Extraction and analysis of 3D kinematic parameters of Table Tennis ball from a single camera

Conference Paper · January 2021

DOI: 10.1109/ICPR48806.2021.9412391

4 authors, including:

Jordan Calandre
La Rochelle Université
8 PUBLICATIONS **40** CITATIONS
SEE PROFILE

Laurent Mascarilla
La Rochelle Université
57 PUBLICATIONS **361** CITATIONS
SEE PROFILE

Benoit Tremblais
Université de Poitiers
89 PUBLICATIONS **410** CITATIONS
SEE PROFILE

Extraction and analysis of 3D kinematic parameters of Table Tennis ball from a single camera

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

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

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

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

***Abstract*—Vision is the first indicator for coaches to assess the quality of a sport gesture. However, gesture analysis using computer vision is often restricted to laboratory experiments, far from the real conditions in which athletes train on a daily basis. In this perspective, we introduce 3D ball trajectories analysis using a single camera with very few acquisition constraints. A key point of the proposal is the estimation of the apparent ball size for obtaining ball to camera distance. For this purpose, a 2D CNN is trained using a generated dataset that enables a reliable ball size extraction, even in case of high motion blur. The final objective is not only to be able to determine ball trajectories, but most importantly to retrieve their relevant physical parameters. Those kinematics parameters could help coaches and players to better analyze strokes with strong rotation effects during training sessions or matches. With a precise estimation of those trajectories, it is indeed possible to extract the ball tangential and rotation speed, related to the so-called Magnus effect. Validation experiments for characterizing table tennis strokes are presented on both a synthetic dataset and on real video sequences.**

***Index Terms*—Video analysis, 3D trajectory reconstruction, Magnus effect, Deep Learning, sport performance, Table Tennis**

I. Context

The amount of sport contents is steadily growing on both television and the Internet. Human pose estimation, as well as classification of sport activities have rapidly become popular areas of research in computer vision. Several datasets like UCF-101 [7] have emerged to determine the type of action in a video sequence. Great progress has been made with the advent of convolutional neural networks for video analysis [4], [18], which had led to a huge breakthrough in image classification.

In the field of computer vision, the fine-grained analysis of human actions or gestures is often confined to laboratory experiments, far from real conditions in which athletes train on a daily basis. The presented work aims at developing sport analysis tools that can be used in sports facilities and outdoor. In particular, the athletes are practicing in *ecological* conditions (*i.e.* natural conditions), without markers nor sensors that could interfere with their practice.

Our case study is table tennis, and we are working together with professional athletes and coaches to determine their needs. In particular, stroke detection along with a precise gesture analysis are of great interest to them. This task is difficult, because table tennis has a large number of stroke classes, with low inter-class variance. In order to classify these strokes, a publicly available dataset was created [11].

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 physical parameters from those trajectories.

Most classical 3D motion analysis methods using motion capture [14] 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 [8], [12], but these devices remain complex and costly to use in a standard sport hall or outdoors.

Our proposal focuses on the study of the ball trajectory, with a 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 on itself), 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. It is worth noting that this rotation is not that prominent in some other sports such as volleyball [2] where balls are pushed with the arms and not hit by a racket: in these cases, the ball trajectories can be reasonably approximated by parabolas.

In the following sections, we focus on physical parameters estimation of fast moving table tennis balls using a 2D to 3D approach. Firstly, a convolutional neural network is used to obtain accurate pixel size of the ball during its trajectory. It proved to be robust enough to properly operate even in presence of high motion blur. Secondly, this size estimation, along with camera calibration matrices, are used for recovering the 3D real-world coordinates of the ball from its image coordinates. Finally, the temporal succession of extracted ball positions generates piecewise 3D curves defined on intervals separated either by a table bounce, or by a racket impact. On each interval, knowing the physical forces applied on the

ball (gravity, Magnus effect and air drag effect), enables to compute relevant physical parameters such as rotation speed. These parameters are quantitative measures of the efficiency of a stroke and can provide a feedback on the athlete's performance.

II. 3D Ball Trajectory Estimation

The 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. The impact of the ball on the racket usually corresponds to an abrupt change of direction.

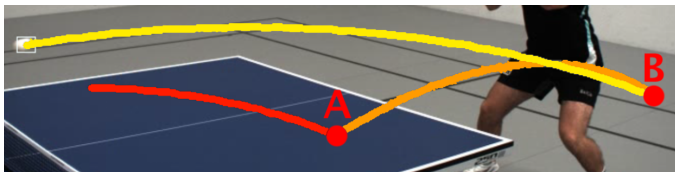

Fig. 1. Piecewise trajectory of a ball during a match

Figure 1 shows an example of a ball trajectory composed of three intervals: a first curve from the initial hit, off-camera, to the bounce on the table at point A; a second curve from point A to the impact on the player's racket at point B; a third curve from point B to a off-camera point. Point A is characterized by a change of direction along the y axis in the image frame and point B by an abrupt change of direction along the x axis. Three typical stroke types will be considered (see section IV-A), and we made the assumption that the ball lies in a plane between two strokes (equation 2). Contrary to [17], using a camera focused on half of the table does not allows to estimate the trajectory plane between two rebounds of the ball. The analysis of a stroke made by a player can be induced by the ball trajectory from the impact on his racket till its opponent's stroke. Each of the two parts of this trajectory can be analyzed separately (Fig. 1).

A. Forces and aerodynamics of the ball

Newton's second law of motion states that the net force $\mathbf{F}$ (the vector sum of all forces acting on the ball during its trajectory) is equal to the mass m of the ball times its acceleration $\mathbf{a}$:

$$m\mathbf{a} = m\frac{d\mathbf{V}}{dt} = \mathbf{F}_G + \mathbf{F}_A(\mathbf{V}) \tag{1}$$

where $\mathbf{V}$ is the ball velocity vector, $\mathbf{F}_G = -m\mathbf{g}$ with $\mathbf{g}$ the gravity vector field, and $\mathbf{F}_A(\mathbf{V})$ the aerodynamic force applied on the ball:

$$\mathbf{F}_A(\mathbf{V}) = \mathbf{F}_L(\mathbf{V}) + \mathbf{F}_D(\mathbf{V}) \tag{2}$$

Without this last force $\mathbf{F}_A(\mathbf{V})$, the ball trajectory would be a parabolic arc; when taking $\mathbf{F}_A(\mathbf{V})$ into account, the parabola is modified by the drag effect $\mathbf{F}_D$ and by the Magnus effect (lift) $\mathbf{F}_L$ [9], [13], [16] (see Fig. 2).

The drag acts as a frictional force opposite to the relative motion of the ball. Given C_D the coefficient of friction, ρ the air density ($1.2kg/m^3$), $A = \pi.r^2$ the surface of friction with the air for a ball of radius r (here $r = 2cm$), $\mathbf{V}$ the velocity vector of norm V and polar angle θ (in the coordinate system related to the center of mass of the ball, see Fig. 3), the drag force $\mathbf{F}_D$ is written as :

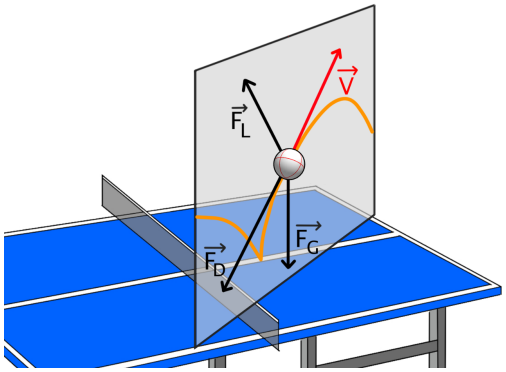

Fig. 2. Forces applied on the ball (black) and ball velocity vector (red)

$$\mathbf{F}_D(\mathbf{V}) = -\frac{1}{2}C_D\rho AV\mathbf{V} \tag{3}$$

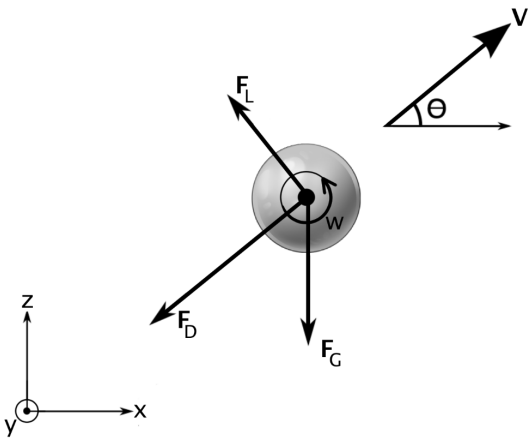

Fig. 3. Forces applied on the ball are in the 2D plane that contains its trajectory

When the ball spins, difference of air pressure on the upper and lower sides of the ball causes a modification of its trajectory called the Magnus force [1]. Given $\mathbf{w}$ the angular velocity vector of the ball of norm w, and S_0 the so-called "lifting" parameter, the Magnus force is written as:

$$\mathbf{F}_L(\mathbf{V}) = S_0\mathbf{w} \wedge \mathbf{V} \tag{4}$$

This force, orthogonal to the velocity vector, induces topspin or backspin. For the considered stroke types (section IV-A) [9], it is reasonable to assert that between two strokes, the ball moves in a 2D plane [9], [17]. The angular velocity vector $\mathbf{w}$ is then orthogonal to the plane $x-z$ during the ball trajectory. The lift coefficient C_L is written as:

$$C_L = \frac{2S_0}{\rho A}\frac{w}{V} \tag{5}$$

and its magnitude:

$$F_L(V) = \frac{1}{2}C_L\rho A V^2 \tag{6}$$

From Newton's second law (equations 1 and 2), horizontal and vertical components of F_D and F_L, respectively (F_{Dx}, F_{Dz}) and (F_{Lx}, F_{Lz}), write:

$$\begin{cases} F_{Dx} &= -\frac{1}{2}C_D\rho A V^2\cos(\theta) \\ F_{D_z} &= -\frac{1}{2}C_D\rho A V^2\sin(\theta) \\ F_{Lx} &= -\frac{1}{2}C_L\rho A V^2\sin(\theta) \\ F_{L_z} &= +\frac{1}{2}C_L\rho A V^2\cos(\theta) \end{cases} \tag{7}$$

At time t, we denote by $V(t)$ the velocity vector norm, $\theta(t)$ its angle with respect to the x axis, and $C_L(t) = \frac{2S_0}{\rho A}\frac{w}{V(t)}$ the lift coefficient. Using equation 1 with $\frac{d\mathbf{V}(t)}{dt} = \left(\frac{d^2x(t)}{dt^2}, \frac{d^2z(t)}{dt^2}\right)$, equations of motion projected on the x and z axes write:

$$\begin{aligned} \frac{d^2x(t)}{dt^2} &= -\frac{1}{2m}\rho A V(t)^2(C_D\cos(\theta(t)) + C_L(t)\sin(\theta(t))) \\ \frac{d^2z(t)}{dt^2} &= -g - \frac{1}{2m}\rho A V(t)^2(C_D\sin(\theta(t)) + C_L(t)\cos(\theta(t))) \end{aligned} \tag{8}$$

Given at $t = 0$ the initial position (x_0, z_0) of the ball , norm V_0 of the velocity vector, angle θ_0 and the angular rotation speed w, it is possible to use Euler's method to iteratively compute the successive positions $(x(t), z(t))$ of the ball center of mass along the trajectory path. Let us note that w is required to calculate the lift coefficient C_{L_0} (5). Given a small time step Δt, equation 8 leads to:

$$\begin{aligned} x(t+\Delta t) &= x(t) + V(t)\cos(\theta(t))\Delta t \\ &\quad - \frac{1}{4m}\rho A V(t)^2(C_D\cos(\theta(t)) + C_L(t)\sin(\theta(t)))\Delta t^2 \\ z(t+\Delta t) &= z(t) + V(t)\sin(\theta(t))\Delta t \\ &\quad - \frac{1}{2}(g + \frac{1}{2pm}\rho A V(t)^2(C_D\sin(\theta(t)) \\ &\quad + C_L(t)\cos(\theta(t))))\Delta t^2 \end{aligned} \tag{9}$$

For computational purposes, these equations have to be discretized. Given a positive integer i, successive ball positions $(x[i], z[i])$, rotation angle $\theta[i]$, lift coefficient $C_L[i]$ and speed norm $V[i]$, equation 9 becomes:

$$\begin{aligned} x[i+1] &= x[i] + V[i]\cos(\theta[i])\Delta t \\ &\quad - \frac{1}{4m}\rho A V[i]^2(C_D\cos(\theta[i]) + C_L[i]\sin(\theta[i]))\Delta t^2 \\ z[i+1] &= z[i] + V[i]\sin(\theta[i])\Delta t \\ &\quad - \frac{1}{2}(g + \frac{1}{2m}\rho A V[i]^2(C_D\sin(\theta[i])) \\ &\quad + C_L[i]\cos(\theta[i])))\Delta t^2 \end{aligned} \tag{10}$$

III. Proposed method

In order to extract the kinematic parameters of the ball, an accurate camera calibration is required to be able to project the objects from the 3D world space to the 2D camera space. To that end, we use a PyTorch [15] object detection library `Detectron2` [5] to detect the ball and initialize a robust tracker, CSRT [10], to obtain continuous ball trajectories on image sequences. A CNN is trained using cropped region centered on the ball to obtain an accurate pixel-size estimation. This information, together with calibration matrices allows to infer the distance between the camera and the ball for each frame. We are then able to apply our equations to the 3D trajectory and obtain the physical parameters.

A. Camera Calibration

The first step of the proposed process is to correct geometrical distortions by calibrating the camera. This is done only once before starting any acquisition. Commonly, camera parameters are split into intrinsic and extrinsic parameters.

The intrinsic matrix contains the focal length, image sensor format, and principal point and is obtained using checkerboard patterns [20].

The extrinsic matrix contains the information to transform 3D world coordinates to 2D camera coordinates. It is obtained using the knowledge of some elements of the scene (standard dimensions and heights of the table and the net) and by manually locating their pixels coordinates in the images (shown in Fig.4). This mapping is exploited by a "Perspective-n-point" type algorithm [6] with the Levenberg-Marquardt optimization method.

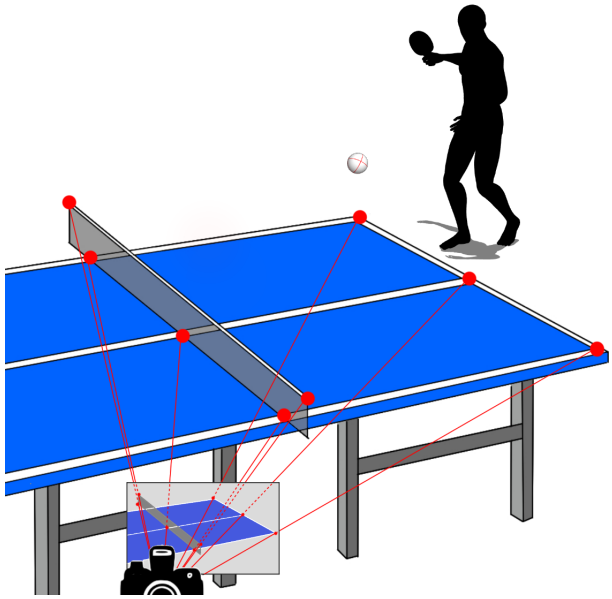

Fig. 4. Reference points used in the 3D to 2D projection matrix

B. Ball tracking and trajectory segmentation

The next step is to track the ball through the image sequence. 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. 6). To initialize a ball tracker in our sequences, we used `Detectron2` [5]. `Detectron2` is performing well for ball detection, even with reasonable motion blur, but fails when motion blur is too important (for some `Top Spin` strokes for example). Once detected, associated with a fast and reliable tracker, CSRT [10], the ball is successfully tracked in all our video sequences. The ball, when bouncing off the table or hit by a racket, quickly changes its trajectory. This information is used to temporally segment the data. Figure 1 illustrates such a typical situation.

C. Ball size estimation using CNN

Knowing the ball positions in the images is insufficient 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. 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:

$$D = f \times H/h \tag{11}$$

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 read world space by applying the extrinsic calibration matrix.

Due to high speed and motion blur, exact ball size in pixels is hard to obtain. As a small error on this estimation results in a consequent error on the ball-to-camera distance, the estimation of the ball location in the 3D space can be erroneous. To solve this issue, we designed a convolutional neural network trained on a generated dataset created with the Blender software (section IV-B). A generated dataset has many advantages: to train the network, as many sequences can be generated as needed and the exact 3D position of the ball and the type of strokes are known.

The architecture of the proposed CNN is summarized in Table I. This network uses 5 consecutive cropped areas of 128×128 pixels centered on the ball to estimate its size on each frame. The ball size does not vary much for consecutive frames, and using a time window of 5 frames helps to regularize ball size estimation. It must be noted that due to the required size accuracy, the video was not spatially downscaled. The size ground truth was extracted from the actual 3D position using equation (11) and was used to compute the loss function (Mean Square Error).

TABLE I
NETWORK ARCHITECTURE FOR BALL SIZE ESTIMATION

Input Size	Operator	In-channels/Out-channels
128x128	Conv 3x3	5/32
128x128	Leaky Relu	32/32
128x128	MaxPool 2x2	32/32
64x64	Conv 3x3	32/64
64x64	Leaky Relu	64/64
64x64	MaxPool 2x2	64/64
32x32	Conv 3x3	64/128
32x32	Leaky Relu	128/128
32x32	MaxPool 2x2	128/128
16x16	Flatten	128/-
32768	FullyConnected	-
8192	Relu	-
8192	Dropout	-
8192	FullyConnected	-

IV. DATASETS USED

Two different datasets were used for building up experiments: a real one filmed in a sport hall with teachers and players from La Rochelle Table Tennis club, and a generated one, mainly for training purposes.

A. The real Dataset

Several stroke sequences focusing on a single player were recorded, using two high speed synchronized cameras (240 fps). 29 sequences were annotated in total, and each video represents a shot sequence between two players. The three considered strokes are: `Top Spin`, `Counter Attack`, and `Push` shown in Fig. 5.

Fig. 5. Extracted frames of a `Top Spin` (left), a `Counter Attack` (center) and a `Push` (right)

Those strokes have typical trajectories that differ from each other by translation and rotation speeds. `Top Spin` is an offensive stroke, that causes the ball to move fast with a high forward rotation speed. `Push` is a defensive stroke, that slows down the ball speed and makes the ball rotates backward. `Counter attack` is intermediate between those two strokes, and causes the ball to move at medium translation and rotation speed. Table tennis rackets usually have a surface made of rubber to maximize the grip on the ball to convey a spin. Video samples of such strokes can be seen online[1]. The angle of the racket face when a player hit the ball creates a spinning motion of the ball and is visually distinctive, especially for `Top Spin`. Each sequence starts when the ball appears in the camera field of view, and stops when it disappears. Their duration is in average 0.6 second (145 frames). Knowing the two extrinsic camera calibration matrices, 3D reconstruction of the scene is possible using triangulation. The pixel size of the ball can also be estimated and will be used as ground truth in experiments of section V-C. Each sequence is labeled with the stroke performed, however, rotation or translation speed of the ball are unknown.

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

B. Generated dataset

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 [3]. Camera calibration matrices obtained on the real dataset (section IV-A) contain all geometric information required to generate close to reality images. To be close to real sequences, we have used 3D models of table tennis ball and table, and the virtual camera was put at the same location and orientation as the one provided by the extrinsic matrix (Fig. 4). To improve the virtual camera configuration, the focal lengths given by the intrinsic matrix of the real dataset were used.

Each of the three considered stroke types (`Top Spin`, `Counter attack`, `Push`) are defined by translation and rotation speed. Ranges have been chosen based on our prior knowledge and are presented in Table II.

TABLE II
VALUE RANGES (TRANSLATION AND ROTATION) FOR EACH STROKE CLASS

Stroke	**Translation (m/s)**		**Rotation (rotations/s)**	
	min	max	min	max
Top Spin	10.00	16.66	30.00	70.00
Counter attack	4.17	10.00	10.00	20.00
Push	1.38	5.55	-15.00	0.00

As already mentioned, the main difficulty for ball size estimation is the motion blur effect. To generate photorealistic images, the `Cycles` rendering tool was used[2]. The blur parameter depends on the camera acquisition framerate, and was tuned to match real sequences, as illustrated in Figure 6.

Fig. 6. Real motion blur (left) and generated frame (right)

The dataset is composed of 200 videos (representing a total of 30,807 frames). The sequence generation process is as follows:
– An initial position above the opponent table side is randomly chosen. Initial parameters (speed and rotation of the ball) are chosen within the range of the `counter-attack` stroke using a uniform distribution (Table II).
– Using equation 10, the ball trajectory simulation is done, checking at each step the correctness of the stroke: the ball must bounce off the table only once and must not touch the net. In case of failure, a new random set of parameters is chosen with the same initial position until the stroke is valid.
– When the ball is two meters away from the net, a new stroke is simulated. One type of stroke is randomly chosen among its three possible types. As in the first step, parameters are chosen within the range of the given stroke.
– If the trajectory is not valid (the ball does not touch the table, hit the wrong side of the table, touches the net...), a new random set of parameters with the same stroke type is generated until the trajectory is valid.

For each valid shot sequence, exact 3D positions of the ball and the used parameters are stored. The 2D positions of the ball in the image and its pixel size are obtained from the camera calibration matrix (equation 11).

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

V. EXPERIMENTS

Each step was analyzed for evaluating our approach. The first step evaluates the precision of the 3D ball position estimation on our synthetic dataset. The second step, is the validation of the parameters extracted by fitting equations on the trajectory. Finally, we used our CNN and fine-tuned it on real sequences to do an action recognition task.

A. Ball size and 3D position estimation

To evaluate the 3D position of the ball, several experiments were conducted using the generated dataset (section IV-B). The goal was to estimate in the frame the ball size as close as possible to the observed one, even in presence of strong motion blur. A reliable ball size estimation is indeed important for obtaining a precise ball to camera distance. The dataset is composed of 133 train sequences and 67 test sequences. Implementation was done using the PyTorch framework [15], running on a GPU GeForce GTX 1070 with 48GB RAM and an Intel i7-7700HQ processor.

The ball size estimation is done using the proposed CNN with a sliding window of five consecutive frames. No other information is used in the network during the training stage (like ball speed or the stroke type). Due to the different motion blurs induced by slow and fast strokes, results for the three stroke types are presented separately in Table III. The average camera to ball distance error between the estimated 3D position and the ground truth increases when the ball is moving faster (average error of 6.43 cm for the `Top Spin`, compared to 5.27 cm for a `Push` stroke).

TABLE III
AVERAGE DISTANCE ERROR BETWEEN THE REAL AND ESTIMATED 3D POSITION OF THE BALL (IN CM)

Stroke	3D estimation	Planar regression
Top Spin	6.43	4.28
Counter attack	5.84	3.86
Push	5.27	3.52

However the estimated ball size value oscillates around its correct size, leading to errors in the estimated 3D position (see Fig.7). As the ball is assumed to have a trajectory that lies in a plane, a planar regression is used to project those 3D points onto a 2D plane. The resulting 'smooth' positions of the ball

are located at the intersection between the regression plane and the ray from the camera to the 3D points. This leads to a smoothed trajectory on which the average error for a `Top Spin` decreases from 6.43 to 4.28 cm. This error represent less than 2% of the distance between the camera and the center of the table (that is 248.65 cm).

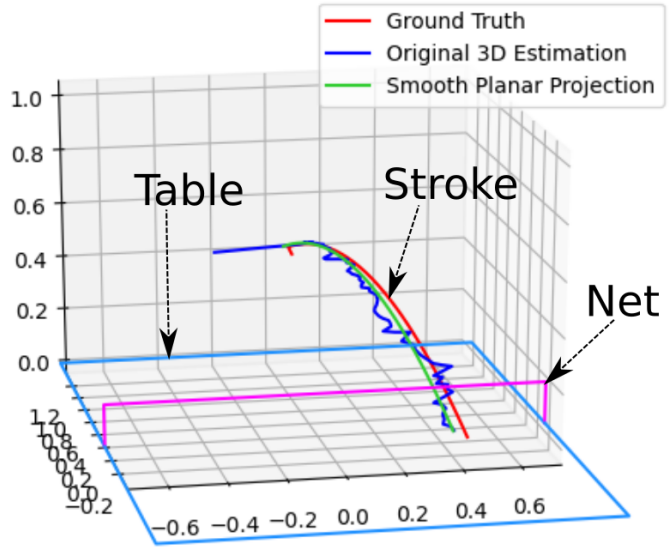

Fig. 7. Comparison with the ground truth of reconstructed 3D trajectory and smooth trajectory obtained after planar regression

Figure 8 shows a plot of the ball size estimation for a `Top Spin` stroke. The corresponding estimated 3D trajectory is shown in Figure 7. As expected, a small variation on the size of the ball impacts its estimated 3D position due to an error on the camera to ball distance. However, the trajectory resulting from the proposed smoothing process is very close to the ground truth.

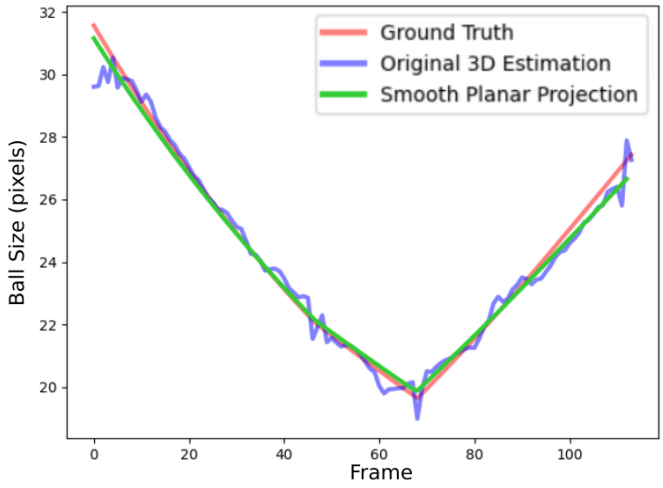

Fig. 8. Extracted ball size in pixel using proposed CNN

B. Estimation of forces applied on the ball

The estimation of forces applied on the ball is now restricted to the previous estimated 2D plane. On this 2D plane, the successive positions of the ball along its trajectory can be computed, given initial values for equations 10 and using Euler's method. The required initial values are: the position of the ball in the 2D space, the velocity vector (magnitude and direction), and the rotation speed. Position in the 2D space is the only known initial value. The other initial values are found by grid search minimization of the quadratic error between the predicted parameters using Euler's method and the estimated 2D plane positions. The grid search minimization uses two successive coarseness to fit the target curve. A first search is done between the min and max values of each parameters (translation: from 1.4 to 16.6 m/s, and rotation speed: from -15 to 70 rotations per second), with a step of one unit. After this rough approximation of the parameters, a more accurate grid is used around the obtained values (-5/+5) using a step of 0.1, which better fits the trajectory.

Table IV presents the errors obtained between the ground truth and the estimated parameters using grid search minimization.

An example of obtained trajectories for a `Counter Attack` and a `Push` stroke is presented on Figure 9. The impact of the Magnus effect on the ball trajectory is very noticeable when comparing the estimated trajectory with the one estimated without taking into account the aerodynamic force $\mathbf{F}_A(\mathbf{V})$ (same angle and initial speed). During a `Counter Attack`, the rotation speed causes a drop in the trajectory of the ball due to the interaction with the air. The ball falls earlier and faster when the Magnus effect is taken into account ($\Delta x < 0$). On the other hand, during a `Push`, the rotation on itself prevents the ball to quickly drop, and has a longer trajectory ($\Delta x > 0$).

TABLE IV
AVERAGE ESTIMATED ERROR OF THE EXTRACTED PARAMETERS FOR EACH OF THE THREE STROKE TYPES

Stroke	Translation speed (m/s)	Rotation speed (rotations per second)
Top Spin	0.75	4.48
Counter attack	0.16	2.70
Push	0.09	0.99
Global	0.41	3.04

C. Action recognition from extracted ball kinematic parameters

The relevance of the extracted ball kinematic parameters have been tested for fine-grained action recognition on the real dataset introduced in section IV-A. On these sequences, the stroke type has been labelled by expert knowledge, velocity or rotation speed of the ball are however unknown. These physical parameters are obtained as follows. For each video sequence of the dataset, the 3D ball positions are obtained by triangulation as described in section IV-A. As the calibration matrices are known, the size of the ball in pixel can be

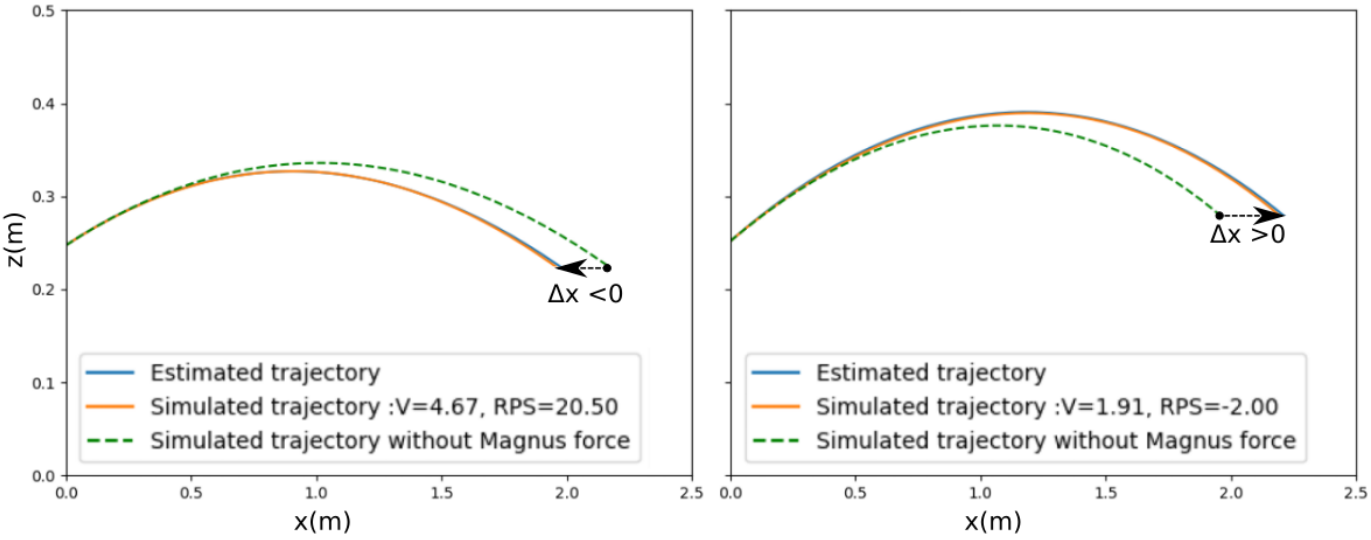

Fig. 9. Comparison between estimated trajectories with and without Magnus effect on `Counter Attack` (left) and `Push` (right) strokes

calculated for each frame. The proposed CNN, pre-trained on the generated dataset is then fine-tuned on 20 stereo sequences from the real dataset. Once this transfer learning process is achieved, the estimation of ball kinematic parameters can be done on real images from a single camera. In order to validate the approach, the rotation and translation parameters calculated on 9 other sequences (three sequences for each stroke type) are presented in Table V.

TABLE V
CLASSIFICATION RESULTS FROM EXTRACTED KINEMATIC PARAMETERS ON THE REAL DATASET

Name	Speed (m/s)	Rotation per sec.	Ground truth	Naive Bayes Prediction		
				TS	Counter	Push
Seq. 1	13.00	38.50	Top Spin	**1.00**	0.00	0.00
Seq. 2	14.00	32.50	Top Spin	**1.00**	0.00	0.00
Seq. 3	13.00	32.00	Top Spin	**1.00**	0.00	0.00
Seq. 4	10.00	10.00	Counter	0.03	**0.97**	0.00
Seq. 5	9.80	9.50	Counter	0.00	**1.00**	0.00
Seq. 6	9.00	9.00	Counter	0.00	**1.00**	0.00
Seq. 7	5.00	-15.00	Push	0.00	0.00	**1.00**
Seq. 8	4.80	-13.50	Push	0.00	0.00	**1.00**
Seq. 9	5.00	-14.00	Push	0.00	0.00	**1.00**

It can be observed that these two kinematic parameters are sufficient to identify the types of strokes: using a simple classifier such as a Naive Bayes Classifier [19] leads to perfect classification results.

VI. CONCLUSION

Using a single camera, the proposed method allows to compute 3D kinematic parameters of table tennis balls. It relies on the knowledge of the standard dimensions of the objects of interest (ball and table). Contrary to most approaches for sport gesture analysis, which perform this 3D reconstruction using stereo cameras or markers that involve many practical constraints, our approach greatly simplifies the setup of the capture devices. The calibration process to obtain the intrinsic parameters of the camera can be indeed performed only once, the placement of the camera being then relatively free as long as at least 4 points of interest (table corners, net ...) are visible. Increasing the number of points increases the calibration accuracy. This approach satisfies a strong demand from sports coaches that expect to study sports gestures in *ecological* conditions without constraints for the players. With an average translation speed error of 0.41 m/s (2.6% on a scale from 1.4 to 16.6 m/s), and a rotation error of 3.04 rotations per second (3.5% on a scale from -15 to 70) on the synthetic dataset, our model allows a precise estimation of the forces applied to the ball in three dimensions. It can thus potentially be of great help to coaches for analyzing player strokes during their training. On the real dataset, all the kinematic parameters extracted from the video sequences are correlated to our knowledge on the considered strokes.

This paper opens perspectives for the analysis of sports gestures 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. There are still many possible improvements on the proposed method. For instance, the calibration process could be improved and made fully automatic by improving points of interest detection in the calibration stage. The estimation of the Magnus effect could also be obtained more efficiently by improving the equation solver.

VII. ACKNOWLEDGMENTS

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 by the MIRES federation.

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] 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.

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

[4] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 1933–1941, 2016.

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

[6] 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.

[7] Amir Roshan Zamir Khurram Soomro and Mubarak Shah. UCF101 - Action Recognition Data Set, 2012.

[8] 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.

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

[10] 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.

[11] Pierre-Etienne Martin, Jenny Benois-Pineau, Boris Mansencal, Renaud Péteri, Laurent Mascarilla, Jordan Calandre, and Julien Morlier. Sports Video Annotation: Detection of Strokes in Table Tennis task for MediaEval 2019. In *Proc. of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-29 October 2019*, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] Shuiwang Ji ; Wei Xu ; Ming Yang ; Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:221–231, Jan 2013.

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

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