

Pool Cue Guide

Determination of Guide Vectors Under Adverse Lighting, View Aspect and Scale

Shaun Weatherford

Department of Electrical Engineering
Stanford University
Stanford, United States
sweather@stanford.edu

Abstract—This paper describes and demonstrates an algorithm which determines successful cue ball impact direction vectors for a standard striped ball versus solid ball pool game, known as the Eight-ball variant of pool. From an input picture of a pool table with random cue, eight, striped and solid ball positions, the algorithm prioritizes view aspect, lighting illumination, light aspect and scale invariance. View aspect issues are reduced by rectification to a top-down view. Lighting illumination issues are reduced by white-balancing ball colors. Light direction issues are reduced by template matching. Scale issues are reduced by scaling operations to the determined table and ball size. Image processing techniques used by the algorithm include homography, image segmentation, morphological erosion, template matching, edge detection, and color processing.

Keywords—pool; cue guide; rectification; template matching; image processing; homography; image segmentation; morphological erosion; edge detection; color processing

I. INTRODUCTION

Pool, also known as pocket billiards in the United States, is played on a table with six pockets located along the wall edges (or “rails”) lining the perimeter of the table. This table typically has a ratio of length to width of approximately two, with larger tables ranging in size from 7 feet by 3.5 feet, to 9 feet by 4.5 feet. Miniature tables can be much smaller, of length as low as 2 feet or smaller. Onto this table are placed up to 16 balls, with all but the cue ball numbered and colored either a solid color, or white with a color stripe. There are 7 solid balls, and 7 stripes.

For many variants of pool, the white cue ball is hit head-on with the tip of a pool stick. The cue ball is thus directed towards impact either with another ball, or with the rail of the table with subsequent banking an impact with another ball. The goal is to impact the balls such that a chosen ball type or ball number is landed in one of the six pockets. Eight-ball is a common variant in which a player chooses either solids or stripes to land in pockets. If the player lands the opposite ball type, the opposing player wins the point. Furthermore, both players must avoid landing the eight-ball (the black ball) in a pocket. Other variants may focus on landed a particularly-numbered ball, or achieving a particular number of banks before landing. In Nine-ball, for instance, the player must impact the lowest-numbered ball on the table.

In a full-size pool game, the view aspect could be from many frontal angles, and lighting could range from lamps to natural illumination. Scale would be determined both by the distance the photographer retained to the table, and whichever image resolution the camera provided. With this in mind, a miniature table-top pool table with playing surface dimensions of 37 inches by 18.5 inches was used for this project. This size was chosen as the table was easy to move to achieve different illumination, and different view aspects (such as top-down), which could be difficult to accomplish with a particular full-size table constrained in a particular pool room environment.

The pool variant of Eight-ball was chosen, as this variant not only entails discriminating the white cue ball from other balls, but also adds the challenge of discriminating solid balls from stripes, and both from the eight ball. Thus the algorithm was developed to process the pool table picture, and then create two output pictures marking those impact (guide) vectors to the cue ball which result in landed balls: one with cue guides superimposed for stripe balls, and one for solids. To constrain the difficulty level of the project, only solutions for up to one cue ball bank and one impact ball bank were calculated. Furthermore, multiple ball impacts were not calculated.

To achieve accurate pool cue guide information in a wide variety of lighting, view aspect and scale conditions, the following image processing steps were used: rectifying the 3D image to a top-down view (using table edge detection, corner projection and homography), separating and identifying pockets and different types of balls (using image segmentation, morphological erosion and region labeling), refining localization of ball positions (using template matching), identifying inner wall “rails” (using edge detection and integrated gradient vectors), and compensating ball colors for different illumination for stripe versus solid color ball discrimination (using white balancing of ball colors from cue information).

II. PRIOR AND RELATED WORK

A. Homographic Rectification to Top-Down View

Image rectification of a pool table picture for cue guide determination has been used before. Takahashi, Kasai and Suzuki [1] rectified pool table images by estimating camera lens distortion and correcting using these estimated

coefficients. From the included rectified images it appears, however, that the rectification did not use homography in the rectification, as there is still apparent skew to the table lines. This may simply be the edge and line detection, however. Problematic to the use of this technique in the present project was the need to estimate lens distortion coefficients. The present project will take any pool table image as input, without knowledge of the lens characteristics used for the picture-taking. Therefore, a simpler technique of rectification using homography on projected corners of the image was used. The drawback to the present method to be delineated further in this paper is of course curve line distortion in the image (such as in the extreme a fish-eye view) will not be corrected.

Jebara, Eyster, Weaver, Starner, and Pentland [3] discuss a technique for future development which would estimate the pool table's 3D coordinates from multiple views of the table (multiple frames of video in this case), and then provide a rectified top-down perspective. But this technique requires multiple pictures. The present work in is limited to one picture image. Therefore, homographic rectification, rather than 3D modeling, was the technique selected.

B. Separating and Identifying Pockets and Different Types of Balls

Reference [1] divides the table into 10 x 10 pixels, then looks for the peak hue in the color component histograms. Assisting the process was the easily discernible hue of each ball color from the table cloth color, and the known 10 x 10 pixel region best used for known table dimensions. The present work allows for discriminating balls from tables of the same color, and allows for various scales of balls, making simple hue detection on a 10 x 10 region less useful. Furthermore, the use of largest hue may be problematic for stripe balls, which can exhibit peak histogram values in the white region, and it appears striped balls were not present in the pictures of the test set-up in this reference.

Ling, Li, Xu and Zhou [2] discuss techniques to discriminate green balls from a similar-hue green table cloth background. Green balls are more reflective than table cloth, therefore, gray-scale amplitude information in image segmentation thresholding can be used to identify green balls. The present paper uses a technique which works in the same manner – additional gray-scale Otsu thresholding on ball amplitude helps identify green balls missed by simple hue detection differences from background table.

Reference [3] trains a probabilistic model of the pool table and balls, which requires multiple training samples, under multiple imaging situations, to optimize. This technique was not adequate for the present paper, in that multiple pictures of the pool table are not available. Compounding this is that any pool table with any color background and any color balls is allowable in the present method.

C. Refining Localization of Ball Positions

Reference [1] uses template matching to refine ball position estimate, using an ideal ball pattern. A similar method was used in the present paper, with the difference that an ideal ball

template was unknown. A varying size template method had to be developed for varying pool table ball sizes.

Reference [3] uses a perceptual measure of symmetric enclosure instead of template matching. It requires use at an appropriate scale to consistently detect circular objects. The present paper uses a method which allows larger variance in scale, so the method in Reference [3] was not suitable.

D. Identifying Inner Wall "Rails" for Banking Shots

Reference [1] uses table edge diamond markers to determine inner rails, which requires a known and marked pool table. It appears that Reference [2] uses gradient edge detection and the Hough transform to detect pool table edge lines, but relies on the lines detected as being the inner rails, whereas in many pool table types these detected edges may instead be outer felt edges, beyond the rails. And even if rails are the exclusively detectable region, they may not be detected if the gradient is small and obscured by noise. Reference [3] again requires training data over different images of the same table to converge to a robust solution.

E. Compensating Ball Colors For Different Illumination, to Discriminate Stripes versus Solids

The included references did not specifically address discriminating stripe from solid balls under different illumination, with the exception that Reference [3] cites training models of stripes and solids with a Gaussian color model. It is not clear if this method would address illumination concerns, and whether it requires multiple pictures of the same pool table and balls.

F. Ball Trajectory and Impact Physics

References [1] includes advanced physics modeling of the balls and their impacts with each other and table rails. These models requires knowledge of specific table rail cushioning, and ball properties. The present paper allows for arbitrary table rail and ball characteristics, and therefore these properties are unknown by the algorithm. Therefore, advanced modeling of physics could not be used.

III. DESCRIPTION OF THE ALGORITHM

The following processing steps were undertaken to address short-comings as mentioned concerning prior work, as applied to the present project.

A. Homographic Rectification to Top-Down View

First, the dominant color components of the table were captured by applying a view-box around the center portion of the image, and obtaining the RGB component histogram peaks in this box. The assumption was that the center portion of the image is mostly the pool table, and the histogram peaks would represent the dominant table top color.

The table's dominant RGB color values were used in a weighting vector, which was multiplied by each pixel in the image. The result was compared to a threshold, and binarized (tabletop white, background black). The resulting binarized image was region labeled, and the region's bounding box

calculated. The bounding box vertices provided the top, bottom, left and right pixel regions over which to search for table edges. Constrained to these pixel regions, searching was executed from the edge of the picture inwards, until the white binarized table region was found. The pixel locations where table edge were found and used in a Hough transform to find the corresponding lines and their characteristic equations.

From these table edge lines, artificial table corners were projected. These corners were then input into a Homography transform, along with corner locations for a scaled top-down view of the table. The homography transform was then used to rectify the pool table image to a top-down view (see Fig. 1 for original view, Fig. 2 for rectified view).



Fig. 1. Original pool table image, before rectification.

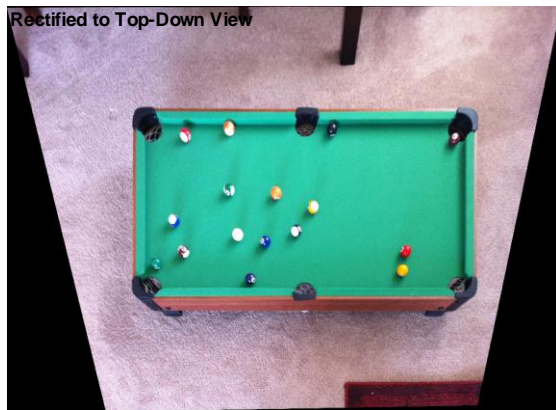


Fig. 2. Pool table image, after rectification.

The advantage to this method using homography is it does not rely on known table dimensions, but rather uses the general pool table scale of length as twice the width and applies this to scaled picture values which the transform will execute. The table and camera specifications do not have to be known or estimate, and picture scale is less of an issue.

B. Separating and Identifying Pockets and Different Types of Balls

From the rectified image of the pool table, pockets and balls were identified in the following manner.

The rectified image was binarize using the table weighting vector found for rectification, but applied to the image post-rectification. The binarize regions were region-labeled, and the table bounding box found. The bounding box was used to create a binarized filled rectangle of dimensions equal to the table dimensions, and an exclusive-or was taken of the resulting image with the original binarized table image, leaving images of the pockets and balls. This image was then morphologically eroded to isolate balls from each other and pockets (see Fig. 3 for a picture of the resulting binarized pocket and ball image, and of using erosion to separate objects). These eroded objects were then region labeled.

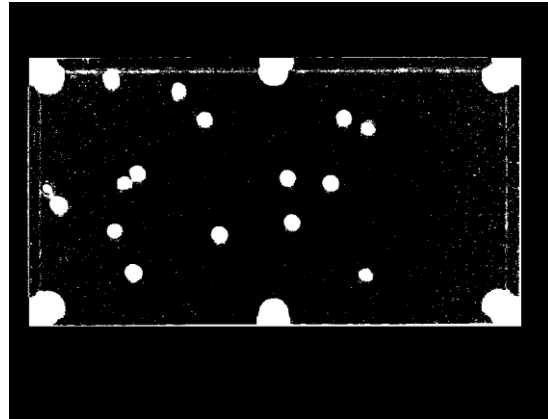


Fig. 3. Binarized pockets and balls.

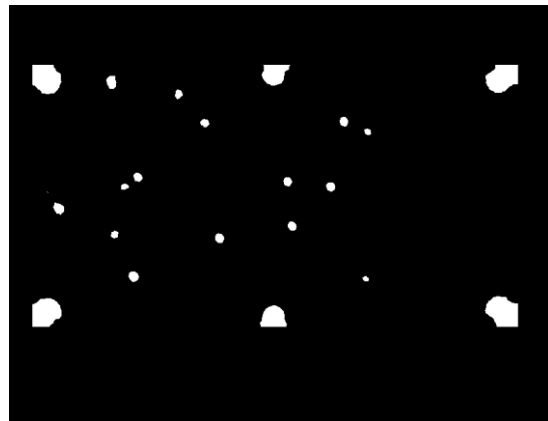


Fig. 4. Binarized pockets and balls after erosion (notice green ball middle left has nearly been removed).

Objects closest to the table bounding box vertices (and midway along the length dimension) were assigned pockets, the remainder potential ball candidates.

Using simple color weight vector thresholding to discriminate ball from background table can result in missed balls, particularly for green balls of the same color as the table. Figs. 3 and 4 show an example of a solid green ball (at middle left of figure) which was mostly missed in its green part detections, leaving only the white portions of the ball helping in discrimination. Morphological erosion nearly removed this ball from the image. Therefore, it was also necessary to add

Otsu thresholding as a technique to detect green balls. This helped detect green balls of the same hue as the table, but different in reflection or color amplitude. Fig. 5 shows binarization after adding Otsu thresholding, which improved detection of the green ball's green portions.

Noticeable in Fig. 5 is that the Otsu thresholding can add false objects. These were removed from consideration by comparing the peak object histogram amplitude to a threshold, and removing those object which did not exhibit sufficient large-amplitude reflection (similar to the method used in Reference [1]).

The objects labeled from the simple hue discrimination algorithm were considered of primary importance, as this test worked best for finely locating all balls other than green. Otsu's method coupled with hue testing added green balls missed by the hue test, but adversely influenced position estimates of balls, since ball shadows would now be included as part of an object. Therefore, balls detected by Otsu's method but missed by the simple test were added to the final ball list. For balls detected by both methods, the hue objects were used in the final ball list.

After culling ball objects, the cue and eight ball were found from RGB color component values and amplitude, with the cue containing the largest histogram values around white and the largest reflection amplitude, and the eight ball containing the lowest values.

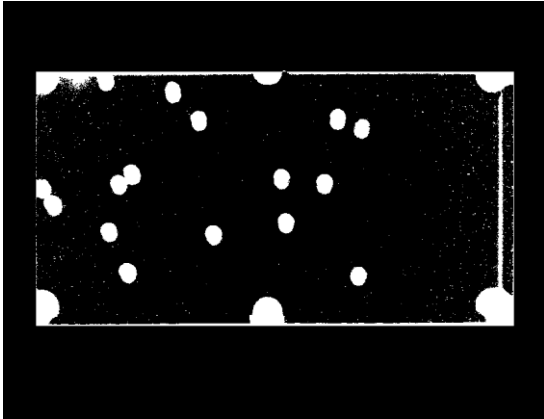


Fig. 5. Binarized pockets and balls with addition of Otsu amplitude thresholding; note improved green ball detection at left.

C. Refining Localization of Ball Positions

Similar to the method in Reference [1], a ball template was used to refine localization of each ball. Unlike Reference [1], the ball template was not an ideal ball, but rather was created by averaging all of the ball objects' binarized sub-images (a sub-image being constrained to the edges of the ball). This average was rounded, resulting in a binarized template of the average ball characteristics. This template was then AND-convolved with each ball image to find the ball position yielding the highest correlation match. This approach reduced susceptibility to scale, and was fairly fast due to the AND-convolution (an AND operation of the binarized template with the binarized ball image, rather than a multiplication).

D. Identifying Inner Wall "Rails" for Banking Shots

Inner rails were found by first using the binarized rectified image of the pool table to find outer table edge lines in the same manner used for homography. After region labeling of the binarized image, the bounding box vertices of the table were found, and used to project lines inward from the edge of the picture, to thus find top, bottom, left and right outer walls.

In addition, vectors radiating inward from each table edge pixel location were captured, and a gradient vector (central difference) for each pixel value vector was found. These gradient vectors were summed together for each vector along the length of the table. The peak in the resulting gradient vector was thus integrated over the length of the table. This process allowed detection of low-amplitude inner wall "rails", with the caveat that these rails are parallel to the edges of the table. Fig. 6 provides an example of low-amplitude gradients for the rails, for which rail locations were found and superimposed on the picture.



Fig. 6. Low-amplitude rail gradients at upper left and lower right; shadow of rail has larger gradient informatino than rail itself.



Fig. 7. Low-amplitude gradient rails found and marked.

E. Compensating Ball Colors For Different Illumination, to Discriminate Stripes versus Solids

Using the cue ball as a reference, all balls were white-balanced. The RGB weightings needed per pixel to

compensate the cue ball to white were found, and applied to each ball sub-image.

This process was necessary to discriminate white balls from solids under different lighting conditions. Discrimination was accomplished by multiplying a white RGV weighting vector by each pixel in a ball sub-image, then binarizing to a threshold. Binarized pixels were added, then divided by ball radius to determine percentage white. Balls with white percentages exceeding a threshold were assigned stripes, those not were assigned solids.

F. Ball Trajectory and Impact Physics

Rectification of the image helped in the accuracy of ball banking calculations. If implemented earlier, physics calculations could have been greatly simplified, as walls are parallel to the x and y axes of the picture, making bounce vector calculations simple. Angles alone were used in the trajectory calculations, since object material physics were not known a priori for a given table and balls (on purpose, since this algorithm is meant to accommodate different tables and balls).

G. Image Marking of Pockets, Ball Types, and Cue Guides

Finally, two output images were created. Both of these images were marked with lines indicating rails, filled boxes marking pocket centers, and rings around each ball. Stripes were marked with a broken ring, solids with a continuous ring. The cue ball was marked with a blue ring, and a blue box in the center. The eight ball was marked with a red ring and red mark in the center. All balls were white-balanced in the marked image, to provide a visual indication of the quality of the white-balancing.

The cue guide solutions for the stripes player were superimposed on the marked picture, with radial lines emanating from the cue ball in the direction at which cue ball impact would result in a landed stripe ball. Another image was similarly created for solids players. Figs. 8 and 9 below provide example pictures for the marked stripe and solid cue guide solutions.



Fig. 8. Original image.

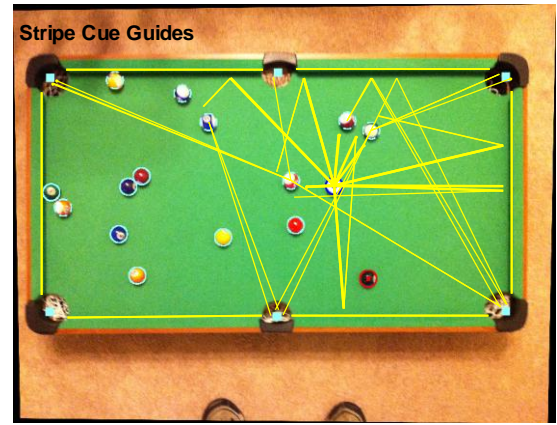


Fig. 9. Rectified and marked with stripe cue guide solutions.

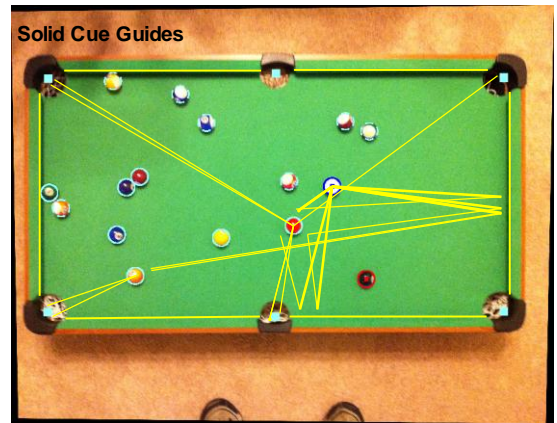


Fig. 10. Rectified and marked with solid cue guide solutions.

IV. RESULTS

The algorithm was tested under different lighting conditions, lighting angles, table view aspects, and camera image resolutions. In this manner, degree of invariance to illumination coloring, ball shadows, rotation and 3D perspective, and scaling were tested.

A. Illumination Coloring and Angle Invariance

Pictures of the reference miniature pool table were taken under two main lighting conditions: yellow (warm) lamp lighting, and natural sun lighting, as provided by a window. For both sources of lighting, different lighting angles were provided to vary shadows.

For lamp lighting, identification of all balls worked well, but the yellow stripe was misidentified as a solid. Also, position localization of the green solid was problematic, as the green ball detection algorithm is susceptible to ball shadows. Fig. 11 below shows the green ball mislocated due to its shadow.

With sun illumination, identification of balls was problematic. Fig. 12 shows an example wherein both green balls were missed. Also of interest is the homographic distortion of the upper right purple ball's shape. Fig. 13 shows another natural lighting testing, with different lighting angle

(and therefore different shadowing) which resulted in all green balls detected, but the red and orange solids misidentified as stripes.

B. Table View Aspect Invariance

Of interest is that a top-down view instead of the frontal view of under the same conditions as those for Fig. 13 in all other respects resulted in properly identified stripes versus solids. This indicates potential view aspect dependence.

C. Scale Invariance

Also of interest is that a reduced scale image similar in all other conditions to those for Fig. 13 resulted in a missed cue guide, as shown in Fig. 15. The other scale test did not indicate differences between the high and low resolution versions.

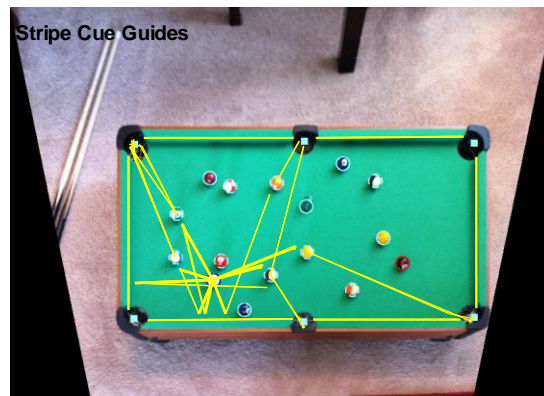


Fig. 13. Natural lighting 2. Green balls detected but red and orange solids misidentified as stripes.

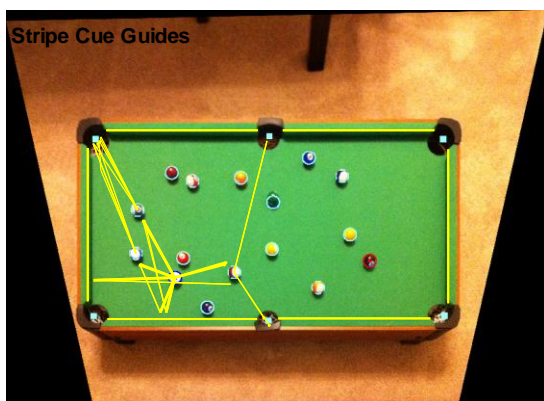


Fig. 11. Lamp lighting. Solid green ball mislocated due to its shadow, and yellow stripe misidentified as solid.

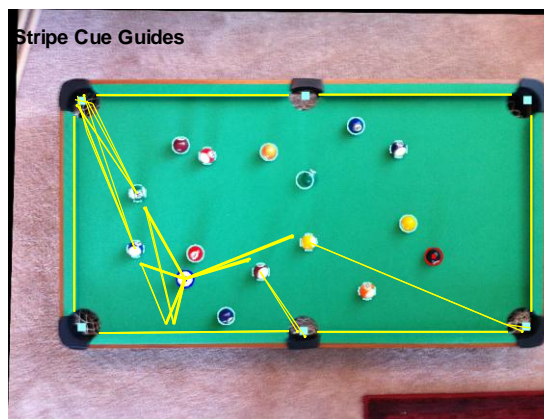


Fig. 14. Natural lighting 2, top aspect. Stripes and solids properly identified.

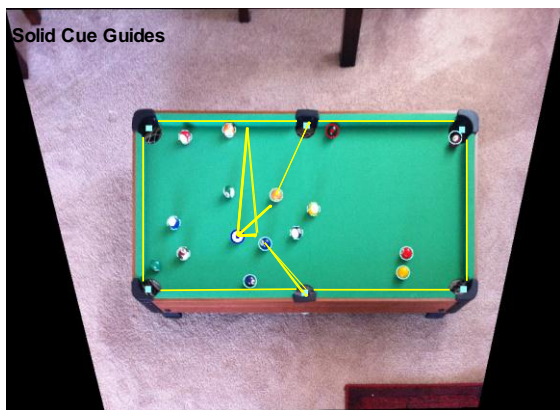


Fig. 12. Natural lighting. Green balls missed due to similar color and amplitude as table.

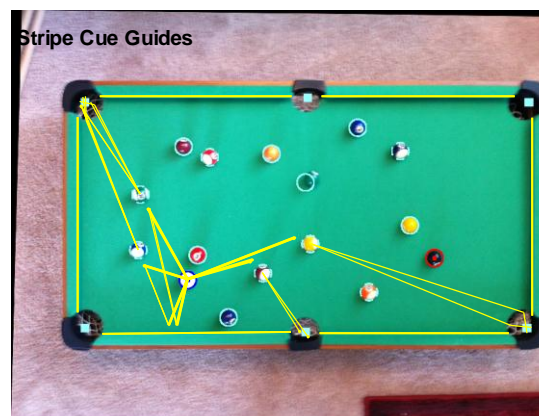


Fig. 15. Natural lighting 2, low res scale. In comparison with Fig. 13, cue guide missed.

V. CONCLUSIONS

This paper outlines an algorithm which works well in identifying pockets, rails and non-green balls under variations in lighting, view aspect, and scale. It does not work well in identifying green balls under natural sun light conditions, but does with the lamp lighting used in tests. Neither does it work well in locating positions of green balls under shadow conditions, understandable since shadows are also green given a green table. Future work may include amplitude threshold refinement to improve detection of green balls, and ball sub-picture thresholding to improve green ball localization in presence of shadows.

Of interest is the dependency of the algorithm's solid vice stripe ball discrimination and ball location on view aspect. Ball location issues are understood, as homography distorts the shapes of object which are not in-plane with the transformation. This may be improved in future work with affine correction of objects distorted by homography. Solid vice stripe discrimination, however, should not be dependent on view aspect, as the color components of each ball should not change. Homographic rectification does change the scale slightly, so variance from scale cannot yet be ruled out as the cause. Future work will focus on blur correction of the image, as adjacent pixel colors may influence white pixel color when blurred.

Also of interest is the dependency of cue guide detection on scale. This is understandable if the cue guide solution applies to a very narrow range of impact angles, which are not

simulated accurately under lower resolution image situations. Even so, the exact mechanism causing the issue is unknown at this time, and requires further investigation.

ACKNOWLEDGMENTS

The author would like to thank Professor Bernd Girod for his excellent course on image processing, and assistant David Chen for his teaching support. The author would especially like to thank assistant Matt Yu for his mentorship on the author's project, and helping the author investigate resources on image homography.

REFERENCES

- [1] M. Takahashi, T. Kasai, and Y. Suzuki, "Support System for Pocket Billiards," SICE Annual Conference, August 2008.
- [2] Y. Ling, S. Li, P. Xu and B. Zhou, "The Detection of Multi-Objective Billiards in Snooker Game Video," Third International Conference on Intelligent Control and Information Processing, July 2012.
- [3] T. Jebara, C. Eyster, J. Weaver, T. Starner, A. Pentland, "Stochastics: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers," First International Symposium on Wearable Computers, October 1997.
- [4] R. Hartley, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2003.
- [5] D. Young, 2008, Homography_Solve Algorithm. [On-line]. Available: <http://www.mathworks.com/matlabcentral/answers/26141>. [January 2012].