Method to Identify and Score Darts thrown into Dartboard

Jacob D. Delaney
Stanford University
jdd1372@stanford.edu

***Abstract*—The author examines the problem of scoring a game of darts by use of image processing techniques. Dart enthusiast today are forced to choose between playing on regulation equipment and manually keeping score, or to use inferior, non-regulation, electronic dartboards. This paper presents a novel method of using images gathered from a stationary camera to automatically segment the board into each scoring region, then identify the position of darts thrown into the board.**

***Index Terms*—Image processing, darts, unsupervised segmentation, straight line detection, color morphological processing, foreground detection.**

I. Introduction

The game of darts is a throwing sport in which participants toss projectiles into a circular target attached to a vertical surface. The target is divided into many regions which correspond to different point and multiplier values. Darts is commonly played across North America and Europe and is a popular past time for many so an application that allows players to more conveniently keep score would be widely beneficial.

"Darts" is a general term for a targeting game following this basic premise and many game variations exist within this archetype; all utilizing the standard dart projectiles with a regulation dart board. Each game variant may have different objectives for which players to aim but identifying the region in which the dart has hit in the dartboard is necessary for proper scoring. This paper proposes the use of image processing to identify thrown darts and to detect which scoring region of the dartboard the dart has struck.

The method outlined within takes images from a stationary mounted camera as inputs. First an image of the dart board with no darts is used to determine point regions, then additional images of darts within the board are processed to provide an output that concludes which player threw the dart and the point value of the thrown dart.

II. Related Work

Electronic dartboards that automatically keep score using sensors built into the board itself are widely available. These types of boards are generally not preferred by dart enthusiast, however, due to the necessity to use plastic hardware that is prone to break and has a perceived 'amateur' feel.

There is no known work directly related to the specific task of identifying and localizing darts using image processing. Though, there has been extensive research in the areas of color morphological processing, region detection, foreground detection, and adaptive thresholding; the results of which are used to achieve the goals of this project.

III. Techical Approach

A standard dartboard is circular in shape and divided radially into twenty sections, alternating in color between light and dark spaces, which each represent a separate point value, with two concentric circles in the center known as the bullseye. Additionally, the board contains two outer rings, alternating between green and red spaces, which subdivide each scoring section into single, double, and triple point value areas, Fig 1. Keeping accurate score requires that the image processing approach can correctly determine both the location that the dart lands on the board and the point value of that region of the board.

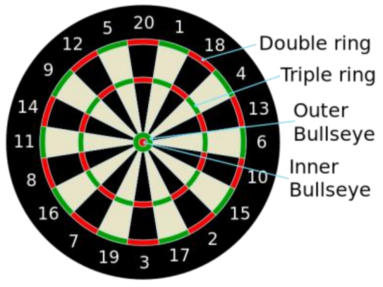


Fig. 1. Regulation Dart Board

The approach used in this project is divided into two main tasks: region segmentation to assign a point value to each pixel in the image, and dart localization to locate the dart in the image and determine the pixels that correspond to where the dart tip contacts the board. The following two subsections describe the methodology used to accomplish each of these objectives.

A. *Region Segmentation*

The board region segmentation process is performed on a background image, $B_{x,y}$, of the dart board before play begins. This section examines the processes implemented to identify multiplier regions based on color morphological processing, and the radial dividers based on straight line detection algorithms.

We begin by converting $B_{x,y}$ to grayscale and thresholding using Otsu's method [1]. The resulting black and white image is used to crop $B_{x,y}$ to remove superfluous background pixels and decrease processing time. Next, we use the known color regions of a standard dartboard to create a pixel mask of the dart board point multiplier regions. First, with the knowledge that the double and triple rings consist exclusively of alternating red and green areas, we are able to take the intensity of the red and green channels of $B_{x,y}$, subtract the grayscale background image to eliminate the white pixels, and threshold again using Otsu's method, to obtain red and green pixel masks. A pixel-by-pixel logical OR of these two masks results in a mask that contains all double and triple multiplier regions, as well as the bullseye regions in the center, Fig. 2.

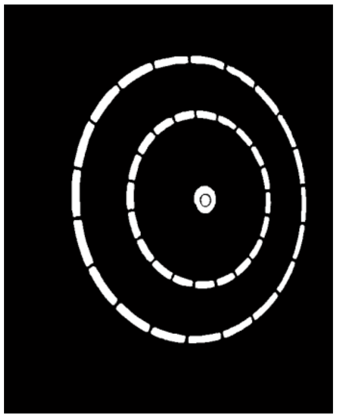


Fig. 2. Red + Green Pixel Mask

All of the multiplier regions can now be derived from the combination of the red and green pixel masks. First, the binary image in Fig. 2 is 'closed' by dilating and subsequently eroding by a structural element calculated to be roughly the size of the wire dividers. This process removes the spaces dividing the red and green masks and creates continuous rings. Next, the holes are filled inside the resulting continuous rings to create a binary map of the entire scoring area. The inverse of this region (outside of the scoring area) constitutes the zero multiplier region, Fig. 3(a), in which all darts count for zero points.

The single multiplier region, Fig. 3(b), is then found by subtracting the red and green masks from the scoring region because it is known that the single region consists only of light and dark regions. Next, the double multiplier region, Fig. 3(c), is established by filling the holes in the single region and subtracting from the entire scoring region. Similar manipulation of the board region masks are then used to determine the triple, outer bullseye, and inner bullseye regions – shown in Fig. 3(d), Fig. 3(e), and Fig. 3(f), respectively.

With all the multiplier regions now known, the next task is to segment the board based on the metal radial dividers to assign point values for each slice of the board. A Canny edge detection operator [2] is used on the grayscale conversion of the background image, $B_{x,y}$, to create a black and white image of the strongest edges. The Standard Hough Transform [3][4] of the binary edge map is then computed and the ten greatest peaks are determined, Fig. 4. These peak angles correspond to the angle of the metal dividing bars that bisect the board, Fig. 5. Using the dart board's standard numbering pattern, a point value is determined for pixels that fall within each straight line angle. With the information gathered from the multiplier regions and the straight line angles, we can derive an algorithm that relates every pixel on the background image to a point value.

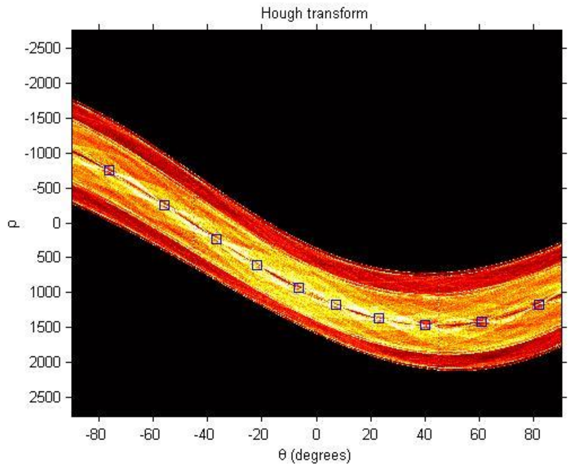


Fig. 4. Hough Transform of Background Edge Map

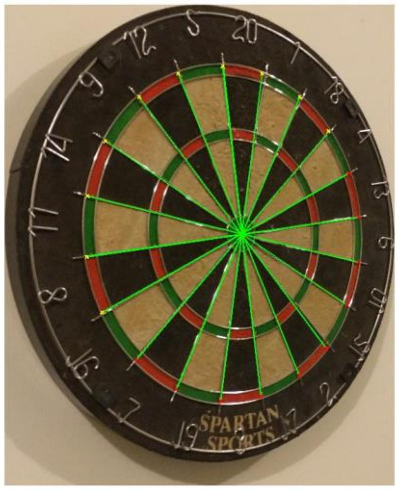
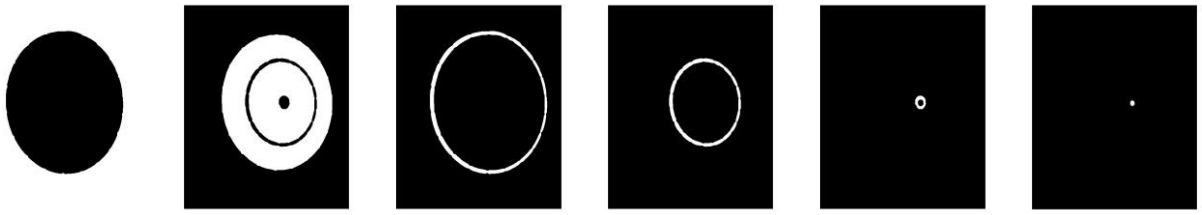


Fig. 5. Hough Lines Superimposed on Background Image



(a) (b) (c) (d) (e) (f)

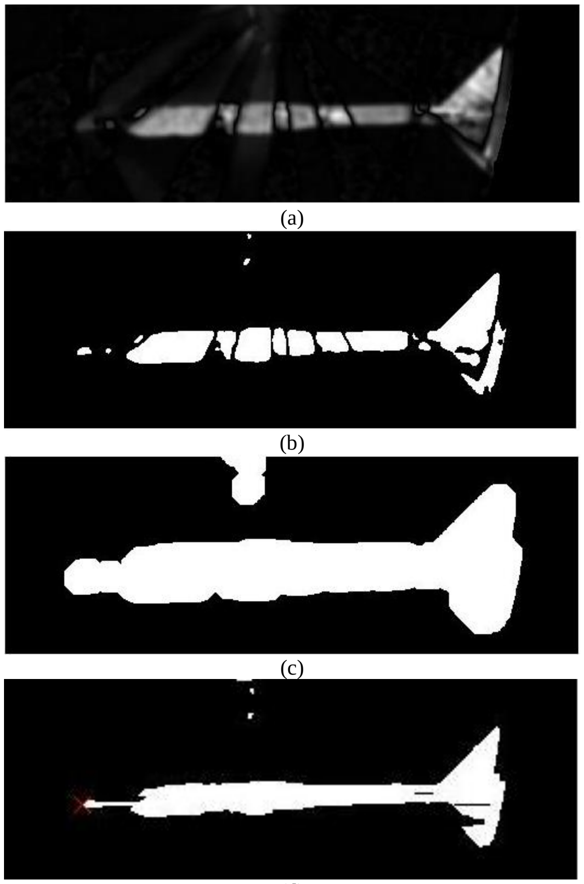
Fig. 3. Point Multiplier Region Maps – (a) Zero Multiplier Region, (b) Single Multiplier Region, (c) Double Multiplier Region, (d) Triple Region, (e) Outer Bullseye, (f) Inner Bullseye

B. Determining Dart Location

Once the background image is processed to create a point map, gameplay can begin. After each dart is thrown, the stationary camera takes another image, $I_{x,y}$, of the dart within the dartboard and the image processing technique attempts to discern the location of the dart and which player the dart belongs to. This process begins by optionally using a SIFT match technique to match features of $I_{x,y}$ and $B_{x,y}$ and find the transform matrix to perform a Homographic transpose to better match the two images. This process is used to account for small movements in the camera between the times the two images are taken.

With the two images now aligned, each image is filtered with a Gaussian kernel to smooth out sharp transitions such as light glares. Then an image difference technique is used to detect the dart in the foreground [5]. An adaptive difference equation is used because taking the absolute difference results in intensity variations between the regions light and dark patterns on the background image, which make global thresholding difficult. The algorithm used normalizes the difference of the dark regions and the light regions separately then combines the result to form a grayscale image of dart in the foreground, Fig. 6.

This grayscale image is the converted to a binary image by thresholding using Otsu's method to create a binary mask of the

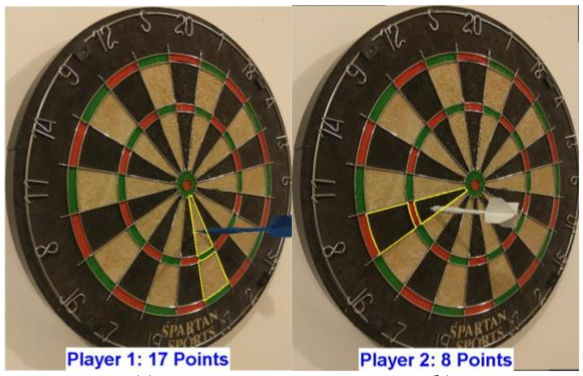

(a)

(b)

(c)

(d)

Fig. 6. Foreground Dart Detection (a) grayscale image, (b) binary representation, (c) orientation model, (d) final detection image with extrema pixel marked



(a) (b)

Fig. 7. Output Display Images

dart. This dart mask is dilated using a round structural element to adjoin any segmented regions, followed by calculation of the orientation [6] to determine the primary angle that the dart entered the board. Next, the original dart mask is 'closed' by dilating and subsequently eroding by a straight line structural element rotated based on the orientation of the dart. This results in the final foreground image that we use to determine the pixel location of the dart tip, based on the extreme value along the angle of orientation. This pixel location is compared to the point map to determine the point value of each dart throw.

The final task is to determine which player's dart is in the image, $I_{x,y}$. In this project, the darts were color coded to improve the foreground detection accuracy and assist in dart recognition. Therefore, the player can discerned based on a comparison of color channel values. Sample results using this technical approach are shown in Fig. 7.

IV. Exprimental Results

Test images were gathered from a stationary camera that was adjustable to capture images at different viewing angles. Several simulated dart throw images were captured intended to test all regions of the board from each angle. The methodology proposed in the paper was developed and tested in MATLAB. The effectiveness of the point map algorithm and the accuracy of the dart location algorithm were evaluated.

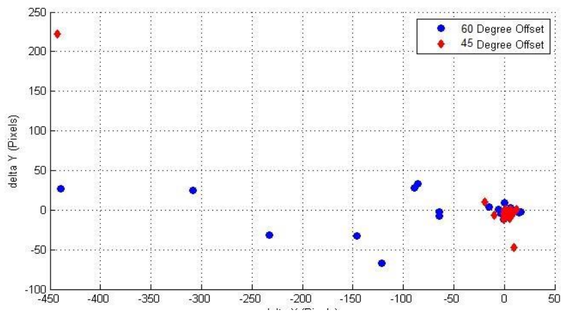
The point map accuracy was tested by running each background test image through the developed MATLAB script and hand detecting the error regions using the MATLAB `roipoly` function. Since this technology is ultimately intended to be used in a real time system, accuracy and speed are key benchmarks. Table 1 shows that the region detection algorithm is consistently very accurate when the camera is mounted from a narrow viewing angle but quickly miscalculates images from very wide angles. The time scales nearly linearly with image size. Therefore, in practice, the image resolution could be decreased to increase computation time without greatly decreasing the accuracy.

Table 1. Region Segmentation Results

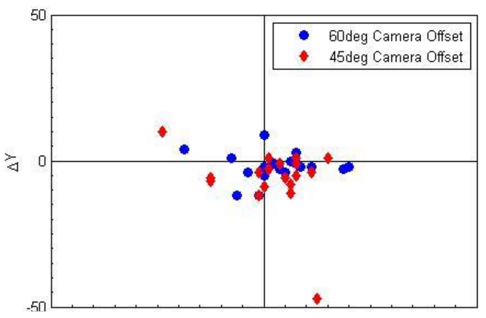
Viewing Angle			*60°*	*45°*	*30°*
Image Size Reduction	*x1*	*Error Pixels*	25988	29125	12064
		Board Pixels	2278058	3044984	3411796
		Percent Error	98.86%	99.04%	99.65%
		Norm. Time	0.75	0.90	1.00
	x2	*Error Pixels*	39785	7715	4276
		Board Pixels	569523	761228	852845
		Percent Error	93.01%	98.99%	99.50%
		Norm. Time	0.42	0.50	0.51
	x4	*Error Pixels*	20048	2983	1886
		Board Pixels	142375	190277	213176
		Percent Error	85.92%	98.43%	99.12%
		Norm. Time	0.29	0.26	0.22

The effectiveness of the dart location procedure was tested by running each dart image and its respective background image through the dart detection algorithm and comparing the generated hitpoint pixel coordinates to the actual pixel coordinates that were manually determined by inspection of the image. The difference between the actual and tested pixel coordinates for viewing angles of 45° and 60° are shown in Graph 1, results from the 30° viewing angle were very erratic and excluded from this plot. This is due to the algorithm not being able to detect accurate dart orientation from such a straight-on observation point. It can be seen that from a moderately narrow viewing angle the method is highly precise but for wider viewing angles, the precision drops considerably. Test data shows that for a 45° viewing angle the average displacement from the origin is only 13.17 pixels for a native 3024x4032 resolution image, excluding outliers from not properly segmenting the foreground.

Graph 1. Dart Detection Error



Graph 2. Dart Detection Error (Excluding Outliers)



V. Conclusions

The results obtained using this technique provide a good proof of concept for digitalizing dart scoring while maintaining the use of regulation equipment. The proposed method was able to localize darts to within a few pixels of their actual position and report the points scored. Further work includes porting this MATLAB model to take advantage of modern DSPs in order to speed up processing time. Also, additional work in areas such as multiple dart detection and foreground recognition is suggested in order to increase the accuracy and precision of the dart localization and to make this method practical in real-world use. Furthermore, this paper provides only an intuition for the best camera viewing angle. Further research is necessary to find the optimal offset angle for the stationary camera.

VI. References

[1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[2] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[3] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 (January, 1972).

[4] L. A. F. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," Pattern Recognit., vol. 41, no. 1, pp. 299–314, 2008

[5] Efficient Salient Foreground Detection for Images and Video using Fiedler Vectors. Federico Perazzi, Olga Sorkine-Hornung, Alexander Sorkine-Hornung. Eurographics Workshop on Intelligent Cinematography and Editing, May 2015, Zurich, Switzerland

[6] Vailaya, Aditya, et al. "Automatic image orientation detection." Image Processing, IEEE Transactions on 11.7 (2002): 746-755.

[7] "Supervised ordering in: Application to morphological processing of hyperspectral images." Velasco-Forero, Santiago, and Jesus Angulo. Image Processing, IEEE Transactions on 20.11 (2011): 3301-3308