Determining Shot Accuracy of a Robotic Pool System

Lam, J.; Long, F.; Roth, Gerhard; Greenspan, M.

Third Canadian Conference on Computer and Robot Vision [Proceedings], 2006

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Determining Shot Accuracy of a Robotic Pool System *

Lam, J., Long, F., Roth, G., and Greenspan, M.
June 2006

* published at the Third Canadian Conference on Computer and Robot Vision. Québec City, Québec, Canada. June 7-9, 2006. NRC 48490.

Canada

Determining Shot Accuracy of a Robotic Pool System

Joseph Lam[1] Fei Long[1] Gerhard Roth[2] Michael Greenspan[1,3,4]

[1]Dept. Electrical & Computer Engineering, Queen's University, Kingston, Canada
[2]Institute for Information Technology, National Research Council of Canada
[3]School of Computing, Queen's University, Kingston, Canada
[4]corresponding author: michael.greenspan@queensu.ca

Abstract

A process is described to determine the shot accuracy of an automatic robotic pool playing system. The system comprises a ceiling-mounted gantry robot, a special purpose cue end-effector, a ceiling-mounted camera, and a standard bar pool table.

Two methods are compared for extracting the homography between the camera and the table plane. A challenge was the relatively large area of the table surface, which required a similarly large chessboard pattern to determine the point homography. In contrast, the Dual Conic method required only a set of orthogonal lines as a calibration target, which was more convenient to manipulate, and could be calculated from the integration of multiple images with multiple target locations. The Dual Conic method was shown experimentally to recover the homography with a similar, and sometimes greater accuracy than the Chessboard method.

An experimental procedure was devised to measure the accuracy of an automatic shot using a sequence of images acquired from the overhead camera. For a set of 10 shots, the average absolute angular error in placing a shot was determined to be 0.74^0*, with a standard deviation of* 0.96^0*.*

Keywords: vision-guided robotics, calibration, homography

1 Introduction

Robotic Gaming Systems are of growing interest to the research community, owing both to the intrinsic appeal of the games and sports that they address, and also to the technical challenges that they pose. Games such as soccer [1], table soccer [13], and table tennis have all been the subject of robotic gaming research, the common objective being not only to play, but to compete successfully against humans.

Since the early 1990s, there have been a number of efforts to develop robotic systems to play cue sports such as pool, billiards, snooker, carom, etc. [12, 4, 5, 8, 2, 11]. In many ways, pool is an ideal game for robotic play. It requires a great deal of positional accuracy, at which machines outperform humans. Pool also has a significant strategic component, similar to chess, at which machines have already proven superior to humans. There is every reason to believe that a robotic pool system can be developed to compete against a proficient human opponent.

In this paper, we describe such a system that is under development. The main hardware elements, illustrated in Fig. 1, are: a ceiling-mounted gantry robot, a cue end-effector, a ceiling-mounted camera, and a standard 4'x8' pool table. The gantry has 6 degrees-of-freedom (*DOFs*), so that the cue ball can be struck at any angle. The end-effector is controlled indedpendently of the gantry, and has an additional DOF. While there have been other configurations proposed to play robotic pool, a gantry has a number of advantages and is the platform that the research community is converging upon [12, 4, 2, 11]

This paper describes a framework for sensing the ball positions on the table and accurately placing a shot. The paper continues in Sec. 2 with a description of the system and the coordinate frame transformations required to place a shot. In Sec. 3 two methods are empirically compared for computing the homography between the camera and table coordinate frames. An experiment is presented in Sec. 4 for calculating the accuracy of a shot, and the paper concludes in Sec. 5 with a discussion of how to improve upon shot accuracy.

2 System Description

Pool is a sport that requires accuracy, both to sink the current object ball, and to predict and control the arrangement of the balls following a shot, in or-

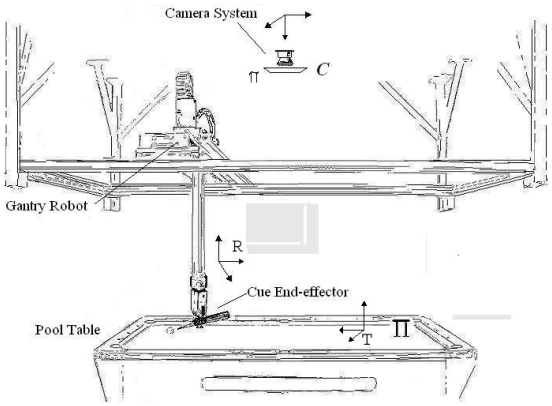

Figure 1: Hardware and Coordinate Frames

der to facilitate future shots. Accurate shooting depends upon accurately sensing the ball locations with the camera, accurate positioning of the cue via the gantry in 5 DOFs, and accurate control of the cue speed when the shot is placed.

In the experimental platform illustrated in Fig. 1, the end-effector has been custom designed to be very accurate at controlling cue velocity. The gantry, however, is a standard commercial system, and like all such systems tends to be highly precise and repeatable, but has low absolute positioning accuracy, on the order of centimeters. It is possible to design a gantry robot that has fine-grain accuracy ($\sim 15\mu m$) over the desired workplace. For example, Coordinate Measurement Machines (CMMs) have such accuracy over similar working volumes. This high degree of accuracy comes at a cost, however, and such a device would be expensive, delicate, and unlikely to maintain accuracy while absorbing the impacts required when placing shots. A more reasonable approach is to demand less absolute accuracy from the primary positioning device, and to rely upon a vision system for calibration and correction.

There are three coordinate frames that we consider, as illustrated in Fig. 1: the camera frame C, the table frame T, and the robot frame R. C is defined by the retinal plane π of the camera, which spans the $x-y$ axes of C. T is defined by the playing surface Π of the table, which spans the $x-y$ axes of T. The z-axis of the robot frame R is parallel to the vertical rail of the gantry. In general, these frames will not be perfectly aligned, i.e., π and Π will not be exactly parallel, and T_z will not be perfectly perpendicular to π or Π. It is therefore necessary to calibrate the system to determine the transformations between C, T, and R. It is possible for the robot to perform shots without a consideration of the T frame, using, for example, visual-servoing techniques. However, for advanced play (which is the goal of this system) an accurate knowledge of the table state is essential to plan shots strategically. Recovery of the T frame is therefore a required capability of the system.

In previous work [11] we have presented a technique to relate R and T through C. In particular, a method was described to calibrate R so that the gantry can be accurately positioned with respect to balls sensed within π that lie on Π, with an average positioning error of 0.6 mm, and a standard deviation of 0.3 mm. In this paper, we compare two techniques for determining the homography between C and T so that the balls can be accurately located within the table frame T, and we measure the resulting shot accuracy.

3 Metric Rectification

The purpose of image rectification is to undo perspective distortions and recover the metric properties of the scene. Rectification allows metric properties from the world plane, such as angles and length ratios, to be measured directly from a perspective image. For example, the locations of the balls sensed within the camera frame C can be accurately recovered within the table frame T, once the image plane π has been rectified with respect to table plane Π.

3.1 Projective Transformation

A 2D projective transformation, known as a *homography* [7], is a mapping between two planes. Such a mapping preserves collinearity so that any line will be mapped to another line. It can be expressed as a non-singular 3×3 matrix H in homogeneous coordinates, so that for any point $\vec{x}$ in Π there exists a unique corresponding point $\vec{x'}$ in π, which is expressed as:

$$\vec{x'} = H\vec{x} \tag{1}$$

It has been shown [9, p 42] that H can be uniquely decomposed into $H_S H_A H_P$ as follows:

$$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ V^T & v \end{bmatrix} = \begin{bmatrix} A & t \\ V^T & v \end{bmatrix} \tag{2}$$

Such a decomposition represents a hierarchy in the projective transformation group, as each successive transformation is the next higher level in the hierarchy than its predecessor, with H_S, H_A, and H_P representing similarity, affine, and projective transformations, respectively. Matrix $A = sRK + tV^T$

is a non-singular matrix and K is an upper triangular matrix with $\det(K) = 1$. There are a variety of decomposition techniques can be applied to achieve such a decomposition from a single projective transformation matrix, such as Singular Value Decomposition. Similarly, affine and projective effects can all be separated in this format.

3.2 Two Rectification Methods

Two methods are described in this paper to recover the geometric relationship directly from the image plane without any prior knowledge of the camera properties. This requires estimating the homography that undoes the perspective effect, and is typically achieved by placing objects with a known geometry in the image plane, and using this known geometry to compute the rectifying homography.

3.2.1 Chessboard Method

The most common approach is to use a chessboard pattern, and then to establish point correspondences between the chessboard pattern in the image plane, and the pattern in it's frontal position. The idea is to establish point correspondences between the corners extracted from the two images with the known corner locations in the pattern, and then to estimate the projective matrix in Eq. 1 from these correspondences. Within an image of this pattern, each corner of a square can be detected using a subpixel corner finder[6]. In this way, for each square corner, there will be a correspondence between $\vec{x'}$ (the image point expressed in 2D image coordinates) and $\vec{x}$ (the world point defined by the physical pattern in 2D chessboard coordinates).

For the homography H, there are nine elements, but only the ratios of the nine elements are significant. There are therefore only eight degrees of freedom in this matrix. Each correspondence of $2D$ points generates two constraints for Eq. 1. If the number of correspondences is $n=4$, then an exact solution can be obtained. In practise, there exists noise and uncertainty from the extracted feature points. More correspondence points are normally used to achieve a better estimate. Typically, RANSAC [3] is used on to estimate a candidate homography and identify inliers, and a refined homography for all inliers is then computed using a suitable linear minimization scheme.

3.2.2 Dual Conic Method

Another approach to rectification is based on the observation described in the previous section that there is a natural hierarchical decomposition of H. This implies that it is not necessary to rectify the image by a full projective transformation as is done in the chessboard method. Alternatively, we can rectify the image at different levels and combine the results together to achieve a full rectification.

The Dual Conic method [9] uses the fact that a projective transformation up to the affine level can be determined by decomposing the transformed Dual Conic of the circular points at infinity, denoted as $C_\infty^{*'}$. A property of $C_\infty^{*'}$ is that it is fixed under the projective transformation H if and only if H is a similarity transformation. $C_\infty^{*'}$ therefore encapsulates the similarity transformation that remains unknown. In this way, if we define $U = H_P H_A$, then a further decomposition can be written as :

$$C_\infty^{*'} = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \tag{3}$$

The central degenerate matrix is the value of C_∞^* expressed in its canonical form, i.e., within a Euclidean coordinate frame, and Eq. 3 transforms C_∞^* under projective transformation U. The above decomposition can be established by Singular Value Decomposition, i.e., $UDV = SVD(C_\infty^{*'})$ where $H = U$. The rectifying projective matrix can then be defined up to an affine level as $H^{-1}{=}U^{-1}$.

Under a projective transformation, the relationship between an orthogonal line pair and $C_\infty^{*'}$ remains unaffected, even though the line pair may not appear orthogonal after the transformation. If lines $L{=}(l_1, l_2, l_3)$ and $M{=}(m_1, m_2, m_3)$ are orthogonal, then in the transformed space the orthogonality relation $L^{T'} C_\infty^{*'} M' {=} 0$ still holds. This can be written in matrix representation as:

$$\begin{pmatrix} l'_1 & l'_2 & l'_3 \end{pmatrix} \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{pmatrix} m'_1 \\ m'_2 \\ m'_3 \end{pmatrix} = 0 \tag{4}$$

and then expanded as a 6 parameter constraint:

$$\begin{aligned}(l'_1 m'_1, 0.5(l'_1 m'_2 + l'_2 m'_1), 0.5(l'_1 m'_3 + l'_3 m'_1), \\ 0.5(l'_2 m'_3 + l'_3 m'_2), l'_3 m'_3) c = 0\end{aligned} \tag{5}$$

$C_\infty^{*'}$ can be expressed in coefficient matrix form as a normal conic $C = \{\ a \quad b \quad c \quad d \quad e \quad f\ \}'$. It is symmetric, with five degree of freedoms, and each orthogonal line pair will give a constraint for $C_\infty^{*'}$. A minimum of five such pairs will therefore build up a general linear equation system $AC = 0$, which can then be solved to estimate a value for $C_\infty^{*'}$. This implies that in a projective transformed space (i.e.

an image plane) we can solve $C_\infty^{*'}$ by a minimum of five transformed orthogonal line pairs.

The central idea of the Dual Conic method for computing a homography is to detect orthogonal line pairs and thereby estimate $C_\infty^{*'}$. This approach has three potential benefits over the standard Chessboard method, the first of which is that many standard consumer objects, such as sheets of cardboard and carpenter's squares contain accurate orthogonal lines, and so calibration objects are easy to find. In contrast, very flat and dimensionally accurate chessboard patterns are not readily available, and can be difficult to maintain, especially for large patterns.

The second benefit is that in the Chessboard method, a single image is acquired of the pattern at a single location, so that the object coordinate frame can be established. Alternately, in the Dual Conic method, the target can be placed at multiple locations over the scene plane, and the results from each location can then be integrated into a single recovered homography. Alternately, it is also possible to use many orthogonal line targets placed at different locations within a single image. This is especially beneficial if the scene plane is large, as in our case, whereby a large chessboard pattern would be required to cover the plane and recover the homography with a suitable accuracy. A minimum of five such detected line pairs will provide an exact solution for $C_\infty^{*'}$. In practise, due to image noise and the limitations of any feature detection algorithm, more data is desirable.

The final benefit is that the Dual Conic method has the potential to be more accurate than the Chessboard method, as demonstrated in the following section.

Once H has been recovered, it can then be further scaled from the affine to the metric level by recovering a simple scale factor [10].

3.3 Rectification Results

In this experiment, the accuracies of the two rectification methods are compared. The camera used was a color Point Grey Dragonfly with 1024×768 pixels fitted with a $\sim$ 6.5 mm focal length lense. The distance from the table surface to the camera was $\sim$ 2.35 m, and the table dimension was 2.032 m $\times$1.016 m. Let C be the camera frame in camera pixels units and T be the table frame in metric units, i.e. mm. The objective was to estimate the position of targets of known location in T by rectifying images of the target in C using both methods, and to compare the accuracy of the estimates. The target was a flat pattern, the chessboard pattern, which lay on the table surface parallel to Π. The mapping from T to C is a homography between the pool table plane Π and the camera image plane π, i.e., $H : (x, y) \in T \longmapsto (u, v) \in C$.

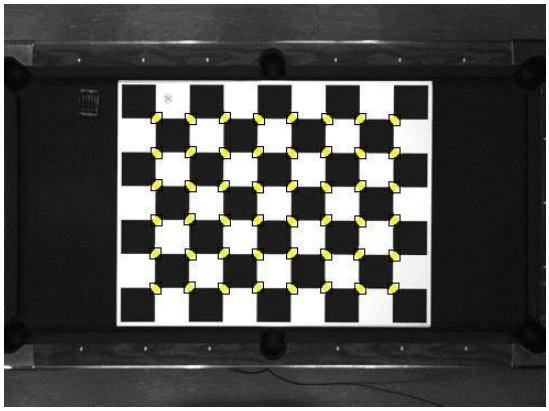

Figure 2: Chessboard with 48 Detected Corners.

Due to the height of the ceiling, it was necessary to use a small focal length ($\sim$ 6.5mm) lense to image the complete surface of the table. Such wide angle lenses typically suffer from appreciable radial distortion, which had to be corrected prior to image rectification. We employed a standard radial distortion correction routine based on the camera calibration algorithm of Zhang [14] using a planar chessboard pattern. The routine provided a good estimate for all of the camera's instrinsic parameters, including the radial distortion factors.

The Chessboard method requires the construction of a flat chessboard pattern. To achieve a good estimate, the chessboard should cover the entire area of interest and provide as many feature points as possible with a clear square pattern. We designed a 1.38m $\times$1.09m chessboard with 7×9 squares printed by a commercial large scale printing device. It covered about 2/3 of the table surface. Each square was 150 mm $\times$150 mm and it provided 48 inner corners as feature points. The printing accuracy was 600 dpi (dots per inch). The mechanical accuracy of the printing device was specified as less than 0.1 mm, which we were able to verify manually. The pattern was mounted on a foam board which was flat, rigid, light and easily repositioned. Fig. 2 shows an image where all 48 corners where correctly and accurately detected.

In contrast to the Chessboard method, the Dual Conic method required only a simple calibration object that consisted of one orthogonal line pair. We used a $\sim$ 600 mm $\times$600 mm flat wooden square. Such objects are mass produced using high accu-

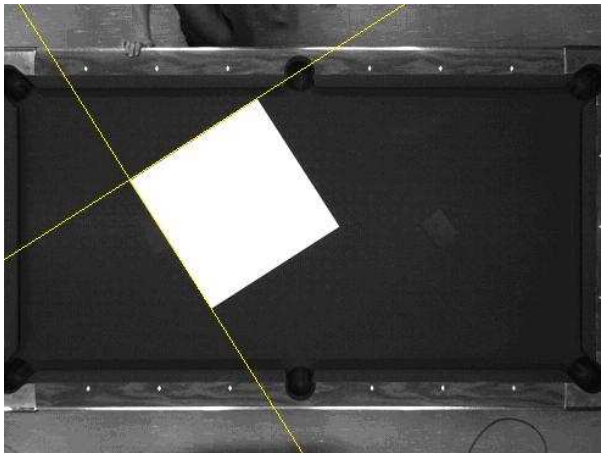

Figure 3: Square With Detected Orthogonal Lines.

racy industrial cutting processes. Each pair of adjacent edges provided a pair of orthogonal lines. The size of the square was chosen so that each edge was long enough to provide a good estimate of a line and also the whole square was small enough to be moved around the table conveniently. For each location, an image was acquired, radial distortion was corrected, and two orthogonal lines were extracted, as in Fig. 3. The line extraction was achieved using a RANSAC method, followed by a least squares estimation of the inliers.

In each calibration trial, about 50 such orthogonal line pairs were extracted by placing the square pattern at different random locations on the table surface. We then estimated $C_\infty^{*'}$ in the image space using the Direct Linear Transformation (*DLT*) linear optimization algorithm [7]. The DLT optimization method minimizes algebraic distance, and requires input data normalization as an important preprocessing step. Both the point data, in the case of the Chessboard method, and the line data, in the case of the Dual Conic method, where normalized prior to DLT minimization.

3.3.1 Comparison of the Two Methods

The goal was to determine which of the two methods rectified more accuractely. The square pattern was randomly placed on the pool table surface, and for each location, an image was acquired and an orthogonal line pair was extracted. After applying both rectifying matrices on each line pair, the resulting line angles where measured. Perspective distortion does not preserve angles between lines, so that any variation from the true 90^0 separation of the lines in the rectified images was due to rectification inaccuracy. The angle error was the absolute difference between the resulting measured angle and the true 90^0 angle. This angle error was also computed prior to rectification as the initial error.

The results were compared based on 50 line pairs for both methods, and are tabulated in Table 1. The maximum, minimum, mean and standard deviation of 50 line pairs' angle errors are shown for both methods.

It can be seen that the unrectified absolute mean angle error for all 50 line pairs was about 0.14^0. The homography estimated from the Dual Conic method has corrected this error to 0.087^0, which was about 40% better than the initial error. The homography estimated from Chessboard method resulted in about 0.11^0 error, which was 30% better than the initial error. We found that in most cases, the Dual Conic method provided a better estimate as indicated by angle error. This was reasonable as the Dual Conic method is an estimation process based on the minimization of orthogonal angles.

Another comparison was based on the metric distances between two points on the table surface. The chessboard was used as the physical pattern to do this test, because the physical dimensions of the pattern were known precisely. The process was straightforward: we placed the chessboard on the table surface at a few different locations. For each location, an image was acquired and all corner points were extracted. The rectification was then applied to these points to compute the transformed points in the table frame, i.e., in metric units. Finally, the relative distance error for these transformed points was computed based on their known locations in the pattern.

In this test, the maximum error from both methods was less than 3 mm and the mean error was less than 1 mm. Both methods performed similarly in the scale and distribution of the error when the testing target was extracted from different locations on the table surface. The maximum error was 2.79 mm over a feature separation distance of 1000 mm, with the homography computed from the Chessboard method on the lower right corner of table. In all three positions, the Dual Conic method produced a smaller average and maximum error for most of the measured distances. Overall, the Dual Conic method gave a smaller or similar scale of maximum and average error than the Chessboard method.

An example of the distribution of the errors from this test is illustrated in Fig. 4. Here the magnitude of the error with respect to the center of the pattern is illustrated as a color-coded relief map of the pattern surface. The errors for both the Dual Conic and Chessboard methods are shown.

	Angle Error (degrees)			
	Max	Min	Mean	Std. dev.
Unrectified	0.406	0.001	0.142	0.093
Dual Conic	0.500	0.000	0.087	0.085
Chessboard	0.500	0.000	0.109	0.091

Table 1: Line Angle Error Comparison

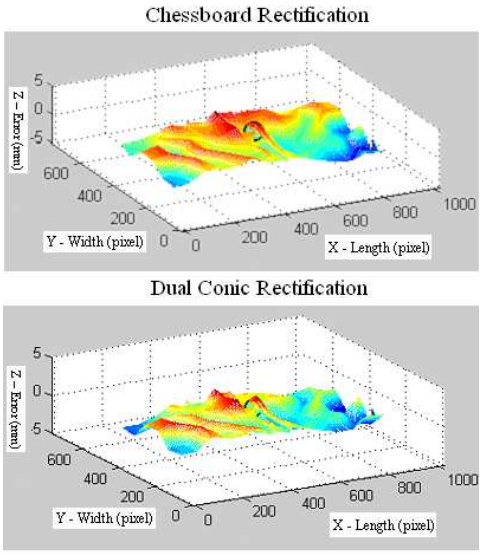

Figure 4: Table Surface Error Mapping

3.3.2 Discussion

The Chessboard method is the most straightforward and well known rectification method. Its popularly is due to its simplicity and ease of the calibration process. The Dual Conic method is a more flexible and reliable method for this applicaton because it demands less of the physical pattern. It is more suitable for a large object plane and provides a more accurate metric angle result and a similar relative distance result to the Chessboard method. The remaining errors of the vision system are due to the non-linear, non-perspective errors from the individual cameras and lenses, and are not accounted for by the rectification process and must be accomodated for either method by further processing.

4 Shot Accuracy

The computed transformations were applied to determine the cue position when placing a shot, and the resulting shot accuracies were measured using the vision system. In addition to the homography estimation errors, there were a number of other sources of error that affect shot accuracy, such as: camera calibration errors, camera-to-robot frame calibration errors, robot positioning error, image noise, shadows, etc. This experiment evaluated the cumulative effect of all errors when placing a shot.

4.1 Experimental Design

The experimental design was to locate the position of the cue ball on the table using the vision system, and place a shot toward a specified target location. The trajectory of the cue ball was extracted from a series of images that were acquired at ~ 40 msec. intervals following the shot, and the shot error was calculated as the inscribed angle between the line defined by the initial ball location and the target location, and the extracted ball trajectory.

The cue ball location is transformed by the homography from the camera to the table frame. The position of the robot to place the shot is first calculated in the table frame, and is then transformed to the robot frame. The sequence of computations in this experiment is similar to those required to place a shot in a game scenario, and the resulting acuracies are therefore indicative of those that would be achieved in a game. The shot sequence can be outlined in five major steps:

1. ***User provides a target point.*** In the first step, a target point is provided near the foot of the table, specified in table coordinates. It may seem that it would be easier to select a point directly in image coordinates. By doing so, however, testing the homography would be omitted, since the robot coordinate could be directly computed from the correspondences obtained by the camera-to-robot coordinate calibration [11], thereby bypassing the camera-to-table homography transformation that is required in a game scenario.

2. ***Cue ball placed in front of target point.*** The cue ball is placed near the head of the table, about 4 diamonds from the target point, far enough so that the camera can observe the ball trajectory. The ball location is accurately extracted from the image using a sequence of background subtraction, edge detection, RANSAC circle extraction, and least square circle best fit of the inliers.

3. ***Robot moves to striking position.*** To move the robot to its striking position, its location and striking angle within the robot coordinate frame are calculated based on the extracted cue ball image coordinate and the target point in table coordinates. This is found by locating a point behind the line that connects the cue ball and target locations in table coordinates. The robot coordinates are computed by applying the transformation between the table and

robot coordinate frame.

4. *Robot shoots cue ball to target point.* The strike velocity was selected as 0.75 m/s. This speed was deemed to be fast enough so that the ball's trajectory is largely unaffected by the surface texture of the table. It was also slow enough that a number (~ 5) of images of the ball could be acquired at the maximum camera acquisition rate, prior to its impact with the target point, and so that motion blur was minimized yeilding largely circular images of the ball.

5. *Cue ball trajectory computed.* Prior to shooting, the robot is positioned to its calculated shot location, and a set of 30 background images are acquired. The robot is then commanded to shoot, and a series of images are acquired at the maximum camera acquisition rate. Each image is subtracted from the background images by comparing the mean and standard deviations of each pixel, and the ball is extracted as a circle in each image. The circle centers from each image are then used to best fit a straight line, which represents the ball trajectory. As the shot is commanded with a level cue, the resulting ball trajectory will be rectilinear except for small deviations caused by the texture of the table surface.

The above steps were repeated for 10 different targets and cue ball locations, and the shooting accuracy was estimated for each shot by calculating the angular error between the desired and actual trajectories. The results from one shot are illustrated in Fig. 5. The white line is the desired trajectory which connects the initial ball and the target locations. The green line indicates the actual trajectory, with the four circles representing the ball locations extracted from each image. The angle between the two lines is a measure of shooting error. The results from these trials are tabulated in the *test 1* column of Table 2.

The perspective warping that exists within the images does not preserve angles between lines, so the ball locations were transformed from image to table coordinates, and the trajectories computed in table coordinates. This resulted in a more accurate expression of shot error than if the trajectories were expressed in image coordinates, which would have exhibited a bias due to perspective warping.

4.2 Improving Shot Accuracy

Another goal of this experiment was to attempt to improve the shot accuracy by introducing a linear correction term. The assumption was that the errors exhibited in the above shot experiment where due to a systemic bias, that could therefore be fed back to the positioning system to improve accuracy.

shot	*test 1*	*test 2*	*test 3*
1	0.61	0.53	-0.87
2	0.34	-0.62	0.68
3	-1.07	-0.66	-1.46
4	1.16	-0.36	-1.47
5	-0.81	0.52	-1.66
6	0.08	-2.15	-0.19
7	0.05	-0.47	-1.88
8	2.37	0.71	-1.69
9	0.42	-0.54	0.98
10	0.48	1.07	-1.36
Mean	0.36	-0.20	-0.89
|Mean|	0.739	0.763	1.224
Stdev	0.96	0.94	1.03

Table 2: Shot Accuracy Results (degrees)

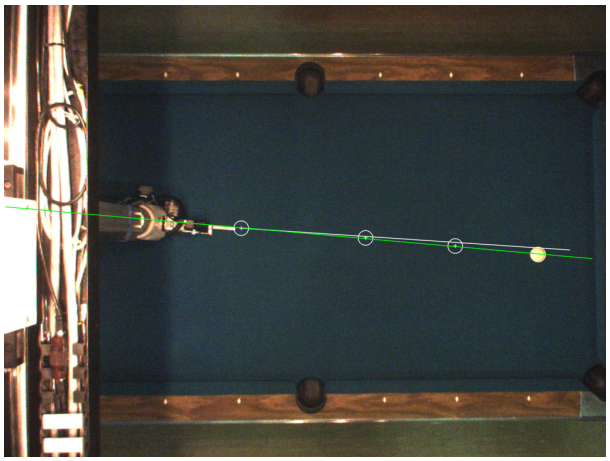

Figure 5: Ideal (white) and actual (green) trajectory.

The average angular error from the first 10 trials was calculated as 0.364^0, and this value was used to correct the robot shooting angle. An additional 10 shot trials were then performed using this correction term, and the angular errors where calculated.

The result from this set of trials is tabulated in the *test 2* column of Table 2. It can be seen that by introducing a correctional term using the calculated mean from the first trial, the signed mean angle error was reduced. The mean angular error for the corrected shot trials was -0.20^0, which is nearly half (56%) of the magnitude of the initial pre-corrected mean trial error of 0.36^0. However, such a simple linear correction is only useful if the goal is to average out the performance of every shot taken. The absolute mean showed that if performance was based on placing a shot closer to the unbiased 0^0 angle, then the angle errors in fact increased slightly with

the addition of the linear correction term, from 0.74^0 in test 1 to 0.76^0 in test 2. The standard deviations for the two trials were nearly identical.

This procedure was repeated, with the average error for the second set of trials fed as a linear correction term for an additional set of 10 trials, tabulated in the *test 3* column of Table 2. In this case, both the mean and absolute mean errors increased from the previous two trials. The conclusion that we draw is that no simple additive bias exists, and that a linear correction term is not effective at improving shot accuracy.

5 Conclusion

We have compared two methods for determining the homography between the table and camera planes. The main challenge was the relatively large area of the table surface, which required a similarly large chessboard pattern to determine the point homography. In contrast, the Dual Conic method required only a set of orthogonal lines as a calibration target, which was more convenient to manipulate. It could also be placed repeatedly at various positions on the table, with the information from multiple images integrated into a single recovered homography. Our experimentation showed that the Dual Conic method recovered the homography with a similar and sometimes greater accuracy than the Chessboard homography, as measured by both angular and point reconstruction errors.

The table to image frame transformation described by the homography was combined with the image to robot frame transformation to execute shots. An experimental procedure was described to measure the accuracy of a shot using images acquired from the overhead camera. The ball location was automatically determined, and the pipeline of transformations was the same as those required to place object ball shots, so the accuracy measure is indicative of the accuracy that the system can achieve in an actual game.

The mean absolute angular error in placing a shot was 0.74^0. This is not as accurate as a skilled human, and so it will be necessary to improve the accuracy of the system to achieve a level of competitive play against a proficient human opponent. Aside from replacing the hardware with more accurate (and more expensive) components, there are two approaches to improving the accuracy of the system. One is to identify the sources of error through a process of calibrating the robot. As there are many possible sources of systemic bias in the sensing and positioning devices, this would be a complex task. The second approach is to add more and higher-accuracy cameras to the system, and use visual feedback to sense and correct positioning errors when they occur.

References

[1] http://www.robocup.org.

[2] M. E. Alian, S. Shouraki, M. Shalmani, P. Karimian, and P. Sabzmeydani. Roboshark: A gantry pool player robot. In *ISR 2004:* 35^{th} *Intl. Sym. Rob.*, 2004.

[3] R. C. Bolles and M. A. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *Seventh International Joint Conference on Artificial Intelligence*, pages 637–643, Vancouver, British Colombia, Canada, 1981.

[4] S. Chua, E. Wong, A. W. Tan, and V. Koo. Decision algorithm for pool using fuzzy system. In *iCAiET 2002: Intl. Conf. AI in Eng. & Tech.*, pages 370–375, June 2002.

[5] H. Denman, N. Rea, and A. Kokaram. Content-based analysis for video from snooker broadcasts. *Computer Vision and Image Understanding*, 92(2/3):176–195, Nov./Dec/ 2003.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[7] R. Hartley and A. Zisserman, editors. *Multiple View Geometry in Compuer Vision.* Cambridge University Press, 2000.

[8] L. Larsen, M. Jensen, and W. Vodzi. Multi modal user interaction in an automatic pool trainer. In *ICMI 2002:* 4^{th} *IEEE Intl. Conf. Multimodal Interfaces*, pages 361–366, Oct. 2002.

[9] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *CVPR*, pages 482–488, 1998.

[10] F. Long. Two methods for metric rectification over large planes. Master's thesis, Queens University, Kingston, Ontario, Canada, 2005.

[11] F. Long, J. Herland, M.-C. Tessier, D. Naulls, A. Roth, G. Roth, and M. Greenspan. Robotic pool: An experiment in automatic potting. In *IROS 2004: IEEE/RSJ Intl. Conf. Intell. Rob. Sys.*, pages 361–366, 2004.

[12] S. W. Shu. *Automating Skills Using a Robot Snooker Player.* PhD thesis, Bristol University, 1994.

[13] T. Weigel, K. Rechert, and B. Nebel. Behaviour recognition and opponent modeling for adaptive table soccer playing. In *Proc.* 28^{th} *Ann. German Con. AI*, pages 335–350, 2005.

[14] Z. Zhang. Flexible calibration by viewing a plane from unknown orientations. *ICCV'99: Int. Conf. Comp. Vis.*, pages 666–673, Sept. 1999.