable information about the 3D world from a 2D image, camera calibration is a required step. This is especially important for cameras with large lenses (such as fish-eye cameras) where distortion around the edges dramatically affects the image. Calibration should be done once prior to obtaining image data from a camera for the first time. The intrinsic parameters of the camera can then be calculated and used thereafter. Calibration is usually done using planar checkerboards or circular patterns rather than 3D object for ease of data management and modelling during the calibration process. Typically, images of the target (checkerboard) are captured from multiple viewpoints. Calibration points, internal corners in the case of a checkerboard, are then identified in the image corresponding to known points on the target [4]. Zhang's proposed method of camera calibration [10] is commonly accepted and used by the OpenCV library as well as a basis for other specialised calibration applications ([11], [12], and [13]).

III. METHOD

This section outlines the proposed method in terms of hardware, software and algorithms to achieve basic recognition and pose estimation of a single checkerboard target.

A. *Additional Hardware and Software*

The proposed method was developed on a laptop machine containing an Intel i7 core and 6GB of RAM. To facilitate capturing video, both for live analysis and recording, two types of standard webcam were used. Main development was done using a VGA quality camera inbuilt to the laptop which captured 640x480 resolution images. A 2MP Logitech USB webcam was also used as a comparison. Version 2.4.3 of the popular image processing library OpenCV was used to help implement the proposed method. The camera setup is shown in Fig. 2.

B. *Calibration*

The calibration process is only completed once for each camera used. This is a setup step that is an aside to the main method discussed in this paper; however it is relied upon by the method. The method outlined by the OpenCV Calibration module [5] was followed for this. This process gave two matrices; the camera intrinsic parameters and the distortion coefficients, which were saved for use later in the method.

Figure 2: Cameras used

C. Corner Point Identification

The proposed method determines the location of the internal corner points (ICPs) over two passes. On the first pass, ICPs on the checkerboard are located within the image accurate to within one pixel. If the number of ICPs found matches the number expected, processing moves to determine ICPs to sub-pixel accuracy explained later in the method. The expected number of ICPs is previously computed using the checkerboard dimensions given to the algorithm by the user as shown in equation (1).

Expected ICPs =
(board_height -1)(board_width -1)* (1)

In order to find the approximate location of ICPs in the image on the first pass, the captured image is converted to grayscale. This allows adaptive thresholding of the image to make the appearance of a black and white checkerboard very pronounced.

Firstly, the grayscale image is split into a black image and a white image to successive thresholding is applied up to a set value. The contours that still remain in each image are then checked for approximate 'squareness' within tolerances. If both images have more squares than the minimum set limit the image is marked positive for containing a checkerboard. This method provides a fast check for the presence of a checkerboard and skips further extensive processing if one is not found. As the method checks for internal squares, the minimum dimensions of a compatible checkerboard are set at 3x3 due to the need for at least one internal square.

If a checkerboard is found in the image, it undergoes adaptive thresholding and dilation to further pronounce the presence of squares. This makes detection more robust to variable lighting conditions that a simple threshold. The image is given a boarder corresponding to the average pixel value so that squares moving off the edge of the image can still provide a complete contour. The method then looks for quadrangles in the image and orders these along with information on how they are connected. This leads to the creation of an ordered set of corner points – the ICPs. If the number and ordering of quadrangles matches that of a clean checkerboard image, the ICPs are passed for further refinement.

D. Sub-pixel Accuracy

Once as approximate location of the ICPs is known, iteration over each corner is performed to refine its location to sub-pixel accuracy as shown in Fig. 3 If q is located exactly at the corner point, the vector q-p will be orthogonal to the image gradient DI_p at p. This provides a basis to determine the exact location of q and can be done by iterating to minimise equation (2). This then provides a more accurate set of ICPs from which to estimate the pose of the checkerboard.

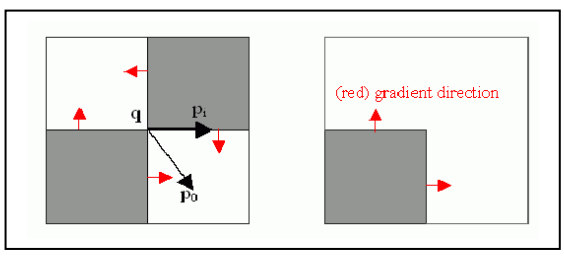

Figure 3: Sub-pixel accuracy with q-p orthogonal to gradient [5]

$$\epsilon_i = DI_{p_i}^{T} \cdot (q - p_i) \quad (4)\ [17]$$

E. Pose Estimation

Standard checkerboards are commonly avoided for use in pose estimation due to their symmetry resulting in rotational ambiguity about the Z-axis (towards camera). However this can be avoided with the proposed method if a checkerboard with one odd-dimension and one even-dimension is chosen; e.g. board size = 6x7.

Estimation of the pose of the checkerboard in 3D by finding extrinsic parameters is done using the refined ICPs, the real-world object points, the camera intrinsic parameters and distortion coefficients. The pose is found such that the reprojection error is minimised, i.e. the sum of squared distances between the observed ICPs and the object points projected into the image plane.

F. Mid-point Identification

The mid-point of the checkerboard is taken to be the intersection point of two lines diagonally crossing the board. Each line passes through two ICPs at opposite corners of the board as shown in Fig. 2 ICP locations are already known in image coordinates from the sub-pixel corner location step. The order of ICPs follows the order of initial corner point identification in B. in which the algorithm locates points moving each column. Therefore each line joins points as shown in equations (3) and (4) where the first point has index 0 and the last point has index *total_number_of_points-1*.

Line1: *point(first)* → *point(last)* (3)
Line2: *point(first + board_width -1)* →
point(last - board_width +1) (4)

The point where Line1 and Line2 intersect is the mid-point of the board. This method is advantageous because it finds the true mid-point rather than the position of the pixel in the middle of the board image. Therefore this method is tolerant to rotations about X, Y and Z and provides a logical single tracking point target useful for navigation.

Most of the method mentioned above relies heavily on, and is implemented by the OpenCV library [5]. The only completely new addition to the method introduced in this paper was the computation of the true centre of the checkerboard.

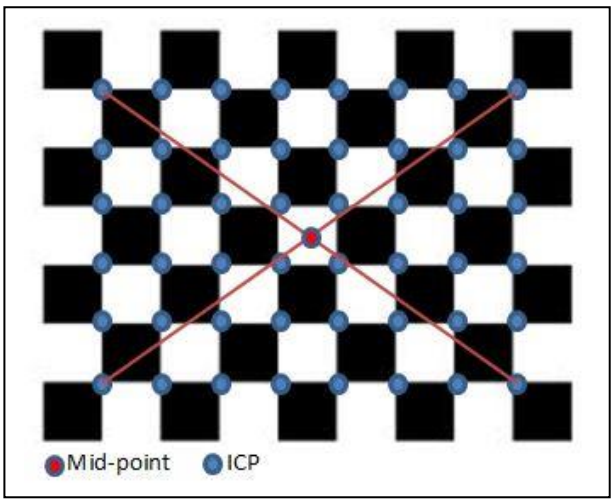

Figure 4: Mid-point Identification

IV. Results

A. Recognition and error

The proposed method can be seen in operation in Fig. 5 – Fig. 9. These show the rotation (in degrees), translation (in centimetres and mid-point of an A4 size checkerboard as seen from various viewpoints. The Z axis translation in this case is relative to a pre-set real-object value of 0.5m. This means that the extrinsic parameters were computed with the assumption that the real checkerboard reference was 0.5m away from the camera.

It can be seen in Fig. 6 that as the distance away from the camera is increased, the accuracy of the midpoint decreases. At 1.2m the midpoint was measured to be accurate to within 0.013m in the X-Y plane (50% of square error). Fig. 7 and 8 show two different extremes of recognizable rotation. Further rotation as shown in Fig. 9 results in loss of marker recognition. It should be noted that the maximum angle of rotation to which recognition could be held decreased as the checkerboard was moved further away from the camera. The maximum rotation angle and recognition in general appeared invariant to background lighting changes however changes in overhead lighting caused significant variance in results.

Table 1 shows the depth measurement accuracy of the method run on the two different cameras using both A3 and A4 size checkerboards as these were deemed to be the most easily printable sizes for a user. A reference depth of 0.5m from the camera was used in extrinsic parameter calculations for this data. Therefore a reported depth of 0.8m is in fact when the checkerboard is 1.3m away from the camera. A number of observations were made during this test:

- Running the algorithm with a square size greater than the real square size produced more accurate results.
- Rotation about the X and Y axis gave understandable results only when the long edge of the board was orientated with the Y axis.
- Rotation about the Z axis was always consistent.

V. Conclusion

This paper has presented a method to estimate the pose of a checkerboard in 6-DOF. This could be used as a very basic fiducial marker for single object tracking or navigation assistance. The method is able to overcome rotational ambiguity within dimensional constraints; however results are affected significantly by overhead lighting conditions and image resolution. This method provides a good platform for future work into increasing the accuracy and robustness of the algorithm. Currently, measurement of accuracy is difficult due to noise in reported rotation and translation values. This could be improved by adding filtering or averaging.

Figure 6: Far pose estimation

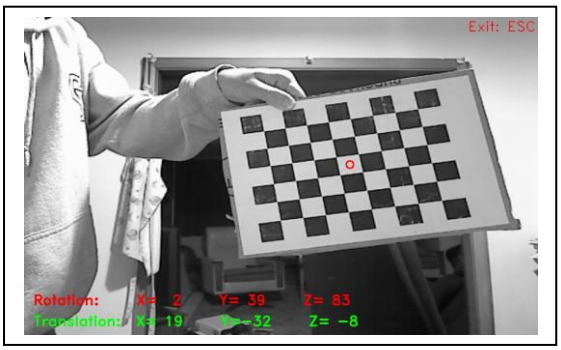

Figure 5: Near pose estimation

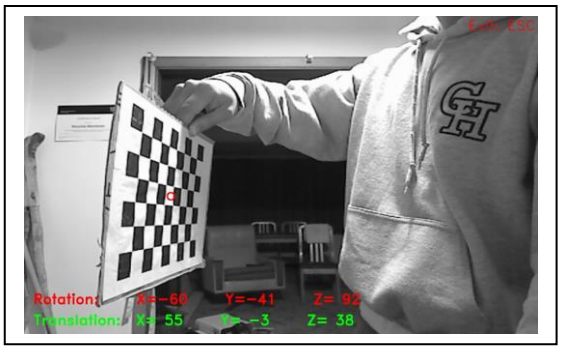

Figure 7: Y-axis rotation extreme

Table 1 Depth Recognition

Measurements for 7x8 checkerboard		**Laptop Webcam**			**2Mp Logitech Webcam**		
		Reported	*Actual*	*Error*	*Reported*	*Actual*	*Error*
Maximum distance from camera (m)	A4 0.028m squares	0.80	0.78	2.6%	0.48	0.75	36%
	A3 0.04m squares	1.44	1.28	12.5%	1.04	1.15	9.6%

REFERENCES

[1] M. Fiala, "Designing highly reliable fiducial markers," IEEE Trans. Pattern Anal. Mach. Intell, vol. 32, pp. 1317-1324, July 2010.

[2] M. Faila, "ARTag, a fiducial marker system using digital techniques," Proceedings of the IEEE Computer Society Conference on computer vision and pattern recognition. Ottawa: Canada, p. 590, 2005.

[3] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," Proceedings of the 2nd IEEE and ACM International Workshop on augmented reality. Washington DC, USA: IEEE Computer Society p. 85, 1999.

[4] B. Atcheson, F. Heide, and W. Heidrich, "CALTag: high precision fiducial markers for camera cabiration," R. Koch, A. Kolb, and C. Rezk-Salama, Eds. Proceedings of the Vision, Modeling and Visualisation Workshop 2010. Germany: Eurographics Association, p. 41, 2010.

[5] OpenCV, "Camera calibration and 3D reconstruction," OpenCV 2.4.5.0 documentation. April 2013

[6] A. Benetazzo, "Accurate measurement of six degree of freedom small-scale ship motion through analysis of one camera images," Ocean. Eng, vol. 38, pp. 1755-1762, September 2011.

[7] S.S. Srinivasa, D. Ferguson, C.J. Helfrich, D. Berenson, A, Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M.V. Weghe, "HERB: a home exploring robotic butler," Auton Robot vol. 28, pp. 5-20, November 2009.

[8] J. Köhler, A. Pagani, and D. Stricker, "Detection and identification techniques for markers used in computer vision," A. Middel, I. Scheler and H.Hagen, Eds. Germany: Dagstuhl publishing, 2010, pp. 36-44.

[9] C. Forman, M. Aksoy, J. Hornegger and R. Bammer, "Self-encoded marker for optical prospective head motion correction in MRI," Med. Image. Anal, vol. 15, pp. 708-719, 2011.

[10] Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell, vol. 22, pp. 1330-1334, November 2000.

[11] V.F.C. Neta and D.B. Mesquita, "On the design and evaluation of a precise scalable fiducial marker framework," 23rd SIGBRAPI-Conference on graphics, patterns and images. Gramado: Brazil, p. 216, 2010.

[12] Z. Wang, W. Wu, X, Xu and D, Xue, "Recognition and location of the internal corners of planar checkerboard calibration pattern image," Appl. Math. Comput, vol. 185, pp. 894-906, 2007.

[13] M. Fiala and C. Shu, "Self-identifying patterns for plane-based camera calibration," Mach. Vision. Appl, vol. 19, pp. 209-216, 2008.

[14] A. Ramey and M.A. Salichs, "Playzones: a robust detectors of game boards for playing visual games with robots," III Workshop de Robótica : Robótica Experimental. Spain: Seville, 2011.

[15] H. Kubota, M. Takeshi and H. Saito, "3D Head pose tracking using a particle filter with nose template matching," Electron. Comm. JPN 1, vol. 94, pp. 34-42, 2011. [Denki Gakki Ronbunshi, vol. 128-C, pp. 1447-1454, September 2008].

[16] S. Ekvall, D. Kragic, and F. Hoffman, "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling" Image. Vision. Comput, vol. 23, pp. 943-955, May 2005.

[17] C.L. Liu, "Three-dimensional hand tracking and surface-geometry measurement for a robot-vision system," Waterloo: Ontario, 2008.

Figure 8: X-axis rotation extreme

Figure 9: Undetectable rotation