International Journal of Robotic Computing
Vol. 1, No. 1 (2019) 23-39
© KS Press, Institute for Semantic Computing Foundation
DOI: 10.35708/RC1868-126249

Real-time 6D Racket Pose Estimation and Classification for Table Tennis Robots

Yapeng Gao, Jonas Tebbe, Julian Krismer, and Andreas Zell

Cognitive Systems Group at the Computer Science Department, University of Tuebingen, Germany
{yapeng.gao,jonas.tebbe}@uni-tuebingen.de
julian.krismer@student.uni-tuebingen.de
andreas.zell@uni-tuebingen.de

Received (11/02/2018)
Revised (06/14/2019)
Accepted (07/15/2019)

Abstract. For table tennis robots, it is a significant challenge to understand the opponent's movements and return the ball accordingly with high performance. One has to cope with various ball speeds and spins resulting from different stroke types. In this paper, we propose a real-time 6D racket pose detection method and classify racket movements into five stroke categories with a neural network. By using two monocular cameras, we can extract the racket's contours and choose some special points as feature points in image coordinates. With the 3D geometrical information of a racket, a wide baseline stereo matching method is proposed to find the corresponding feature points and compute the 3D position and orientation of the racket by triangulation and plane fitting. Then, a Kalman filter is adopted to track the racket pose, and a multilayer perceptron (MLP) neural network is used to classify the pose movements. We conduct two experiments to evaluate the accuracy of racket pose detection and classification, in which the average error in position and orientation is around 7.8 mm and 7.2° by comparing with the ground truth from a KUKA robot. The classification accuracy is 98%, the same as the human pose estimation method with Convolutional Pose Machines (CPMs).

Keywords: racket pose detection; pose classification; stereo matching; table tennis robot.

1 Introduction

Racket sports such as tennis, table tennis and badminton are popular worldwide. From a robotic point of view these sports pose several challenges, which should be addressed in real-time, for example, human motion analysis [15], racket 3D pose detection [21], flying ball position estimation [11] and robot trajectory planning [10]. With motion tracking technology for players or rackets, the robots can achieve an anticipatory action predicted from the human's movements, so that there is more execution time left for hitting movements. Tracking human motions or racket motions also allows robots to imitate the human motion to learn how to play human-like table tennis. When a ball flying towards the robot is recognized, a precise hitting position will be estimated by combing ball position and spin together using a curve fitting algorithm [13] or an extend Kalman filter [22]. Finally, the robot will strike the ball back with an optimal human-like action determined by large amounts of training data.

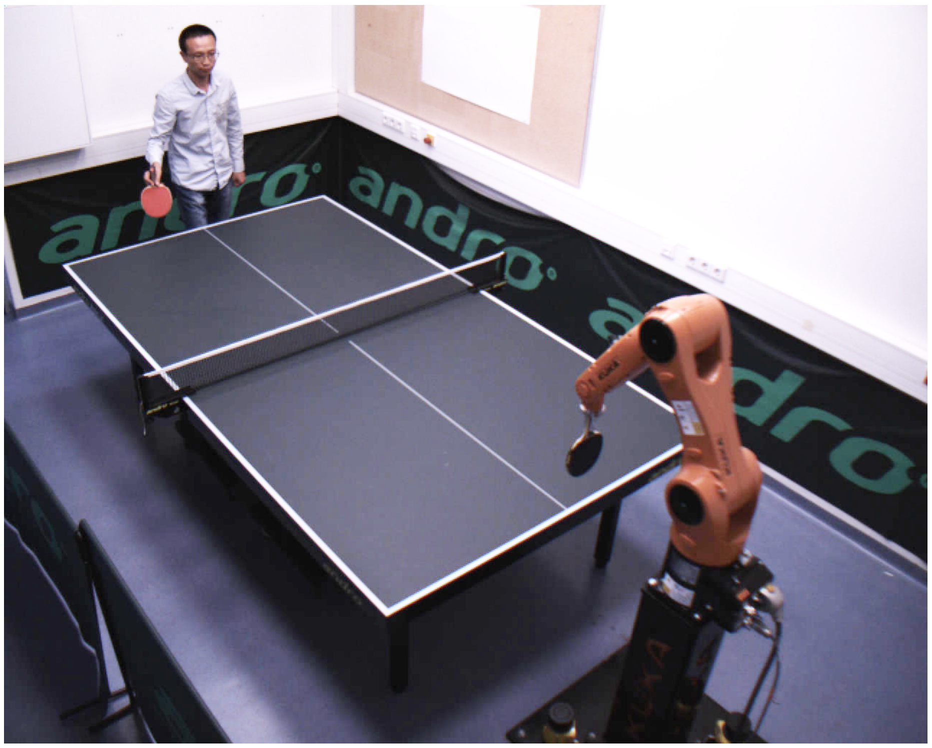

Fig. 1. Table tennis robot system with KUKA Agilus robot. There are four PointGrey Chameleon3 cameras mounted on the ceiling corners far away from each other to occupy more scenario, where two cameras opposite to human are used to detect the racket and another pair is for table tennis ball detection. A table tennis racket is rigidly fixed at the end effector of robot in a type of penhold grip. The robot coordinate is set as the world coordinate and the center of racket is defined as the tool coordinate center.

With various racket movements generating different spin categories, racket sports are full of fun and challenges. To detect the 3D racket pose (position and orientation), much research has been done with sensors and markers. Ohya et al. [8] positioned four stationary cameras in order to cover a large field of view. By assuming the tennis racket to be modeled as ellipse shape, they estimated the 3D racket position with the fundamental matrix, which had ten to forty percent more success rate than only using one camera. Elliott et al. [6] employed a markerless approach with a master camera fixed and a slave camera dynamically located at 21 different positions to detect a set of tennis racket silhouette views. With single view fitting techniques, the 3D racket position was estimated with a spatial accuracy of 1.9 ± 0.14 mm. Chen et al. [4] established a high-speed monocular vision system to track a table tennis racket labeled with some special marker lines in the form of a black rectangle in the middle and a white line parallel to one of the black lines. They can be extracted into several corners as feature points and the pose is computed based on perspective-n-point and orthogonal iteration algorithms. Blank et al. [2] attached inertial sensors into table tennis rackets to detect and classify 8 different stroke types from 10 amateur players. The success rates for detection and classification did reach 95.7% and 96.7%, respectively. Zhang et al. [21] fused inertial measurement unit (IMU) data with the method [4] based on an extended Kalman Filter for obtaining an accurate and robust racket pose. The racket position was computed from cameras and its orientation was estimated from both cameras and IMU resulting in an average angle error of $1.1°$.

In this paper, we present a novel approach for table tennis racket pose detection without markers or IMU based on stereo vision in a table tennis robot system [18]. The system is shown in Figure 1. As the black side of a table tennis racket is nearly invisible against our very dark field enclosure, the current system is restricted to detect the red side only. It can be extracted as a binary contour using a color thresholding method similar to the table tennis ball detection in [18]. To accelerate the detection process, we use bucket fill to find a connected component starting from the estimated point with the specified color threshold that determines the amount of connectivity. By ellipse fitting this contour, we can extract isolated point features located at the intersection area of ellipse and contour. Combing the epipolar constraint with the 3D geometrical size of the racket, we can match the corresponding feature points from two cameras. Triangulation results in 3D points, which are used for fitting the orientation of the plane going through the racket center. A Kalman filter is used to track the 3D pose and smooth the trajectories. Next, we classify these trajectories into five categories using a neural network in order to determine with which kind of spin the ball is played back. In the experimental results, we evaluate the poses against ground truth from a KUKA robot and compare our classification with a different method using human pose estimation.

The subsequent part of this paper has the following structure: Section II introduces related work. The proposed method is presented in Section III. The evaluation and comparison are examined in Section IV. This paper is concluded in Section V.

2 Related Work

2.1 Feature Extraction

Feature extraction involves a detector in the form of points, lines, blobs, or shapes, and a descriptor to generate a unique vector representing these features. ORB (oriented FAST and rotated BRIEF) is one such descriptor [12], which is currently popular. It incorporates the FAST key point detector with modified BRIEF descriptor to provide a fast and efficient alternative for SIFT, SURF, KAZE and BRISK. However, there is an inherent disadvantage in the point-based method in low-textured scenarios in that it will fail due to the lack of reliable feature points. Consequently, line based methods are a possible solution since there are many surfaces like desks, doors and walls in low-textured scenarios, which are rich in line features. In [14], a proposed method with line segments for indoor visual localization is employed to handle low-texture images with a wide baseline, which is far better than other point based methods. In our case, the racket lacks both texture and lines, so that the above methods are not suitable to extract features from the racket face.

2.2 Stereo Matching

Stereo matching defines the correspondence problem, in which we find the corresponding points in two camera images. It is is divided into feature based stereo and area based stereo [12] . Following the feature extraction, feature based stereo utilizes the $L1$ *norm* or $L2$ *norm* for string based descriptors (SIFT, SURF, KAZE etc.) or Hamming distance for binary descriptors (ORB, BRISK etc.) to differentiate features in corresponding pairs [17]. Area based algorithms depend on the epipolar constraint for rectified images to search the corresponding points in the same image rows including local (NCC, SAD) and global methods (dynamic programming, graph cuts). A well known approach for real-time stereo vision is Semi-Global Matching (SGM) [9], which approximates a global 2D matching cost aggregation by minimizing the energy function from 8 or 16 different directions through the image. It can obtain the same accuracy as global matching but with lower runtime. Recently, end-to-end deep stereo gas become very popular to solve the stereo matching problem with CNNs models, consisting of embedding, matching, regularization and refinement modules [19]. However, they currently cannot yet fulfill real-time requirements.

2.3 Pose Classification

Player motion analysis is beneficial because the motion of the player determines the motion of the racket, and consequently the speed and spin of the ball. Chu et al. [5] extracted histogram of oriented gradient (HOG) features from badminton videos and employed a support vector machine (SVM) to classify a player's stroke into six types (clear, drive, drop, lob, smash), which resulted in 83.33% average accuracy. Srivastava et al. [16] developed a sports analytics engine based on an IMU to detect the tennis shot with a modified Pan-Tompkins algorithm, and proposed a time-warping based hierarchical shot classifier by using Dynamic Time Warping (DTW) at the first level (forehand, backhand and serve) and

Quaternion Dynamic Time Warping (QDTW) at the second level (slice and non-slice). The accuracy at DTW and QDTW were 99.6% and 90.7% for professional players, 99.3% and 86.2% for novice players. With CNNs, Bearman et al. [1] addressed the human joint location as a regression problem and used weight initialization from a trained AlexNet to classify human activity into 20 categories with the accuracy of 80.51%. In our work, a neural network including two hidden layers is utilized to train the racket pose trajectories and classify them into five types to achieve an accuracy of 98.7% on strokes of the same player.

3 Approach

3.1 Racket Detection

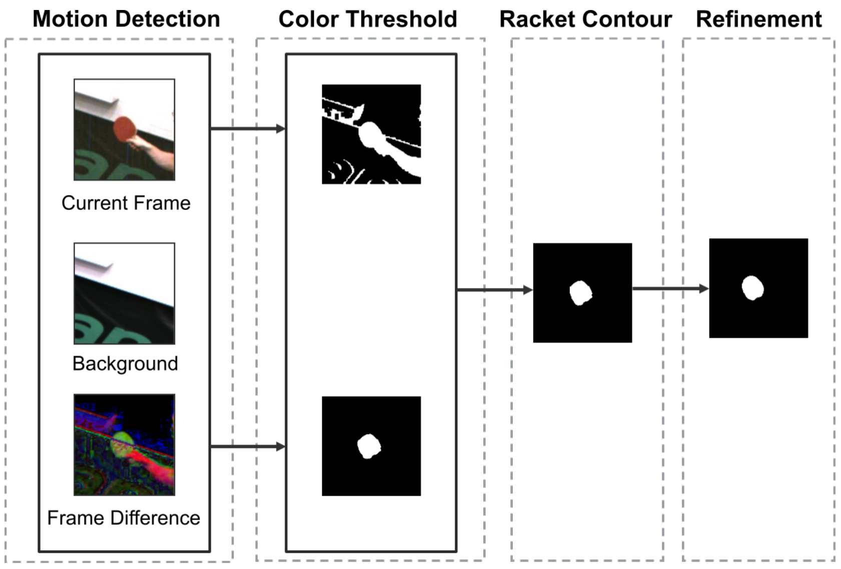

Fig. 2. Racket detection process with dynamic window from the right camera. *Motion detection*: subtract the background from current frame. *Color Threshold*: compute the binary image from RGB space. *Racket Contour*: bitwise AND operation from previous step. *Refinement*: bucket fill results.

To lower the impact on lighting variations, we choose the HSV color space instead of RGB and adopt the color thresholding algorithm similar to [18] with different boundary values to detect the red side of the racket. Multiple features of the racket are fused to extract the whole racket contour, like area and aspect ratio. Fig. 2 illustrates the pipeline of racket detection in the right camera, which includes four steps.

We primarily find the moving objects using a static frame difference method by subtract the background from current frame. The lighting between current frame and background is slightly different because we use the auto-exposure mode that dynamically adjusts parameters including gain, shutter time and white balance. Performing thresholding and morphology operations , we can get the binary images in the *Color Threshold* step resulting in the racket contour processed by bitwise AND operation.

Considering the property of the racket contour, we can determine it based on the following conditions:

$$200px \leq Area \leq 3000px \tag{1}$$
$$1/3 \leq AspectRatio \leq 3 \tag{2}$$
$$0.3 \leq AreaExtent \leq 1 \tag{3}$$

where *Area* is the contour area in pixels. *AspectRatio* is the contour aspect ratio of the minimal containing up-right bounding box. *AreaExtent* is the ratio of *Area* to the bounding box. The contour with the largest area satisfying the condictions is chosen. Its center is used to triangulate the racket's center 3D position.

Once the racket is recognized in both current and previous frames, we first predict the position of the racket in the next frame by adding the current position with the position difference of the last two frames. Then, we exploit a region of interest (ROI) around the predicted position to crop the full image into a dynamic window in order to accelerate the detection process. Secondly, a multithreading technique supplied by C++ is used to execute image processing concurrently for the left and right camera images. The third acceleration method called bucket fill is applied to find a connected component spreading from the seed point until the color value is out of specified range computed as follows:

$$C(x,y)_H - L_H \leq C'(x,y)_H \leq C(x,y)_H + U_H \tag{4}$$
$$C(x,y)_S - L_S \leq C'(x,y)_S \leq C(x,y)_S + U_S \tag{5}$$
$$C(x,y)_V - L_V \leq C'(x,y)_V \leq C(x,y)_V + U_V \tag{6}$$

where H, S, V are the components from HSV model. $C(x,y)_H$ is the H component value at the seed point (x,y). $C'(x,y)_H$ is the repainted H component domain presenting the new racket contour. L or U is maximal lower or upper color difference between the seed point and one of its neighbours. Fig. 2 shows that bucket fill yields better results than color thresholding with a runtime decreasing from 6.5 ms to 2 ms.

3.2 Racket Matching

When the 2D pixel coordinates of the racket's center are available from the two cameras, we can reconstruct these points as the racket 3D position by triangulation. The 3D orientation can be defined as the unit normal vector of the racket plane. Matching the left and right contours directly is difficult because of their random and uncertain shapes.Therefore, we want to find some corresponding feature points on the edge of the racket to recover the 3D plane in point-normal form.

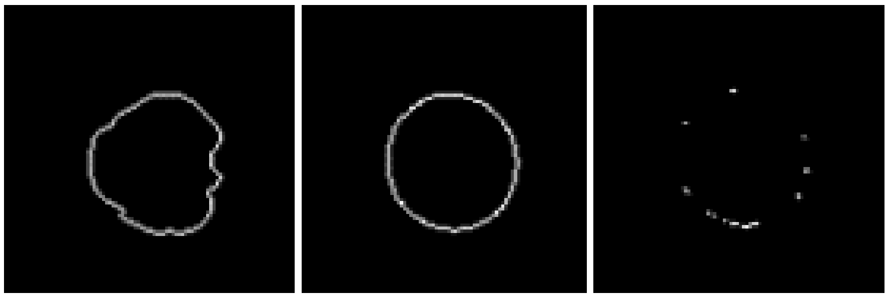

Fig. 3. Features (c) extraction from the intersection area between contour (a) and ellipse (b).

Features Extraction Since the racket plane is low-textured and the two cameras are far away from each other, features detection and description is difficult. The strong edge around the racket contour is used to extract the feature points in the undistorted images, which will not be rectified due to wide baseline and large rotation angle.

We approximate the edge by three ellipse fitting methods supported by [3] including normal least squares (LS), Approximate Mean Square (AMS) and Direct least square (Direct) aimed at finding the best one which has the largest degree of overlapping D between the edge and ellipse formulated as following:

$$D = \frac{N_{overlapping}}{N_{edge}} \tag{7}$$

where $N_{overlapping}$ and N_{edge} are the pixel numbers of the intersection area and the edge. We calculate D for a sequence of images and show the comparison in Table 1. The direct method is adopted for ellipse fitting because of its performance. Then, we choose the points on the intersection area as feature points, shown in Fig. 3.

Table 1. Comparison of ellipse fitting models.

Methods	LS	AMS	Direct
Degree	53.77%	56.60%	58.33%

Stereo Matching Next, we do not intend to match the two sets of feature points to each other, but find the corresponding points in another contour's edge. Depending on the epipolar geometry, we can narrow down the choice of candidates of corresponding points on the epipolar line. The points lying on both edge and epiline are the potential corresponding points of the feature points. Fig. 4 gives an example where P_R and P'_R are the intersection of edge and epiline related to the left point P_L. By means of the racket size, we can find the correct corresponding point from these two candidates described as:

$$P_1 = Triangulate(P_L, P_R) \tag{8}$$

$$P_2 = Triangulate(P_L, P'_R) \tag{9}$$

$$75 \leq |Pos - Center| \leq 90 \quad Pos \in [P_1, P_2]. \tag{10}$$

Here 75 mm and 90 mm are the length of the minor and major semi-axes. $Center$ is the 3D position of the racket. Therefore the inequality should be satisfied for a correct edge point. The algorithm chooses the point having the shortest distance by ($||Pos - Center| - \frac{75+90}{2}|$).

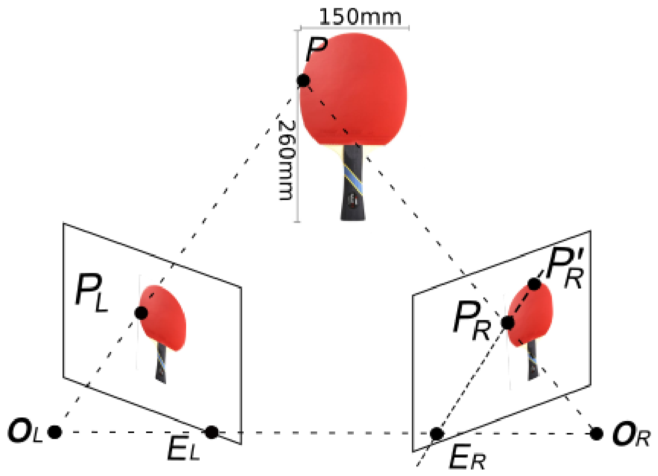

Fig. 4. Finding the potential candidates P_R and P'_R in the right camera corresponding to P_L. O_L and O_R are the optical centers of the cameras lenses. The epipolar line in the right camera passes through the epipole E_R, the image points P_R and P'_R.

Outliers Removal The feature matching method aforementioned can produce many corresponding pairs consisting of inliers and outliers. Because of these pairs on same surface, a homography matrix H for removing outliers can be derived as a 3x3 matrix but with 8 DoF estimated by:

$$s \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{11}$$

where s is a scale factor. $[x_i, y_i]$ and $[x'_i, y'_i]$ are the i^{th} pixel coordinates from left and right cameras. According to this transformation, we can minimize the re-projection error function after projecting points from one image into another given by:

$$\sum_i (x'_i - \widehat{x}_i)^2 + (y'_i - \widehat{y}_i)^2 \tag{12}$$

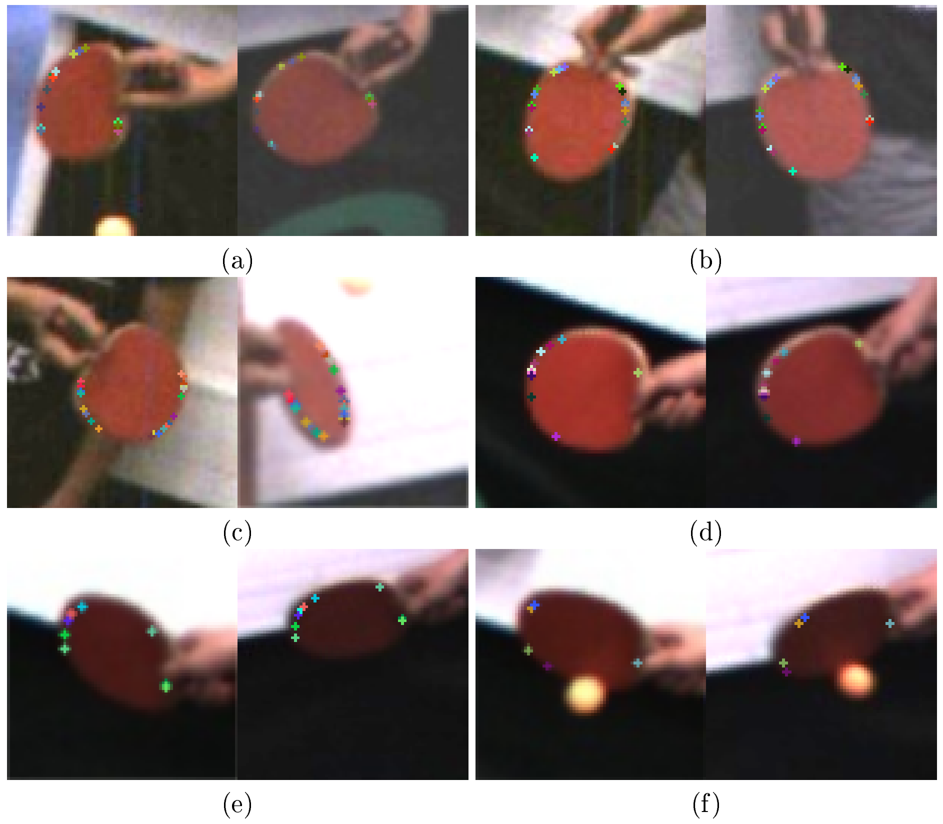

Fig. 5. Feature matching results including six examples in the left and right cameras. The grip type in (a)-(c) is penhold grip. (d)-(f) use the shakehand grip. The corresponding pairs are labeled with same color in order to distinguish the correct pairs.

where $\widehat{x_i} = \frac{h_{11}x_i+h_{12}y_i+h_{13}}{h_{31}x_i+h_{32}y_i+1}$, and $\widehat{y_i} = \frac{h_{21}x_i+h_{22}y_i+h_{23}}{h_{31}x_i+h_{32}y_i+1}$. They are the reprojected image coodinates.

However, using the whole pairs for matrix estimation will lead to a poor result. We utilize the Random SAmple Consensus (RANSAC) [7] to estimate the homography matrix by randomly selecting different subsets of the corresponding pairs and select the subset with the minimal re-projection error. Here, the outliers will be removed if the reprojection error is more than 3 pixels.

The final matching results are shown in Fig. 5 including 3 penhole and 3 shakehand types. Each corresponding features pair in the left and right cameras is labeled with the same color to be distinguished clearly.

Plane Fitting Reconstructing the corresponding pairs by triangulation, we can get a series of 3D points $[x_i, y_i, z_i]^T$ that can be used to estimate the equation of the racket plane $ax + by + c = z$. The centroid of these points is defined by the 3D racket center. The normal vector $[a, b, c]^T$ is described as:

$$\begin{bmatrix} x_0 & y_0 & 1 \\ y_1 & y_1 & 1 \\ & \cdots & \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ \cdots \\ z_n \end{bmatrix} \tag{13}$$

This can be written in the form $AX = B$. A common method to solve X is Singular Value Decomposition (SVD) by which A is decomposed as:

$$A_{n\times 3} = U_{n\times n} S_{n\times 3} V^T_{3\times 3} \tag{14}$$

where U and V are orthogonal matrices, S is a diagonal matrix, and n is the number of corresponding pairs. Then, the last column of V indicates the value of normal vector $[a, b, c]^T$. Normalizing this vector, we can get the unit norm vector representing the racket's orientation. We measure the processing time for racket detection and matching shown in Fig. 6. The stereo matching needs around 8 ms. The total time for racket pose estimation needs about 10 ms, which means we can estimation the racket 6D pose at 100 FPS.

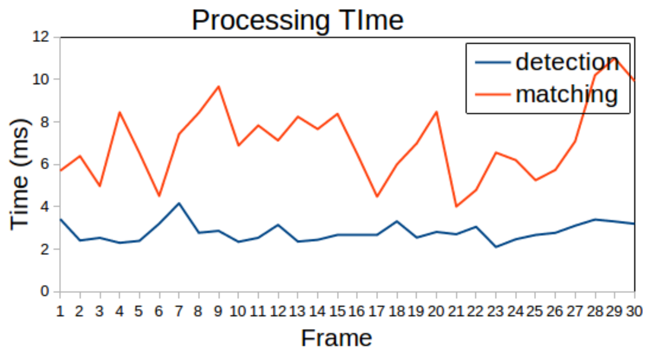

Fig. 6. Processing time for racket detection and matching.

3.3 Tracking

Tracking the racket pose offers two advantages. We can use the estimated pose when there is an occlusion or the racket is disappearing. Also it can provide a much smoother estimation of the racket pose. In this paper, we employ a discrete Kalman filter that is very efficient and powerful for estimating the pose $[x, y, z, a, b, c]^T$. We define the racket state X_t with 15 variables:

$$X_t = [x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \ddot{x}_t, \ddot{y}_t, \ddot{z}_t, a_t, b_t, c_t, \dot{a}_t, \dot{b}_t, \dot{c}_t] \tag{15}$$

A simple motion model is used to compute the next expected state X_{t+1}:

$$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \dot{\boldsymbol{p}}_t * \Delta t + \frac{1}{2}\ddot{\boldsymbol{p}}_t * \Delta t^2 \tag{16}$$

$$\dot{\boldsymbol{p}}_{t+1} = \dot{\boldsymbol{p}}_t + \ddot{\boldsymbol{p}}_t * \Delta t \tag{17}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \dot{\boldsymbol{\theta}}_t * \Delta t \tag{18}$$

where $\boldsymbol{p} \in [x, y, z]$ and $\boldsymbol{\theta} \in [a, b, c]$. Then, we can project the next state and error covariance ahead from current time and update them with current measurement. From Fig. 7, we note that in 30 frames, the estimated pose appears considerably smoother than the original one without Kalman filter.

3.4 Classification

Realizing the exact pose trajectory is not possible for humans, but people can still play table tennis really well due to their ability to recognize different stroke types. For robots, it is important to know not only what the exact pose is, but also which strike type is generated. There are many different types of stroke, but we can divide them into five basic categories: 1) *Counter Hit.* It is used to stop an aggressive, attacking stroke from your opponent by moving the racket and keeping it at the same angle. 2) *Left Spin.* It will be imparted when the racket moves to the left, which make the ball to bounce off in the same direction. 3) *Right Spin.* It is the opposite of left spin. 4) *Top Spin.* It is produced by starting the racket below the ball and hitting the ball in an upward and forward direction, which causes the ball to jump forwards after bouncing off the table and the opponent need to return with the racket face closed. 5) *Back Spin.* It is the opposite of top spin, with a downward stroke of the racket. If the opponent does not reply with a back spin or a strong top spin himself, the ball will drop down and into the net.

We stored the previous 30 frames to extract the trajectories of the racket pose once the ball flying towards robot is detected. To distinguish which spin type these trajectories belong to, we created a classifier based on a neural network containing two operations and two hidden layers able to predict in testing set:

Flatten operation The input shape is 30×6, which means each frame from the previous 30 frames includes six values (x, y, z, a, b, c). This layer converts the 30×6 matrix into a 1D feature vector 1×180 used in the artificial neural network (ANN) classifier. To simplify the dataset and make training more robust, we use the relative position to the last position in the 30^{th} frame instead of the absolute value, and normalize them into unit vectors.

Dense layer It is also called fully connected layer and first performs a linear operation in which every neuron from the previous layer is fully connected to this layer by a weight matrix *kernel* as following equation:

$$output = ReLU(input \cdot kernel + bias) \tag{19}$$

where the shape of *output* adopted in this paper is 128-dimensional. As activation function *ReLU* (Rectified Linear Unit) is used to introduce non-linearity. *bias* is a bias vector created by this layer.

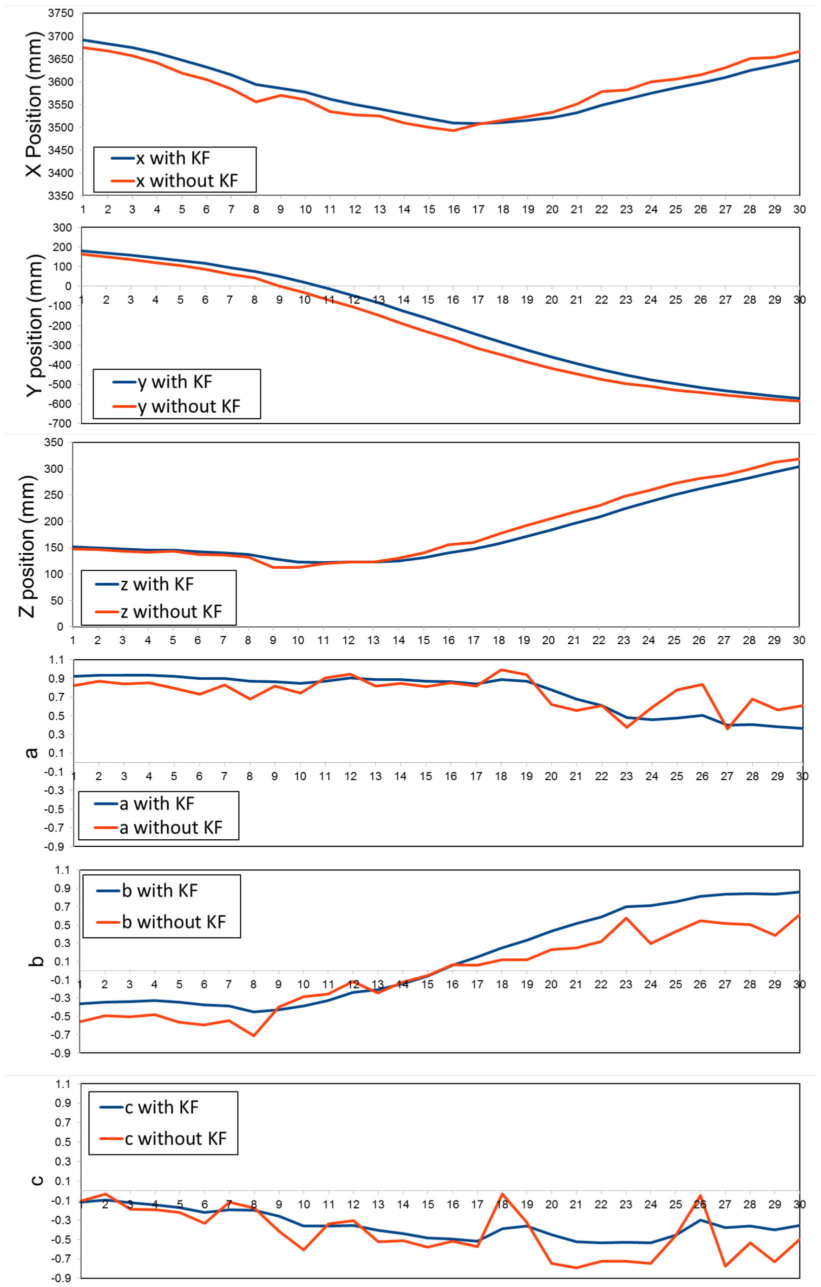

Fig. 7. Kalman filter tracking in 30 frames for racket position (x, y, z) and orientation (a, b, c).

Dropout operation By randomly setting a *rate* of input units to zero during the training phase of this set of units, we can reduce the over-fitting of training data. Here, *rate* is assigned to 20%.

Dense layer This layer performs classification on input units into five categories. We choose the softmax function to activate the Dropout layer.

For each spin type, we recorded 200 videos to generate the racket trajectories and human pose, respectively. Among them, 80% of the dataset are used to learn the classification model, and the remaining 20% are used as test dataset. In training, we use the Adam optimizer and a loss function with sparse categorical cross-entropy. Then we can train the model for a specified number of epochs. To compare the classification difference of pose, position and orientation, we experiment with them, respectively. From Table 2, we can find the best performing is the 6D pose. The accuracy with 3D orientation is much better than the 3D position

Table 2. Classification accuracy comparison.

	6D Pose	3D Position	3D Orientation
Training Set	98.7%	51.58%	94.8%
Testing Set	98.2%	50.63%	93.6%

4 Experiments

In this section, we conduct two experiments to evaluate the performance of our proposed methods. All processes are executed on one host PC with an Intel i5-4590 CPU, 16GB RAM and a GeForce GTX 1050 Ti GPU.

We first use a pair of cameras facing the robot to detect the pose of the racket mounted at the robot end effector shown in Fig. 1, and compare it with the ground truth data read from the robot controller. Then, we adopt an existing 2D human pose estimation model, Convolutional Pose Machine [20], to extract human joints as feature points. We compare this deep neural network with the classified network presented before. The comparison results are shown in the following subsections.

4.1 Evaluation on the KUKA robot

We have already transferred the 3D coordinates from camera to robot by employing a least-squares fitting method with two 3D point sets in our table tennis robot system [18]. The tool coordinate system in the robot was moved from the end effector to the racket center. In our work, the unit norm vector u_T of the red side on the racket in tool coordinates is always $[-1, 0, 0]^T$ by the negative direction of the x axis shown in Fig. 8. Next, we transform this vector to robot coordinates (namely, world coordinates).

The values $[X, Y, Z, A, B, C]$ can be read from the KUKA controller, where X, Y, Z are the racket's 3D position and A, B, C are the Z-Y-X Euler angles.

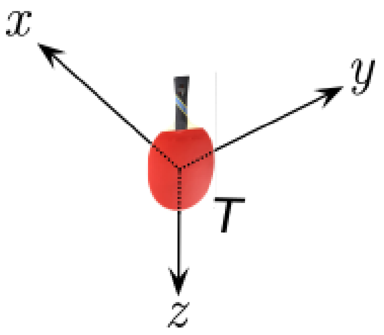

Fig. 8. The tool coordinate system.

The 3×3 rotation matrices about X, Y, Z axes are written as: R_X, R_Y, R_Z.

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos C & -\sin C \\ 0 & \sin C & \cos C \end{bmatrix} \tag{20}$$

$$R_Y = \begin{bmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{bmatrix} \tag{21}$$

$$R_Z = \begin{bmatrix} \cos A & -\sin A & 0 \\ \sin A & \cos A & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Then, the norm vector u_W in world coordinate is derived by:

$$u_W = R_Z R_Y R_X * u_T \tag{23}$$

Now, the $[X, Y, Z]$ and u_W are the ground truth data from the robot. To know the exact racket pose error, we manually control the robot to achieve 50 different poses with various position or Euler angles, and compute the racket pose from robot and cameras. Those angle between two norm vectors from the robot and cameras are defined as the orientation error. As shown from Fig. 9 , the position error is below 13 mm with an average of 7.8 mm and the orientation error is under 15° with 7.2° average value.

4.2 Comparison with human pose estimation

We directly apply the Convolutional Pose Machines (CPMs) to extract the human body keypoints including ear, eye, nose, neck, shoulder, elbow, wrist and hip (14 keypoints in total) in the left camera. A Kalman filter is used to track these keypoints. Human poses are calculated and stored as matrices to express which parts of the body are connected to each other. The visualization is shown in Fig. 10.

By means of the same dataset and classification approach with different input shape 30 frames $\times$ 14 keypoints, we can obtain the test accuracy of 98.4% , which is similar to the proposed method 98.7%.

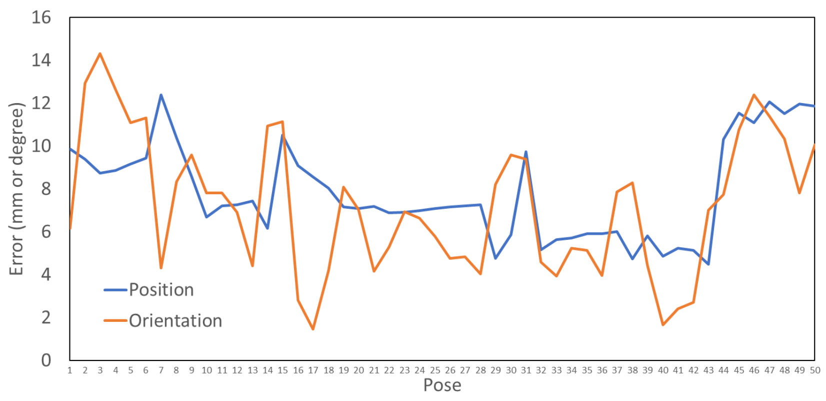

Fig. 9. Racket pose evaluation.

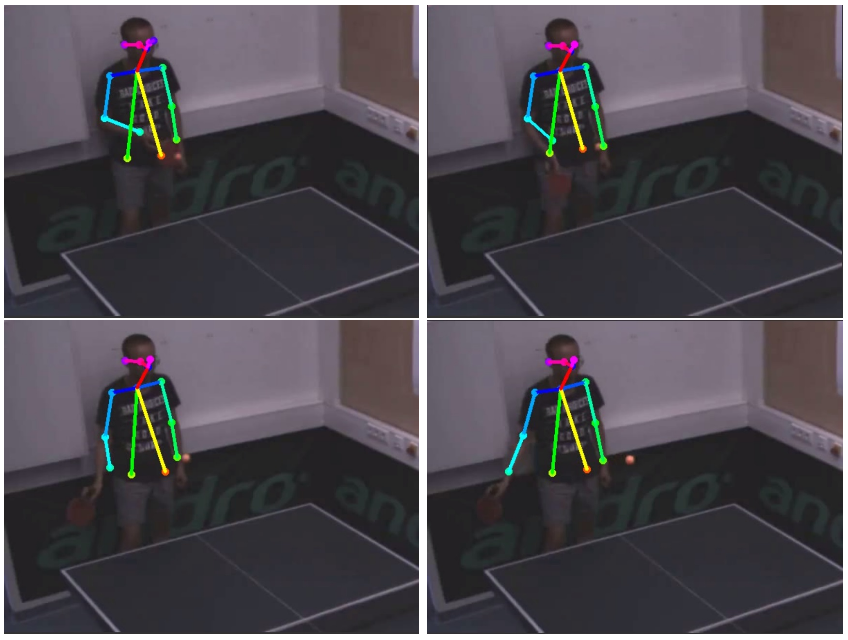

Fig. 10. Human pose estimation in image sequences.

However, CPMs has a crucial issue of hardware consumption. It can not satisfy the real-time requirement in table tennis. Meanwhile, it just provides the approximate pose information that can not be used to calculate the exact 3D position or orientation. In contrast, our proposed method can be run in 10 ms (100 FPS) and give the opportunities to train the robot having a human-like movement.

5 Conclusions

In this paper, we have presented a novel table tennis racket pose detection method based on stereo vision. Through the color and motion segmentation, we can extract the racket contours, then feed them into the proposed wide baseline stereo matching method to generate the 6D pose. With a multilayer perceptron (MLP) neural network, the pose trajectories can be classified into five kinds of spin types. Finally, two experiments are performed to evaluate the accuracy of pose detection and classification. In the future, we will teach the KUKA robot to mimic a human-like movement by imitation learning aiming to cope with various spin.

Acknowledgment

This work was supported in part by the Vector Stiftung and KUKA.

References

1. Bearman, A., Dong, C.: Human pose estimation and activity classification using convolutional neural networks. CS231n Course Project Reports (2015)
2. Blank, P., Hoßbach, J., Schuldhaus, D., Eskofier, B.M.: Sensor-based stroke detection and stroke type classification in table tennis. In: Proceedings of the 2015 ACM International Symposium on Wearable Computers. pp. 93–100. ISWC '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2802083.2802087, `http://doi.acm.org/10.1145/2802083.2802087`
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Chen, G., Xu, D., Fang, Z., Jiang, Z., Tan, M.: Visual measurement of the racket trajectory in spinning ball striking for table tennis player. IEEE Transactions on Instrumentation and Measurement **62**(11), 2901–2911 (Nov 2013). https://doi.org/10.1109/TIM.2013.2265471
5. Chu, W.T., Situmeang, S.: Badminton video analysis based on spatiotemporal and stroke features. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. pp. 448–451. ICMR '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3078971.3079032, `http://doi.acm.org/10.1145/3078971.3079032`
6. Elliott, N., Choppin, S., Goodwill, S., Senior, T., Hart, J., Allen, T.: Single view silhouette fitting techniques for estimating tennis racket position. Sports Engineering **21**(2), 137–147 (Jun 2018). https://doi.org/10.1007/s12283-017-0243-0, `https://doi.org/10.1007/s12283-017-0243-0`
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (Jun 1981). https://doi.org/10.1145/358669.358692, `http://doi.acm.org/10.1145/358669.358692`
8. Hiroshi Ohya, H.S.: Tracking racket face in tennis serve motion using high-speed multiple cameras. In: 2004 Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems (Sept 2004)
9. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 807–814 vol. 2 (June 2005). https://doi.org/10.1109/CVPR.2005.56
10. Huang, Y., Schölkopf, B., Peters, J.: Learning optimal striking points for a ping-pong playing robot. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4587–4592 (Sept 2015). https://doi.org/10.1109/IROS.2015.7354030

11. Lampert, C.H., Peters, J.: Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components. Journal of Real-Time Image Processing **7**(1), 31–41 (Mar 2012). https://doi.org/10.1007/s11554-010-0168-3, `https://doi.org/10.1007/s11554-010-0168-3`
12. Lane, R., Thacker, N.: Tutorial: Overview of stereo matching research. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester (1998)
13. Li, H., Wu, H., Lou, L., Kühnlenz, K., Ravn, O.: Ping-pong robotics with high-speed vision system. In: 2012 12th International Conference on Control Automation Robotics Vision (ICARCV). pp. 106–111 (Dec 2012). https://doi.org/10.1109/ICARCV.2012.6485142
14. Micusik, B., Wildenauer, H.: Descriptor free visual indoor localization with line segments. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3165–3173 (June 2015). https://doi.org/10.1109/CVPR.2015.7298936
15. Mülling, K., Kober, J., Peters, J.: A biomimetic approach to robot table tennis. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1921–1926 (Oct 2010)
16. Srivastava, R., Patwari, A., Kumar, S., Mishra, G., Kaligounder, L., Sinha, P.: Efficient characterization of tennis shots and game analysis using wearable sensors data. In: 2015 IEEE SENSORS. pp. 1–4 (Nov 2015). https://doi.org/10.1109/ICSENS.2015.7370311
17. Tareen, S.A.K., Saleem, Z.: A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). pp. 1–10 (March 2018). https://doi.org/10.1109/ICOMET.2018.8346440
18. Tebbe, J., Gao, Y., Rienitz, M.S., Zell, A.: A table tennis robot system using an industrial kuka robot arm. In: German Conference on Pattern Recognition. Springer, Stuttgart, Germany (2018), in press
19. Tulyakov, S., Ivanov, A., Fleuret, F.: Practical deep stereo (pds): Toward applications-friendly deep stereo matching. CoRR **abs/1806.01677** (2018)
20. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4724–4732 (2016)
21. Zhang, K., Fang, Z., Liu, J., Wu, Z., Tan, M.: Fusion of vision and imu to track the racket trajectory in real time. In: 2017 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 1769–1774 (Aug 2017). https://doi.org/10.1109/ICMA.2017.8016085
22. Zhang, Y., Xiong, R., Zhao, Y., Wang, J.: Real-time spin estimation of ping-pong ball using its natural brand. IEEE Transactions on Instrumentation and Measurement **64**(8), 2280–2290 (Aug 2015)