Table Tennis ball kinematic parameters estimation from non-intrusive single-view videos

Jordan Calandre
La Rochelle Université, MIA
La Rochelle, France
jordan.calandre1@univ-lr.fr

Renaud Péteri
La Rochelle Université, MIA
La Rochelle, France
renaud.peteri@univ-lr.fr

Laurent Mascarilla
La Rochelle Université, MIA
La Rochelle, France
laurent.mascarilla@univ-lr.fr

Benoit Tremblais
Université de Poitiers, XLIM
Poitiers, France
benoit.tremblais@univ-poitiers.fr

***Abstract*—The context of this research is the use of computer vision to assess the quality of sport gestures in non-intrusive conditions, i.e. without any body-worn sensors. This paper addresses the estimation of Table Tennis ball kinematic parameters from single-view videos. These parameters are important for analyzing effects given on the ball by the players, a key factor in the Table Tennis game. We introduce 3D ball trajectories extraction and analysis with very few acquisition constraints. To obtain ball to camera distance, the estimation of the apparent ball size is performed with a 2D CNN trained on a generated dataset. By formulating the problem of trajectory estimation as the solution of an Ordinary Differential Equation (ODE) with initial conditions, we can extract the ball kinematic parameters such as tangential and rotation speeds. Validation experiments are presented on both a synthetic dataset and on real video sequences.**

***Index Terms*—Video analysis, 3D trajectory reconstruction, Aerodynamics, Magnus effect, Sport Performance, Table Tennis**

I. Introduction

Recent works on action recognition from videos range from face micro-movements for emotion studies to macro-movements for sport performance analysis. In particular, fine-grained action recognition is being more and more investigated in recent years due to its various potential applications such as daily living care [5], video security and surveillance [6] or in sport activities [14]. The difference with coarse-grained action classification lays in the high intra-class similarity of the actions. Movements performed are often similar since they focus on one particular activity. Moreover, the background scene and manipulated objects cannot give much information on the action performed either, since videos are recorded in the same context.

The target application of our research is fine-grained action recognition and analysis in sports with the aim of improving athletes performances. Our case study is table tennis, and stroke detection along with a precise gesture analysis are of great interest for professional athletes and coaches. This task is difficult and is a fine-grained classification problem, because table tennis has a large number of stroke classes, with low inter-class variance. Consequently, all possible information should be extracted, from the performed movement itself to the trajectory given to the ball. To go a step further, the analysis of strokes can be improved by studying the effects given on the ball by the player. Strokes types are characterized by their rotation or translation speed, and hence these physical parameters can be used to quantify their efficiency.

In this article, we present for this purpose a method that reconstructs the 3D ball trajectory from a single camera view and extracts kinematics parameters from those trajectories. Most classical 3D motion analysis methods using motion capture [18] are far from real training conditions and can alter how the player performs his strokes. A more valuable alternative could be the use of multiple cameras [11], [15], but these devices remain complex and costly to use in a standard sports hall or outdoors. Our proposal focuses on the study of the ball trajectory, with special care on the rotation speed estimation, which is a clue of the effect given on the ball. Indeed, in most ball sports, when a player hits the ball, it starts to spin on itself which potentially modifies its trajectory. In the special case of Table Tennis, due to the small size and low mass of the ball compared to its speed (translation and rotation), the effect of this rotation is crucial for the players. It exceeds the effects that can be given to tennis balls or baseballs, and the use of spins of varying intensity is a key factor for success in a match.

Using a single high-speed camera, without depth-map or body-worn sensors, we build a CNN to extract the apparent ball size in pixels even with motion blur. Combined with precise camera calibration using table corners, it is possible to compute the ball retro-projection. We overcome many limits of our previous approach [2]: by improving the 3D ball estimated position, we are able to increase the camera field of view. This way the whole table can be acquired, leading to a more precise height estimation during calibration. As the whole trajectory can be captured, we can now handle the complete ball trajectory, including rebounds on each side of the table. We also modified the way kinematic parameters are extracted. We previously used the Euler's Method to simulate the trajectory, which made the final curve-fitting problem complex for estimating the translation and rotation speed. This was unsatisfactorily solved with grid-search using different padding to be fixed empirically. Here, we reformulate the problem of the trajectory estimation as the solution of an Ordinary Differential Equation (ODE) with initial conditions, leading a more efficient estimation of the trajectory. To simulate realistic trajectories on a synthetic dataset, we also include bounces equations.

The remaining of the paper is organized as follows: in section II, related works are presented for ball trajectory analysis. Section III exposes our method, from ball detection and tracking to the extraction of kinematics parameters. In section IV, a dataset with sportsmen specially created for this work is presented, along with a synthetic dataset. Experiments and Results are presented in section V, and conclusions are drawn in section VI.

II. Related Works

As opposed to other sports like volleyball [3] where the ball is pushed with the arms, the ball in Table Tennis is hit by a surface, creating a fast rotation of the ball on itself. The induced trajectory can not be precisely modeled as a parabola, which stands when only gravity and drag effect occurs. When the ball spins, the difference of air pressure on the upper and lower sides of the ball causes a modification of its trajectory, called the Magnus force [1]. This effect strongly affects a table tennis ball [12], [16], [20]. This force, orthogonal to the velocity vector, induces *topspin* or *backspin*, and can induce a strong *sidespin* for specific strokes, especially for serves. For the considered stroke types (section IV-A), it is reasonable to assert that between two strokes, the ball trajectory lies in a 2D plane [12], [21].

To model the whole trajectory after the ball impact on the racket, two models are used: a flight model and a rebound model.

1) Flight model: The dynamic model that incorporates air drag and spin effects when the ball is in midair is nonlinear.

Given the gravity vector $\mathbf{g} = (0, 0, g)^T$, center of mass coordinates $\mathbf{b} = (b_x, b_y, b_z)^T$, the air drag constant C_D , the air lift constant C_L , and the angular velocity vector of the ball $\omega = (\omega_x, \omega_y, \omega_z)^T$, the flight model equation is given by:

$$\ddot{\mathbf{b}} = \mathbf{g} - C_D \|\dot{\mathbf{b}}\| \dot{\mathbf{b}} + C_L \omega \times \dot{\mathbf{b}} \quad (1)$$

2) Rebound model: The rebound model [17] for a ball of radius r, with incident vector $\dot{\mathbf{b}}_{in}$ and reflected vector $\dot{\mathbf{b}}_{out}$ is given by:

$$\dot{\mathbf{b}}_{out} = \mathbf{A}\dot{\mathbf{b}}_{in} + \mathbf{B}\omega \quad (2)$$

With μ the friction coefficient, ϵ the coefficient of restitution of the table, $\dot{\mathbf{b}}_\tau$ the tangent velocity when the ball hit the table, and $\alpha = \mu(1-\epsilon)\frac{\dot{b}_z}{\|\dot{\mathbf{b}}_\tau\|}$ the sliding parameter (see [17]) :

$$A = \begin{bmatrix} 1-\alpha & 0 & 0 \\ 0 & 1-\alpha & 0 \\ 0 & 0 & -\epsilon \end{bmatrix}, B = \begin{bmatrix} 0 & \alpha r & 0 \\ -\alpha r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

III. Proposed Method

The extraction of kinematic parameters of the ball from 2D images is composed of several steps. The first step is to determine camera calibration matrices. After a precise calibration for both intrinsic and extrinsic matrices, a ball detection is performed combining background subtraction and color filters. The ball is tracked along its whole trajectory using the CSRT [13] tracker. The CSRT tracker works by training Discriminative Correlation Filters with compressed features (Colornames and HoG) for channel and spatial reliability. Slower than KCF [10], CSRT tends to be more accurate and adapts well to scale, deformation and rotation changes.

This step enables to reduce the analysis to a region of interest along the trajectory (see Fig. 1). A 2D to 3D ball retro-projection is done using the calibration matrix and the estimation of the ball size in the image obtained using a CNN (section III-D). After the 3D trajectory extraction, we reformulate the problem of the trajectory estimation as the solution of an Ordinary Differential Equation (ODE) with initial conditions to extract kinematics parameters.

A. Camera Calibration

Two matrices are used for calibrating the scene prior to recording the sequences: the intrinsic matrix contains the focal length, image sensor format, principal point and is obtained using checkerboard patterns [23]. The extrinsic matrix contains information about the scene and how to transform an object from the 3D world to the 2D camera coordinates. System coordinates are relative to the table. This matrix can be obtained by manually annotating the corners of the table and the ground (see Fig. 1), with "Perspective-n-point" type algorithm [9] and Levenberg-Marquardt optimization method.

B. Ball Detection

Even with a high frame rate (in our case 240 frames per second), the ball is often perceived as a blurred and ellipsoidal shape (see Fig. 4). As the ball to camera distance is high compared to our previous work [2], the `Detectron2` framework [8] was not robust enough to detect the ball in fast motion. We overcome this issue by combining background subtraction to remove static objects and colorimetric filters to keep pixels with values close to the ball (white/grey). Once detected, a CSRT [13] tracker is initialized and is able to track the ball in all the tested sequences.

C. Ball trajectory segmentation

The whole trajectory of a table tennis ball can be represented by a set of piecewise curves (altered parabolas). Each curve is connected to its neighbors by a bounce of the ball on the table or by its impact on a racket (see Fig. 1). The impact of the ball on the racket or on the table usually corresponds to an abrupt change of direction. We use this information to detect when the player hits the ball (horizontal change of direction) or when the ball hits the table (vertical change of direction). Each part of a trajectory can be analyzed separately (Fig. 1).

D. Ball size estimation

Knowing the ball positions in the image is not sufficient to estimate its 3D positions, as the ball to camera distance is unknown. Intuitively, when the ball gets closer to the camera in the 3D scene, its pixel size in the images increases, and decreases when it moves away from it. The size estimation in pixels of the ball is then crucial, as it provides information on its distance from the camera.

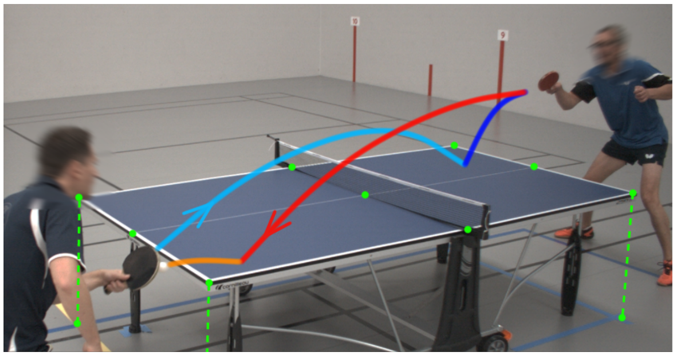

Fig. 1. Piecewise trajectory of a ball during a match. The ball moves in the direction of the arrow. Green dots indicates reference points used for the 3D to 2D projection matrix.

Given H the real size of the ball, f the focal length of the camera obtained at the calibration step, and h the apparent size of the ball in pixels, the ball to camera distance D is computed using a simple homothety:

$$D = f \times H/h \tag{4}$$

Knowing distance D, the intersection between the line passing through the center of the ball and the camera optical center allows to position the ball in the 3D real-world space by applying the extrinsic calibration matrix.

The ball is perceived as blurred due to the high speed and motion blur. An error in the ball size in pixel estimation introduces a consequent error on the ball to camera distance. In [2] a convolutional neural network was designed to extract the ball size precisely.

With the new acquisition conditions, and camera location further away from the table, the previous CNN model was too imprecise and results in a higher error on the ball 3D position estimation. To cope with that, we have generated a new synthetic dataset with the Blender software (section IV-B), and increased the motion blur generation to have a more robust estimation during the learning phase. A new block of convolution/relu/max-pooling was also added and increases the obtained accuracy.

The architecture of the proposed CNN is summarized in Table I. To temporally regularize the ball size estimation, this network uses 5 consecutive cropped areas of 64×64 pixels.

E. Kinematics parameters estimation

With the assumption that the trajectory of a stroke lies in a 2D plane, we perform a planar regression on the estimated 3D ball positions. In our previous work, we used Euler's Method along with grid search minimization with two successive coarseness to fit the target curve. In this work, it is replaced by an Ordinary Differential Equation (ODE) solver based on the Nelder-Mead method [7]. Given initial conditions (initial position and regression plane), we can fit our flight model (Eq. 1) to our estimated 3D points. The solver is applied in two sequential steps: the first one estimates the initial velocity vector (magnitude and direction) of the ball then the second one estimates its rotation speed. This strategy was found to be more precise than a joint estimation of the two parameters.

TABLE I
NETWORK ARCHITECTURE FOR BALL SIZE ESTIMATION

Input Size	Operator	In-channels/Out-channels
64x64	Conv 3x3	5/32
64x64	Leaky Relu	32/32
64x64	MaxPool 2x2	32/32
32x32	Conv 3x3	32/64
32x32	Leaky Relu	64/64
32x32	MaxPool 2x2	64/64
16x16	Conv 3x3	64/128
16x16	Leaky Relu	128/128
16x16	MaxPool 2x2	128/128
8x8	Conv 3x3	128/256
8x8	Leaky Relu	256/256
8x8	MaxPool 2x2	256/256
8x8	Flatten	256/-
16384	FullyConnected	-
4096	Relu	-
4096	Dropout	-
4096	FullyConnected	-

IV. DATASETS USED

For performing action recognition, we recorded stereo high-speed sequences with sportsmen in a sports hall. Due to the lack of ground truth for the translation and rotation speed of the ball, a synthetic dataset was also created to evaluate the estimation of the ball size and the extraction of kinematics parameters. The synthetic dataset was also used for training the CNN in section III-D.

A. Real world dataset with sportsmen

Our dataset was created with players from the Table Tennis Club of La Rochelle (France). Sequences were filmed by two high-speed synchronized cameras for obtaining a ground truth on the 3D locations of the ball by 3D stereo reconstruction. Each video contains one stroke type focused on one player and has been manually labeled. The sequence starts when the opponent hits the ball. The filmed player performs a stroke, and the sequence ends when the opponent hits the ball again. A single stroke video can have between 250 and 350 frames depending on the translation speed of the ball.

Compared to our previous dataset, the distances between the cameras and the table have been greatly increased in order to observe the whole trajectory and the bounces on the table.

Our dataset is composed of three strokes types (see Fig. 2). Those strokes have typical trajectories that differ from each other by translation and rotation speeds. Table tennis rackets usually have a surface made of rubber that increases the effect of spinning given to the ball. The 3 selected stroke types are `Forehand Top Spins`, `Forehand Push` and `Forehand Counter-attack`.

A `Top Spin` is an offensive stroke, obtained when the player brushes the racket against the ball using an upward stroke action. The ball accelerates and rotates at a fast speed.

A `Push` is a defensive stroke. As opposed to the `Forehand Top Spins`, it has backspin, which is obtained by starting the stroke above the ball, and brushing it in a downward motion.

A `Counter-attack`, or `Drive` is used when an opponent is making an aggressive offensive stroke. The rotation speed and translation speed are slower than a `Top Spin`.

Video samples of such strokes can be seen on Figure 2, and more examples are presented online[1].

Fig. 2. Extracted frames of a `Top Spin` (left), a `Counter Attack` (center) and a `Push` (right)

B. Synthetic sequences

Due to the absence of table tennis datasets with ground truth on ball trajectories and speed (translation and rotation), we have generated a dataset using the Blender software [4]. The approach is similar to our previous work [2]. The camera position was set on the real camera position, and obtained using the extrinsic camera matrix (section IV-A). Our blender camera matrix uses the same intrinsic camera as the real-life dataset to have a similar field of view.

To match real-life physics, we use 3D models for the table tennis ball and the table, and generate motion blur with the `Cycles` rendering tool[2] (see Fig. 3). Compared to our previous dataset, motion blur, light and shadows are now much more realistic.

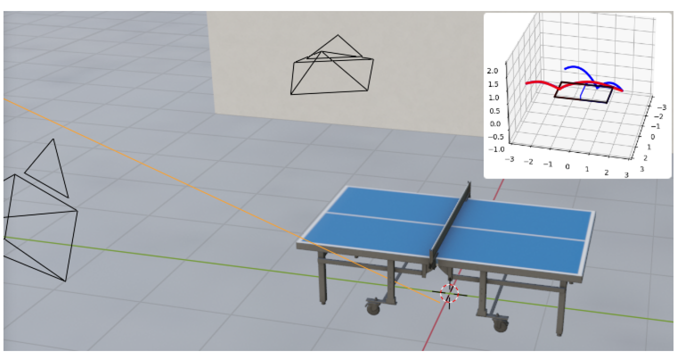

Fig. 3. Synthetic scene generation, and camera positioning

Due to the fact that the bounces are now visible on both sides of the table, we also introduce a physical model for the bounce (Equation 2).

The dataset is composed of 200 videos (representing a total of 60.676 frames). To generate a full exchange between two players, initial 3D position and hit angle are chosen randomly. The ball translation speed is chosen randomly using a uniform

[1]https://vimeo.com/jordancalandre

[2]https://docs.blender.org/manual/en/latest/render/cycles/index.html

Fig. 4. Real motion blur (left) and generated frame (right)

distribution, from 1.389 to 16.666 meters per second, and the rotation speed from -15 to 70 rotation per second (rps). The ball flight and bounces are simulated using equation 1 and equation 2. Checks are done to guarantee a valid stroke (the ball should not touch the net, should hit the correct part of the table, and bounces only once on the opponent side of the table).

For each stroke, kinematics parameters (translation and rotation speed) are saved. For each frame, the precise apparent size of the ball in pixels is also saved, and computed using equation 4 from the exact 3D position of the ball.

V. Experiments

We have tested the different steps that make up our method. The first step evaluates the precision of the 3D ball position estimation using our synthetic dataset. The second step is the validation of the extracted parameters, obtained by fitting a trajectory model. Finally, we used our CNN and fine-tuned it on real sequences to do an action recognition task based on the translation speed and rotation speed of the ball.

A. CNN for 3D ball position estimation

The estimation of the 3D ball position is tested on our generated dataset (section IV-B), as the ground truth of the ball 3D position is available for each generated frame. The dataset is composed of 133 new training sequences and 67 new testing sequences. Implementation was done using the PyTorch framework [19], running on a GPU GeForce GTX 1070 with 48GB RAM and an Intel i7-7700HQ processor.

The ball is detected and tracked as presented in section III-B. To do a temporal regularization of the ball size estimation, two frames before and after the current one are cropped, with size 64×64 pixels. This stack of five frames is used as input to our CNN (see Table I). The whole trajectory is used to train the network (corresponding to four segments on Fig. 1). The CNN output is compared with the theoretic ball size obtained using equation 4 when the real 3D position is known.

The estimated 3D position is then obtained by computing the intersection of the camera-to-ball ray and the sphere of radius D.

For each testing sequence, the 3D ball position estimation is done after the player of interest has performed a stroke. As an example, in Fig. 1, the red curve underlines the trajectory between the first player's hit and the table impact, while the orange curve underlines the trajectory until the next player's hit. We obtained an average error of 5.9 cm for the `Top Spin`, 7.6 cm for the `Counter-attacks`, and 0.6 cm for the Push strokes. The average error for all strokes is 6.68 cm, representing a relative error of only 1.39% of the distance between the camera and the ball.

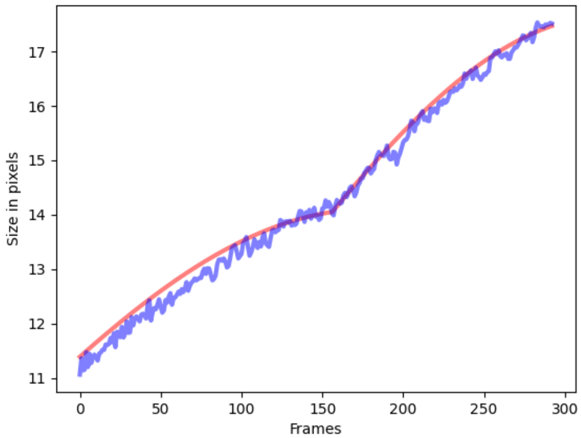

Fig. 5. Extracted ball size in pixels using proposed 2D CNN

Figure 5 shows a plot of the ball size estimation for a Top Spin stroke. The corresponding estimated 3D trajectory is shown in Figure 6.

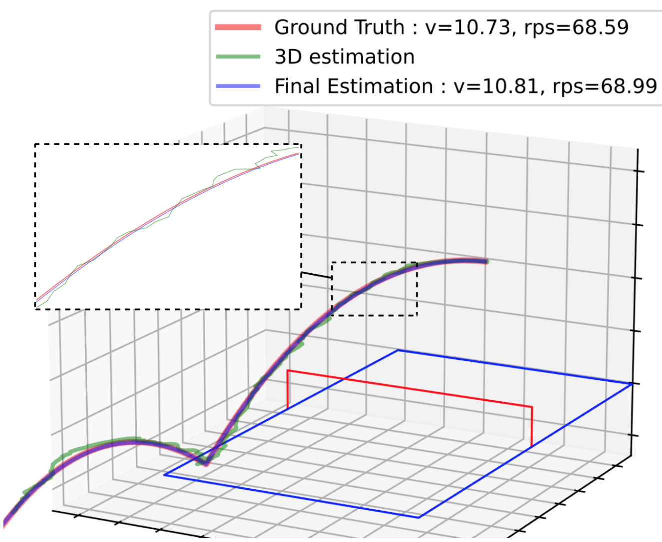

Fig. 6. Comparison between real ball size trajectory, estimated 3D positions, and estimated kinematic parameters

B. Kinematic parameters estimation

The goal of our experiments is to extract the kinematic parameters of the ball trajectory which are the translation and rotation speeds. They are indeed directly linked to the effect given to the ball by the players. We made the assumption that the estimation of forces applied on the ball is restricted to a 2D plane. The regression plane is obtained from the 3D positions using a linear regression.

In our previous work [2], the equation was fitted to the 3D positions using two grid search minimization and two successive coarseness. This estimation was time-consuming and nowhere near real-time. To increase the fitting process, we are considering the trajectory as a solution of an ODE (equation 1). To solve this equation and determine the best trajectory, initial values on the velocity vector and the position are needed. These initialization is obtained for the position and velocity using a degree 4 polynomial approximation on the 2D plane. The rotation speed is initialized at 10 rotations per second (rps), and the translation speed at 10 m/s, which are plausible values for a Counter-attack stroke.

By using the Eq. 2, the fitting is done on the complete trajectory between two players hits, including the table impact (represented by the red and orange segments in Fig. 1).

To avoid impossible translation or rotation speed values during fitting, translation speed are fixed in the range [1.388m/s, 16.666 m/s] (corresponding to [5km/h, 60km/h]) and rotation speed in the range [-15 rps, 70 rps].

Table II presents errors obtained between the ground truth and the estimated parameters for each stroke type.

TABLE II
AVERAGE ESTIMATED ERROR OF THE EXTRACTED PARAMETERS FOR EACH STROKE TYPE IN METRIC UNITS AND RELATIVE TO THE RANGE OF POSSIBLE VALUES

Stroke	Translation speed		Rotation speed	
	m/s	Relative (%)	rps	Relative (%)
Top Spin	0.052	3.405	3.932	4.629
Counter attack	0.024	1.570	2.752	3.324
Push	0.005	0.327	1.461	1.712
Global	0.027	1.767	2.715	3.194

Compared to our previous approach, the computational time is reduced by a factor of 3, from 89 to 27 seconds and kinematic parameters are more precisely estimated, even with a camera further away from the table. The time needed for the fitting of the trajectory was one of the main drawbacks of the grid-search approach. In the perspective to be helpful for players or coaches, performance measures should not rely on heavy computation.

C. Action recognition on the kinematic space

Ground truth for rotation and translation speed are not available on the real dataset introduced in section IV-A. However, the performed strokes are recorded with two synchronized cameras, which enables the computation of the 3D positions of the ball using triangulation. From these 3D positions and calibration matrices, we can deduce from equation 4 the size of the ball in pixels for each frame.

The 2D CNN pre-trained on the generated dataset (section V-A), is fine-tuned on the real dataset using 28 sequences. With the same approach as in section V-B, kinematic parameters can be extracted using only one camera.

The relevance of these extracted ball kinematic parameters has been tested for fine-grained action recognition on the real dataset. On these sequences, the stroke type has been labeled by expert knowledge, velocity or rotation speed of the ball are however unknown.

We tested the action classification on a total of 12 sequences. Considering three stroke types, we obtained an accuracy of 91.77% using a Naive Bayes Classifier [22] on the rotation and translation speed parameters. These parameters can be valuable not only for a classification but can also help players in their practice. For instance, a player performing `Forehand Top Spin` should focus on increasing the translation and rotation speed of the ball for a more efficient stroke.

VI. Conclusion

In this paper, we propose a method that enables us to estimate Table Tennis ball kinematic parameters from single-view videos. These parameters are important for analyzing effects given on the ball by the players, a key factor in Table Tennis games.

Contrary to most approaches for sport gesture analysis, which perform 3D reconstruction using stereo cameras or markers involving many practical constraints, we perform 3D ball trajectories extraction and analysis with very few acquisition constraints.

To obtain ball to camera distance, the estimation of the apparent ball size is efficiently performed with a 2D CNN trained on a generated dataset. Compared to our previous work [2], we have greatly improved the computational cost of the fitting process as well as the accuracy of the kinematic parameters estimation. Furthermore, these parameters appear highly correlated to our knowledge of the considered strokes.

This paper opens perspectives for the analysis of sports gestures in non-intrusive conditions, i.e. without markers: the kinematic parameters of the ball could be used as characteristics for action recognition tasks, performance indicators, or clues for automatic summaries of players' actions during a match.

In the future, we plan to use Deep Learning methods to analyze the trajectory as a whole, not based on frame-by-frame analysis. By simplifying the 3D reconstruction, we hope to reach near real-time computation time.

VII. Acknowledgment

We would like to thank Mathieu GUENIN and all the players of the Club Pongiste Rochelais, who have contributed to the building of the table tennis dataset.

The research is supported by the Region of Nouvelle Aquitaine through the CRISP project and Tannin2 by the MIRES, FR CNRS 3423, research Federation on Mathematics and Computer Science.

References

[1] Lyman J. Briggs. Effect of Spin and Speed on the Lateral Deflection (Curve) of a Baseball; and the Magnus Effect for Smooth Spheres. *American Journal of Physics*, 1959.

[2] Jordan Calandre, Renaud Péteri, Laurent Mascarilla, and Benoit Tremblais. Extraction and analysis of 3D kinematic parameters of Table Tennis ball from a single camera. In *ICPR 2020, 25th International Conference on Pattern Recognition (ICPR)*, Milano, Italy, January 2021.

[3] Hua Tsung Chen, Chien Li Chou, Wen Jiin Tsai, and Suh Yin Lee. 3D ball trajectory reconstruction from single-camera sports video for free viewpoint virtual replay. In *2011 IEEE Visual Communications and Image Processing, VCIP 2011*, pages 1–4. IEEE, 2011.

[4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[5] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, François Brémond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *ICCV*, pages 833–842. IEEE, 2019.

[6] Bharat Singh et al. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, pages 1961–1970. IEEE Computer Society, 2016.

[7] Fuchang Gao and Lixing Han. Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51:259–277, 05 2012.

[8] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, https://github.com/facebookresearch/detectron, 2018.

[9] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision, cup, cambridge, uk, 2003, vi 560 pp., isbn 0-521-54051-8. *Robotica*, 23(2):271–271, 2005.

[10] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *CoRR*, abs/1404.7584, 2014.

[11] Joongsik Kim, Moonsoo Ra, Hongjun Lee, Jeyeon Kim, and Whoi Yul Kim. Precise 3D baseball pitching trajectory estimation using multiple unsynchronized cameras. *IEEE Access*, 2019.

[12] Seiji Kusubori, Kazuto Yoshida, and Hiroshi Sekiya. The functions of spin on shot trajectory in table tennis. In *ISBS-Conference Proceedings Archive*, 2012.

[13] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Pierre-Etienne Martin, Jenny Benois-Pineau, Renaud Péteri, and Julien Morlier. Fine grained sport action recognition with twin spatio-temporal convolutional neural networks. *Multim. Tools Appl.*, 79(27-28):20429–20447, 2020.

[15] Shogo Miyata, Hideo Saito, Kosuke Takahashi, Dan Mikami, Mariko Isogawa, and Hideaki Kimata. Ball 3D Trajectory Reconstruction without Preliminary Temporal and Geometrical Camera Calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[16] T Miyazaki, W Sakai, T Komatsu, N Takahashi, and R Himeno. Lift crisis of a spinning table tennis ball. *European Journal of Physics*, 38(2):024001, dec 2016.

[17] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa. Modeling of rebound phenomenon of a rigid ball with friction and elastic effects. In *Proceedings of the 2010 American Control Conference*, pages 1410–1415, 2010.

[18] S. Noiumkar and S. Tirakoat. Use of optical motion capture in sports science: A case study of golf swing. In *ICICM*, pages 310–313, 2013.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[20] Ralf Schneider, Lars Lewerentz, Karl Lüskow, Marc Marschall, and Stefan Kemnitz. Statistical analysis of table-tennis ball trajectories. *Applied Sciences*, 8:2595, 12 2018.

[21] Lejun Shen, Qing Liu, Lin Li, and Haipeng Yue. 3D reconstruction of ball trajectory from a single camera in the ball game. In *Advances in Intelligent Systems and Computing*, 2016.

[22] Harry Zhang. The optimality of Naive Bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*, 2004.

[23] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1330 – 1334, 12 2000.