Dart Score Estimation

Hao-Yu Hsu
National Tsing Hua University
Hsinchu, Taiwan
maxsyu88@gapp.nthu.edu.tw

Jyun-Yi Wu
National Tsing Hua University
Hsinchu, Taiwan
kershaw@gapp.nthu.edu.tw

Abstract

The author examines the problem that dart enthusiasts find it hard to choose what kind of dartboard to play with. Counting scores manually is needed for regular dartboards, while it is not needed for electronic dartboards. But, electronic ones do suffer from low level of quality in trade-off. In this paper, we proposed a novel method to estimate dart scores by gathering images of dartboard from a stationary camera. The experimental results show that we can successfully segment the board into each score region, then identify the location of darts thrown into the board.

1. Introduction

The game of darts has become a popular sport around the world over the past few years. Two types of dartboards are commonly used, regular ones and electronic ones. Despite the fact that dart lovers and professional players prefer regular ones due to its better quality, the biggest drawback of regular dartboards is to record the score manually. Hence, an automatic dart scoring system would be helpful to solve this issue.

This paper makes use of image processing to segment score regions and determine the location of darts struck on the board. Currently, projects related to dart score estimation have mostly been implemented in software gaming, like web game applications. There is no work that directly evaluate dart score by capturing dart images from a stationary camera in real-world scenario. However, there has been extensive research in image morphology processing, feature matching and adaptive thresholding. All of them helps to achieve the goal of this paper.

The remainder of this article is organized as follows. Section 2 describes the methodology. Section 3 presents the experimental results. Section 4 concludes this article.

2. Methodology

The automatic scoring process is divided into two main tasks, *Region Segmentation* and *Dart Location Estimation* respectively. The purpose of region segmentation is to create a point map, which assigns a score value of each pixel in the image. On the other hand, the objective of dart location estimation is to identify the pixels location of the dart tip. Eventually, the final score can be determined by mapping the dart tip location to the point map created in region segmentation part. Our implementation pipeline is shown in Figure 1.

2.1. Region Segmentation

Region segmentation process is performed on a background image before playing. In this section, we identify multiplier regions by color morphological processing, and the radial dividers by straight line detection algorithms.

The background image is converted to grayscale, then converted to binary image thresholded by Otsu's method [6]. Otsu's method is used to perform automatic thresholding. After that, the binary image is inverted and floodfilled to get the potential area of the dartboard. The process is shown at Figure 2. The background image is cropped based on the potential area in order to reduce the effect of background patterns and computational time.

Alternating green and red regions in double rings and triple rings are utilized for multiplier regions segmentation. The intensity of red and green channels are extracted from the background image, followed by subtracting them with the grayscale image, thresholded by Otsu's method later on, to obtain green and red region masks. A pixel-wise logical OR is performed on both masks to get the binary masks of double and triple multiplier regions, as well as the bullseye region in the center.

All of the multiplier regions can be derived from green and red region masks. First, the binary masks are closed by dilating with a disk-like structural element. This step is to fill up the tiny gap in the double and triple multiplier regions and make them a continuous rings. Then, the continuous rings are filled up from the outer ring and the scor-

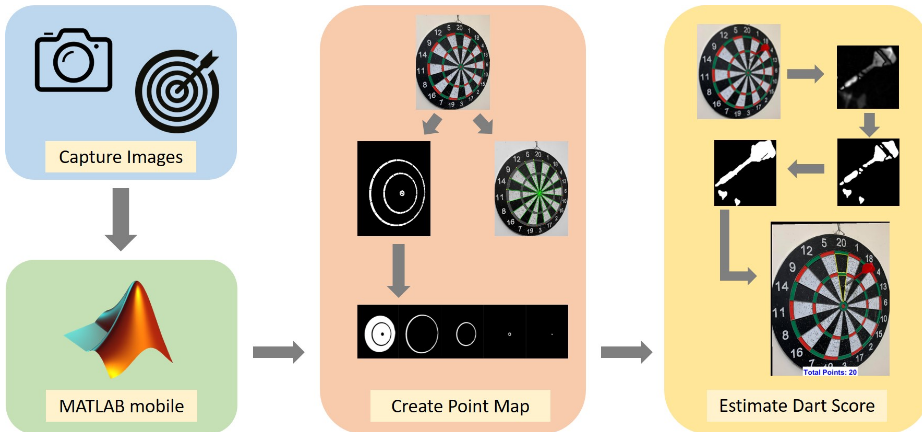
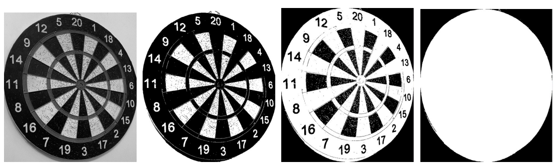


Figure 1: The experiment pipelines



(a) (b) (c) (d)

Figure 2: The process of locating the dartboard (a) Grayscale image (b) Binary image after Otsu's method (c) Inverse binary image (d) Filled image

ing area is formed. The area outside the scoring area is considered a miss, in which counts for zero point. Single multiplier region is derived by subtracting multiplier region masks from scoring area. As for double multiplier region, it is obtained by filling up single multiplier region, then subtracting this region from the scoring area. Similar operation is performed to extract all the other multiplier regions. (See Figure 3)

With the information of multiplier regions, the next mission is to segment the board based on the radial dividers (metal frame) to assign the score value to each pixels of the board. Canny edge detector [1] is applied to grayscale background image to extract the edge contours of the board. Then, Hough Transform [3] is used to transform the extracted image from (x, y) to (r, θ) (See Figure 4). Ten greatest peaks in Hough space are chose to be the radial dividers (See Figure 5), that are ten metal bars separating the score regions. Each score region is known by matching the angles of separating lines with the location of each score value, which is 20, 1, 18, 4, 13... in clockwise direction. Finally, the score of one pixel can be easily obtained based on the radial dividers and the multiplier regions.

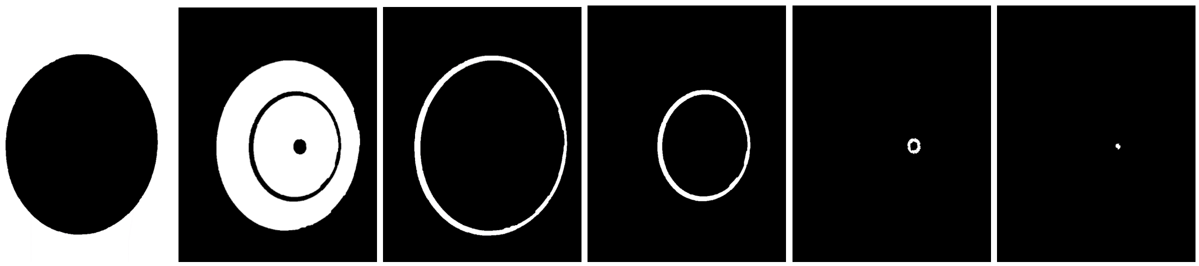
2.2. Dart Location Estimation

In dart location estimation, the dart tip's pixel location is calculated. The approach starts with taking another picture with darts thrown into the dartboard using the same stationary camera. To solve the slight movement issue of the camera, Scale-Invariant Feature Transform (SIFT) [5] is applied on the background image and the dart image respectively to find their common features. Next, a homography matrix [2] is calculated by running RANSAC [4] algorithm in matching SIFT features, by which the dart image is warped back to the background image. Hence, two images are aligned together.

Sequentially, we want to know where the dart is tossed. To do so, the difference image is computed by subtracting the grayscale dart image from the grayscale background image. Gaussian blur filter is applied to lessen the influence of sharp transition, such as light glare, before computing difference image. Since their distinction is tiny, min-max normalization is performed on the difference image to improve dart image's visibility. (See Figure 6)

The following task is to make dart more visible. Otsu's method has been applied on dart image to enhance the contrast and the result image is converted into the binary one. Then, image dilation is applied on the binary image in order to connect scattered parts of darts. For each connected area, the orientation is found by *regionprops()*, a built-in function of MATLAB. To find the location of dart tip precisely, each connected area is closed along its own orientation. (See Figure 7 for the whole process)

In the regular dart competition, chances are it may be multiple darts on the dartboard. So, it is necessary to find the number of darts on the dartboard for automatic scoring. It can be observed that if there are darts on the dartboard, their connected area has width long enough. Therefore, a metric is set $isDart = \frac{DW}{DBW}$, where DW is dart width



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

Figure 3: The region masks of the dartboard (a) miss region (b) single multiplier region (c) double multiplier region (d) triple multiplier region (e) outer bullseye (f) inner bullseye

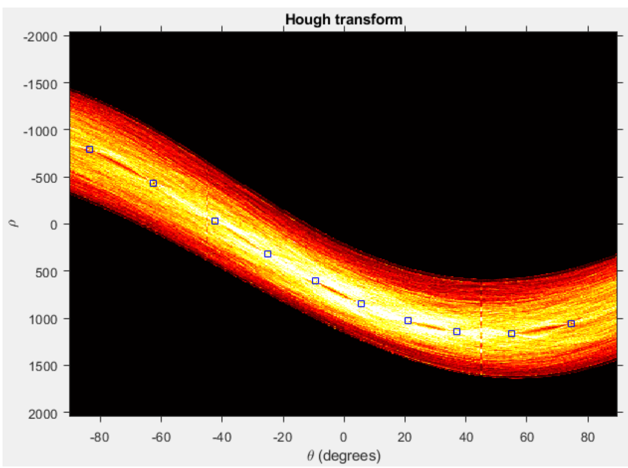


Figure 4: Hough Transform of the dartboard image

and DBW is dartboard width. If $isDart >= threshold$, the connected area is considered a dart. Empirically, the $threshold$ is set to be 0.1.

To find the tips of every found darts, their orientations are found through *regionprops()*. If $angle < 0$, we take the bottom-left coordinate of their connected area. Otherwise, we take the top-left one. These coordinates represent the dart tips. Combined with the point map, the total score can be easily computed.

3. Experimental Results

To setup this experiment, a mobile device is fixed on a phone stand, facing the dartboard from a certain angle about 30 to 45 degrees. Since our proposed method is implemented in MATLAB, MATLAB mobile is chose to perform dart score estimation. In this way, the data is efficiently processed on remote cloud engine without having heavy loads running on local mobile device.

Start the experiment by capturing the background image by the mobile device before throwing darts, and then computing its point map. Later on, the dart image is taken after each throw. Finally, both images are combined and computed based on our proposed algorithm to get the result image, and display it on the device. Some experimental results are shown in Figure 8.

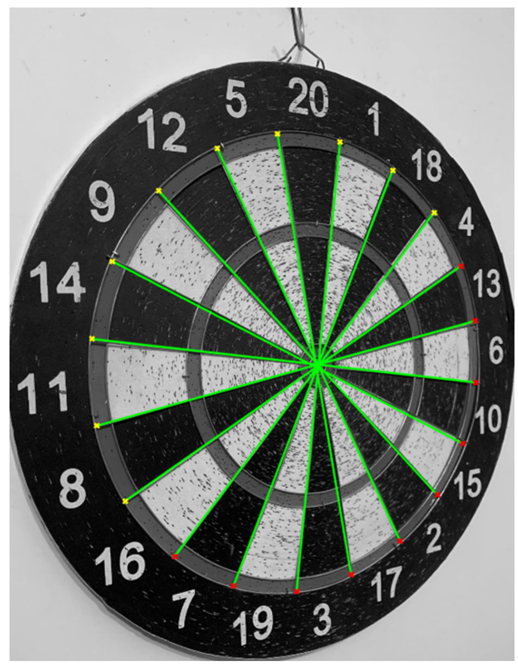


Figure 5: The radial dividers of the dartboard

4. Conclusion and Discussion

We successfully identify score regions dart location for most cases, and further perform automatic scoring with basic Digital Image Processing & Computer Vision techniques.

However, Limited by insufficient image quality (1280×720) due to MATLAB Mobile configuration, the estimated dart location is sometimes imprecise. Also, our approach is

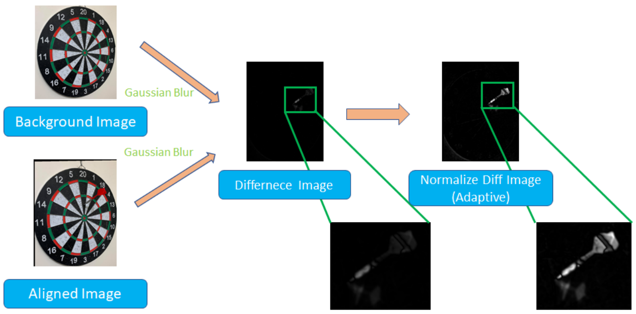
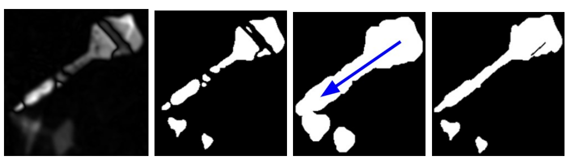


Figure 6: Process of computing the difference image, the dart area becomes more salient after applying min-max normalization on the difference image



(a) (b) (c) (d)

Figure 7: (a) grayscale difference image (b) binary difference image by Otsu's method (c) binary difference image dilated by disk structural element and find the main orientation (blue arrow) (d) image closing on difference image along the main orientation

(a) (b) (c)

Figure 8: Experimental results of our proposed algorithm

incapable of dealing with overlapping darts seen from the point of view of the camera. And if the ambient lighting is not evenly distributed, the shadow of the dart could lead to dart misidentifying.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986.

[2] O. Chum, T. Pajdla, and P. Sturm. The geometric error for homographies. *Computer Vision and Image Understanding*, (97):86–102, 2005.

[3] R. Duda and P. Hart. "use of the hough transformation to detect lines and curves in pictures. *Comm ACM*, 15:11–15, 1972.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981.

[5] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 2:1150–1157, 1999.

[6] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.