A Real-time Velocity Estimation using Motion Blur in Air Hockey

Sh. Rezvankhah, A. A. Bagherzadeh, H. Moradi, B. Nadjar Araabi *Member, IEEE*

Abstract **In this paper a real-time approach for the air hockey puck's velocity estimation based on motion blur is presented. In the proposed approach first a low resolution image is used to detect the puck and then the Fourier transform is used on the full resolution image to extract the direction of the motion. This hierarchical approach provides faster processing speed. The roundness of the ball and the length of the blur are finally used in order to estimate the speed. The proposed approach is implemented on an actual air hockey table using two different cameras, one with 30 and the other one with 60 frames per second. The obtained results show promising performance on velocity estimation using off-the-shelf cameras.**

I. INTRODUCTION

A captured image is in fact an average sample taken over a time period, i.e. the period that the camera shutter is open. Consequently, objects that move during this period will appear blurry, a phenomenon that is referred to as motion blur. The length and direction of the blur represent the velocity of the object and can be used to estimate the velocity from a single motion blurred image.

Another approach to estimate the velocity is by comparing its position in two consecutive frames. However it cannot be used in applications in which the object changes direction too fast such that the camera cannot capture all changes. Consequently, it is impossible to estimate the velocity of the object by interpolating between two consecutive frames. Additionally, this approach is not suitable for applications in which the time for waiting to capture two consecutive images is very valuable and cannot be wasted.

In this paper, the first idea, i.e. velocity estimation using motion blur, is used and a method is developed to estimate the velocity of the puck in air hockey game. Air hockey game is a table game that is used commercially in amusement parks and personally at homes. Air hockey consists of a table with two gates (goals) on the sides of the table (Fig. 1). There are small holes on the surface of the table from which the air flow creates an air bearing on which a puck, i.e. the air hockey's disk made of hard plastic, can float fairly frictionless. The paddle, which is also called mallet, is a slightly bigger disk than the puck with a handle on top to be held by a player. The important issue in this game is the high speed of the puck, (e.g. 15 m/s in professional games), and a fast velocity estimation method is needed to predict the position of the puck to defend the goal or hit the puck on the run. For instance, consider a table of 1.5 meters long and a puck that travels from the opponent side to a player side at 15 m/s. The player has at most 100 milliseconds to predict the location of the puck at his own side and respond to it.

Figure 1. The air hockey table with its goal, paddle, puck. The paddle has been colored in red and the puck in black (on front of paddle). A 2DOF robot is supposed to use the velocity estimation by an off-the-shelf camera to predict the position of the puck and block the goal.

The reason for selecting this platform, to test the proposed approach, is that it includes a fairly high speed moving object, i.e. its puck, and a short motion range that makes decision making period very short. In other words, the 2nd approach, i.e. velocity estimation using consecutive frames, is not suitable for the air hockey application since the puck has a very high speed and can change direction very fast. The direction may change very fast if the puck hits one of the walls.

In contrast to velocity estimation using consecutive frames, the proposed approach relies on motion blur which uses only a single image to estimate the velocity. Consequently, it would not need a high speed camera for velocity estimation and does not require multiple frames to do estimation. Furthermore, it is more suitable for the air hockey application since it only needs a single frame to estimate the velocity and would not suffer from change of the direction when the puck hits a wall. An advantage of this approach in the air hockey application, this method is independent of the table and puck's color, exploits the static table as background and makes use of the circular shape of the puck.

II. Related Work

There have been several speed estimation algorithms based on motion blur and other algorithms in order to estimate the extent of the motion blur. For instance, there is a study to estimate the velocity of vehicles using traffic cameras [7]. Another example of speed estimation is to estimate the speed of spherical objects against a static background [6]. Unfortunately, most of these algorithms are too complex to be done in real time and thus cannot be used in the of case air hockey game.

In another approach [8] it is assumed that the blurred area consists of a solid part and a blurred part and the proposed method measures the length of blurred part [8]. In the case of air hockey, the whole puck looks blurred and there is no solid part.

In the method proposed in [4] uses the texture on the object to estimate the magnitude of velocity. Unfortunately, it is not suitable for speed estimation in air hockey since the puck does not have texture.

In this paper, a method based on motion blur is proposed that does not need multiple frames to determine the speed. Furthermore, it is real-time and can use off-the-shelf cameras to estimate the velocity. Finally, it has does not have the limitations imposed by other approaches such as texture on the moving object.

III. The proposed algorithm

The process of velocity estimation is broken into three main parts: locating the puck, estimating the direction of motion and estimating the velocity magnitude.

Preprocessing needs to be done before these steps can be applied. This step consists of illumination normalization, resolution reduction, background extraction and grayscale conversion. Illumination normalization is required in order to provide resilience to illumination changes such as a player's hand shadow on the table. Two levels of resolution reduction is done to reduce computational complexity which is crucial for processing the whole scene and locating the puck in real-time. Finally, the method proposed in [1] is used to extract the background using several frames. Then, the foreground, represented by FG from now on, is calculated for every captured frame by comparing the captured frame, referred to as FR, and the background, or BG.

Finally, the captured frames are converted to grayscale. The main advantage of using grayscale images is that our method becomes color-independent. In other words, the system can detect a puck or paddle regardless of their colors. Furthermore, the approach becomes faster due to processing grayscale rather than color images.

The preprocessing steps are further explained in the following.

IV. Preprocessing steps

A. Illumination normalization

In the process of extracting FG from FR by subtracting BG, it is important to eliminate the illumination differences in FR and BG and not let it appear in FG. To do so, a method in which the known static background is assumed is used. It works based on the fact that an extremely blurred image is a good estimate of an image's illumination. As the goal of illumination normalization here is to have FR and BG with the same illumination, the difference between the two is added to the other one. The kernel used to blur FR and BG is shown in Eq. 1. It should be noted that if the blur mask is too big, the resulting illumination approximation would not include local illumination. If it is too small, the foreground details are eliminated since they would be considered as illumination changes. For instance, the puck may be considered as illumination change and would be removed since it is fairly small and fits in the mask.

$$k = \frac{1}{size^2}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \quad (1)$$

B. Resolution reduction

Resolution reduction is done in two levels. First the lower resolution is used to approximately locate the puck. Then the higher resolution of the location of the puck, that is the high resolution image of Region Of Interest (ROI), is used to more accurately locate the puck. In other words, the puck is located and processed in a hierarchical manner to increase the processing speed.

There are several resolution reduction algorithms with different speeds and qualities. In this study, a method is used consisting of blurring the image and then choosing the nearest neighbor as the value of the blurred area. The reasons for choosing this method are: a) it runs very fast, b) the resulting image has relatively good quality, and c) it is supported by graphic hardware. In this approach, a kernel is convolved on the image to blur the image (Eq. 2). Then the image size is reduced 4 times by removing even rows and columns.

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (2)$$

C. *Background extraction*

In this approach the background subtraction method proposed in [1] is used on the reduced resolution frames to extract BG from upcoming frames. It is possible to use a single image to extract the BG which would provide slight performance improvements but has less flexibility in comparison to automatic background extraction.

V. LOCATING THE PUCK

After extracting the foreground at two different resolutions, the puck should be detected. It is important to mention that in the lower resolution, the pixels belonging to the puck form a contiguous blob. Consequently, the disjoint set algorithm [5] can be used to detect the segments in the foreground from which one segment is the puck. This is done by scanning the lower resolution foreground once from top left to bottom right and using union operation between every foreground pixel and its left, up and left-up neighboring foreground pixels.

After the segments are determined in the image, the puck should be detected as one of these segments, i.e. the pixels belonging to the same set. The following four features can be used to determine and locate the puck: position of the segment centers, size of segments, position of the segments with respect to the table, and the estimated puck position from the previous frame.

The first feature, i.e. the position of the center of the segment, represents the fact that the segment centers which are near the either sides of the table are usually a players' hand because the player in air hockey is required to stay at one side of the table. The second feature is based on the fact that the size of puck is limited, and so is the size of the segment corresponding to the puck. Consequently, the segments which are too large can be eliminated. The 3rd feature is based on the fact that a segment or part of a segment cannot be out of the table, which may happen with the segments corresponding to a player's hand. After eliminating large segments and those which are not entirely in the table, the segments are scored based on their distance from the player's end of table, i.e. the side of the player who is going to hit the puck. If there are several segments without a distinct score advantage over each other, the position and velocity estimation of the puck in the previous frame are used to estimate the position in the current frame. Then, the nearest segment's center to the estimated position is selected as the puck's position.

After approximately locating the puck, the higher resolution frame is used to exactly determine puck's position. Unfortunately, in this higher resolution FG the puck's pixels do not form a contiguous blob and cannot be easily detected. Consequently, k-means algorithm is executed on the foreground pixels with k corresponding to the number of foreground pixels divided by puck's known size. The result of k-means algorithm will be several centroids, from which we assume the nearest centroid to the approximated puck's position, which is determined in the previous step, to be the exact position of the ball.

The reason to perform puck detection and localization in two steps is that in high resolution foreground the segment size cannot be easily determined. Furthermore, it is not easy to detect and reject the segments with pixels falling out of the table. Finally, it would be much faster to run the localization algorithms in two levels to reduce the computational cost.

VI. ESTIMATING VELOCITY DIRECTION

Now that the exact position of the puck is determined, the operation would be limited to a window around the puck. In order to determine the direction of the puck's movement, a two dimensional Fourier transform is used (Fig. 2).

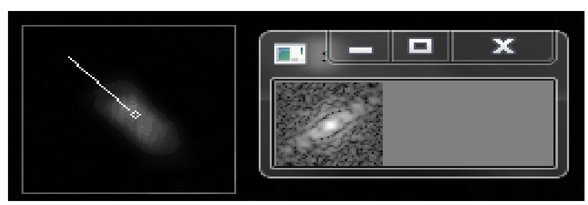

Figure 2. Direction can be extracted using the Fourier transform of a window around the puck in the foreground. The left image shows the direction of motion and the right image shows the Fourier transform.

Fourier transform has already been used to determine the orientation of text in a document. Also it has been used to determine the orientation of fingerprints. In these applications, a Fourier transform based filter is chosen to be applied to the image in all possible directions and choose the direction with maximum response as the direction of the text or the fingerprint. However, it is not suitable for our real-time application since it is not feasible to check all possible directions in real-time. Thus the steerable filters [4] are used. In steerable filters the response of the filter for any desired direction can be mathematically calculated by interpolating the response of a few basis filters. This makes these filters suitable for real-time applications. Eq. 3 shows the three basic steerable filters and the combined filter.

One of the challenges of extracting direction from the Fourier transform is the existence of DC values which cause high values on horizontal and vertical axes of the 2D Fourier transformed signal, shown in figure 3. The magnitude of these values conceals the values related to the original orientation of the puck. The DC values were one of the reasons we calculated BG and then FG. We remove these values by applying the Fourier transform on the FG instead of the FR itself.

$$RG_2^\theta = k_a(\theta)RG_{2a} + k_b(\theta)RG_{2b} + k_c(\theta)RG_{2c}$$

$G_{2a} = 0.921(2x^2 - 1)e^{\mu}$	$k_a(\theta) = \cos^2(\theta)$
$G_{2b} = 1.843xye^{\mu}$	$k_b(\theta) = -2\cos(\theta)\sin(\theta)$
$G_{2c} = 0.921(2y^2 - 1)e^{\mu}$	$k_c(\theta) = \sin^2(\theta)$
$\mu = -(x^2 + y^2)$	

(3)

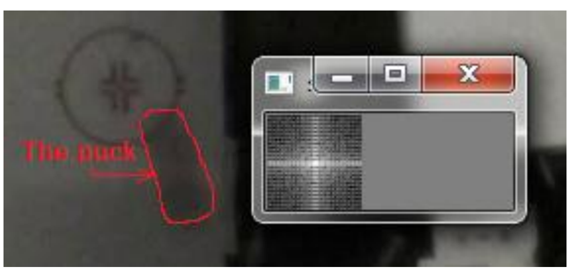

Figure 3. The Fourier transform of FR and the resulting DC values on horizontal and vertical axes

VII. Estimating Velocity Magnitude

The roundness of the puck and the fact that the puck stretches more in higher speeds than the lower speeds is used to estimate the speed. In other words, the higher the speed, the higher the number of blurred pixels along the direction of motion. Consequently, the distribution of the blurred pixels along the direction of velocity is determined, simply by determining the number of pixels along that direction. Similarly the distribution of the pixels along the direction perpendicular to the velocity direction is determined (Fig. 4). As shown in figure 4, the higher the speed of the puck, the more stretched the image of the puck along the direction of the velocity and the sharper the distribution of the pixels compared to the other direction. We show that the puck's speed can be calculated from the sharpness difference of the two distributions. More specifically, the difference between the second derivate of the diagrams would determine the speed of the puck.

To overcome different lighting conditions and the use of various cameras that can make different FG intensities, first a simple thresholding is performed. In other words, this approach would eliminate the need for calibration.

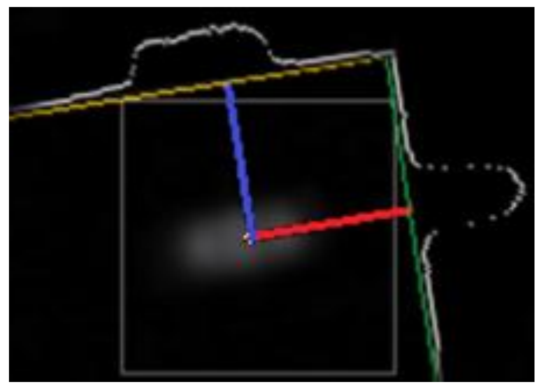

Figure 4. The distribution of the pixels around the puck in FG along the direction of motion and perpendiculr to it. . The distribution along the direction of motion is sharper than the distribution perpendicular to the direction of motion.

VIII. Results

The method is implemented using openCV and can process 30 frames per second taken from a camera in real-time. To test the algorithm, several video streams were taken from an air hockey table. To make the setup more realistic, different players hit the puck at random initial position. Furthermore, two different cameras, one a cell phone camera and the other a flea3 camera by Point Grey, are used to show the robustness against using different cameras. The direction and magnitude of the puck's motion was calculated and compared with the actual values. The average error in the direction of motion was 7.79 degrees. Fig. 5 shows the estimated speeds versus the actual speeds. As it can be seen, the error is fairly small with 0.94 coefficient of determination.

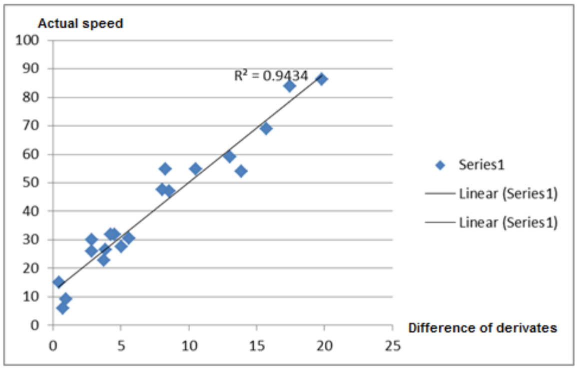

Figure 5. The relationship between actual speed and the calculated difference of derivates from the FG

IX. Calculating the Physical Velocity

In the proposed approach, the estimated direction of motion based on the captured frames is the actual direction of motion on the air hockey table. However, the estimated speed of motion is calculated in pixels per frame and needs to be converted to actual physical speed in meters per second. Consequently the characteristics of the camera are required in order to make this conversion.

In this paper a model of pinhole camera is used in order to calculate the puck's physical speed (Fig. 6). In Fig. 6, Δx represents the movement of the puck in the period the frame FR is being captured. Let's use d as a representation of the distance of the camera from the table and d' as the distance between the pinhole and the camera plane. The velocity estimation using the proposed method which is in pixels per frame is shown by Δp and Δt is the time spent to capture one frame, known as the shutter speed. Finally, let's assume the pixel density (pixels per meter) of the sensory screen is presented by ρ. The physical velocity is calculated using the following equations:

$$\Delta x = \frac{d}{d'} \cdot \frac{\Delta p}{\rho} \qquad (4)$$

$$v = \frac{\Delta x}{\Delta t} \qquad (5)$$

X. CONCLUSION

In this paper, a novel approach for estimating the velocity of a high speed object using motion blur is presented. The main advantage of the proposed approach is its real-time characteristics which makes it suitable for applications such as estimating the speed of the puck in the air hockey game. Furthermore, the approach does not need high speed camera makes it suitable for applications that off-the-shelf cameras are used.

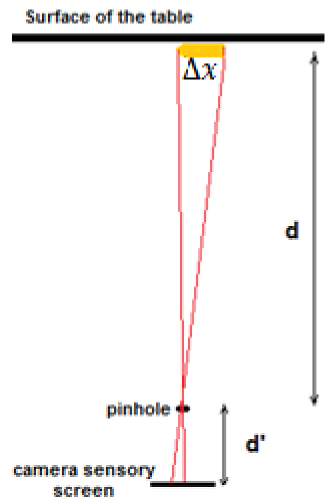

Figure 6. The pinhole camera model used to calculate physical speed

The proposed approach has been implemented on an actual air hockey table and the velocity of the puck has been estimated with good accuracy. The frames have been captured from two different cameras and the robustness of the approach has been proved.

The future work is to improve the accuracy of velocity estimation. Furthermore, impact of errors in velocity estimation in actual game would be evaluated.

Refrences

[1] Z.Zivkovic, F. van der Heijden, Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction, Pattern Recognition Letters, vol. 27, no. 7, pages 773-780, 2006.

[2] Z.Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, International Conference Pattern Recognition, UK, August, 2004.

[3] P. KadewTraKuPong and R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems, 2001.

[4] Rekleitis. I.M. "Optical Flow Recognition from the Power Spectrum of A Single Blurred Image". In: Image processing proceedings, International Conference, vol. 3, pp. 791-794, 1996.

[5] Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford (2001) [1990]. Introduction to Algorithms (2nd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03293-7. Pages 498-522.

[6] H.Y. Lin, C.H. Chang, "Speed measurement of spherical objects using an off-the-shelf digital camera" Proc. 2005 IEEE Conf. Mechatronics pp 66-71

[7] H.Y. Lin, K.J. Li, C.H. "Chang. Vehicle detection from a single motion blurred image", Image Vis. Comput

[8] X. Ting-Fa and Z. Peng. "Image motion-blur-based object's speed measurement using an interlaced scan image", Meas. Sci. Technol. 2010