How Camera Angle Impact Table Tennis Ball Bounce Tracking

Aymeric Erades, Romain Vuillemot

LIRIS, Ecole Centrale de Lyon, France

aymeric.erades@ec-lyon.fr, romain.vuillemot@ec-lyon.fr

Sports video analysis is a crucial aspect of elite competition preparation, but high-quality data is essential to obtain reliable results. We present a method and an empirical model for evaluating the influence of camera angle on the detection of ball bounces in table tennis. Our model characterizes the non-negligible impact of the way games are recorded on accuracy, which should be carefully considered when selecting camera positions and communicating analysis results.

1 Introduction

To improve performance of table tennis players, an approach is to rely upon video tracking during competitions. Video tracking enables the collection of spatial information over time for objects in the scene such as players, the ball, in a non-intrusive way [1]. These data fall into the categories of tracking data [2], which have proven valuable for sports analysis in general. This information can be used to identify players' strengths and weaknesses, analyze their tactics [3], to prepare for upcoming matches, and anticipate an opponent behavior during a rally [4]. Such tactical analysis relies heavily on collected data, and numerous approaches have shown how to effectively use this information for both analysis and visualization [5, 6, 3, 4]. Other research has shown that certain positional data can be re-configured relatively to players [7] showing the importance of position in analysis.

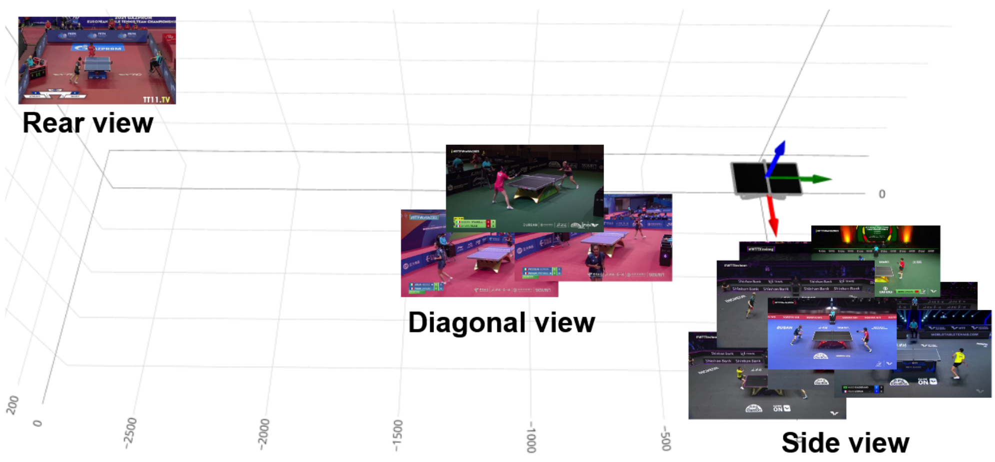


Figure 1: Various points of views collected during table tennis international competitions and their positions in 3D space. On the table, the red axis corresponds to the x axis, the green axis to the y axis and the blue axis to the z axis.

Our approach to collect such data relies upon video tracking techniques, particularly through manual match annotation [8]. Indeed, despite the progress in the fields of computer vision and deep learning, it remains challenging to accurately spot players and balls positions, and classify events [9]. The automatic detection of the table tennis ball fits into a broader issue in the world of sports: the detection of *High-speed and Tiny Objects* [10, 11]. Automated ball tracking remains an open challenge as video quality and framerates are often low on TV. Traditional methods relied on a combination of background or motion subtraction to detect moving objects, and colorimetry to identify the ball among them [12, 13]. More recently, methods based on deep learning have been employed [14, 15], primarily using convolutional neural networks to enable ball detection within the image.

An underexplored area in sports ball tracking is understanding how camera angle impact the tracking quality, regardless it is achieved manually or automatically. The specific characteristics of table tennis mean that choices about how to record are important. Camera positions are often chosen to provide the best view for spectators [16], which does not always make them optimal for data collection. Figure 1 shows different camera angles from various table tennis competitions often found on TV (these videos and their data are part of the dataset [17]). Indeed, broadcasters often use fixed cameras, which must comply with specific positioning guidelines that vary depending on the competition. Despite standardization efforts, notable differences can be observed from one competition to another. Based on Figure 1, we can roughly identify three distinct viewpoints: the *rear view* (in blue), the *side view* (in red), and the *diagonal view* (in green), each provide its own type of geometric distortion of the table tennis game. We next define the types of deformations and characterize their impact on ball bounce tracking accuracy.

2 Problem formulation

A table tennis match can be represented as a 3D scene. The different objects present in such scene include the table (the only fixed object with the floor), and players, their rackets, and the ball, which are moving objects. In an orthographic view, the table has two axes of symmetry: one along the line formed by the net, and the other perpendicular to the net, passing through its midpoint. We chose the origin of the 3D coordinate system as the point of intersection of these two lines on the table, as shown in Figure 1 with a red, green and blue axis. A more abstract representation of such scene is Figure 2 which illustrates the principle of projecting 3D elements and onto the 2D image plane of the camera, as well as the geometric issues that arise from this projection. This projection—and consequently the resulting geometric distortion—depends on the camera, its intrinsic parameters, its position, and its orientation. As TV broadcast videos try to minimize distortion, as mentioned by the European Broadcasting Union[1] (EBU), a maximum height distortion tolerance of between $+1\%$ and -1% is permitted for video broadcasts (except for wide angles, where the tolerance is between $+2\%$ and -2%), we neglect intrinsic parameters, in particular as we are primarily interested in camera angles rather than their distance. For the example in figure Figure 2, which respects the actual dimensions of a table, the table projection represents 0.18 times the table. This ratio means that 1 pixel is equivalent to approximately $5.5cm^2$

We can compute the projection $\mathbf{p}$ of any point $P = [X, Y, Z, 1]^T$, where X, Y, Z are the coordinates in 3D space, onto the camera image using the following equation:

$$\mathbf{p} = \underbrace{\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}\text{ (intrinsic)}} \cdot \underbrace{[\mathbf{R} \mid \mathbf{t}]}_{\text{extrinsic}} \cdot \underbrace{P}_{\text{3D point}} \tag{1}$$

$\mathbf{K}$ is the camera's intrinsic matrix:

- f_x, f_y: focal length in pixels (according to x and y)
- s : skew coefficient (often 0)
- c_x, c_y: optical center (usually the center of the image)

Extrinsic parameters:

[1] `https://tech.ebu.ch/docs/tech/tech3249.pdf`

- $\mathbf{R}(3 \times 3)$ is the rotation matrix
- $\mathbf{t}(3 \times 1)$ is the translation vector

$\mathbf{p} \in \mathbb{R}^3$: homogeneous image coordinates Coordinates in pixels in the 2D image:

- $(u, v) = \left(\frac{p_x}{p_z}, \frac{p_y}{p_z}\right)$

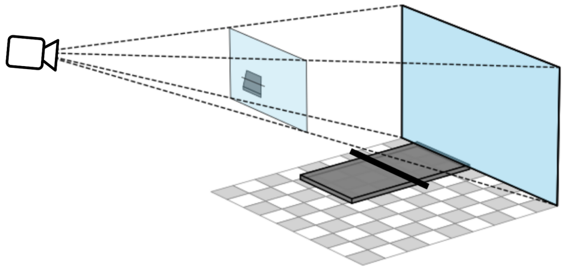


Figure 2: Projection of a 3D scene onto a 2D image showing how physical object change of visual aspect when they are projected on the camera plan. This representation of the table corresponds to the rear view, with a camera located at a height of $10m$ and a distance of $15m$ from the table.

The angle between the camera and the table affects the geometric shape of the table's projection in the image. Positions of the camera $(0, 0, z)$ for any strictly positive real number z are the only ones that preserve the rectangular shape of the table surface without perspective distortion. This rectangular shape is important in sports video analysis as it enables to calculate positions. To measure the error between a real position on the table and its observation resulting from the projection of this 3D scene into the camera's 2D image, we used a simple Euclidean distance:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

,where: $A = (x_1, y_1)$ is the annotated bounce and $B = (x_2, y_2)$ is the ground truth. To give an order of magnitude, we expect our ball bounce tracking to be below 10cm. We chose this level of accuracy because, according to experts from the French Table Tennis Federation, it corresponds to the accuracy achieved by professional players when aiming a particular position on the opposite side.

3 Protocol

As we cannot collect ball bounce ground truth in TV broadcast videos, we recorded videos with known position of balls on a regular table tennis table for various camera angles. The cameras are located at x-axis distances of up to $4.20m$, y-axis distances of up to $3.60m$ and z-axis heights of between $0.14m$ and $4.24m$ above table level. To calculate the position of the cameras, we used OpenCV's calibration functions[2], which allow us to determine the position of a camera using the known positions of six points in 3D space. As reference points, we chose the six characteristic points of the table (the four corners and two at the net), which is of a standard and known type. We placed 8 balls (4 on each side). To position the balls, we measured the distances on the table and marked the positions where the balls should be placed. The table was slightly rough, which enabled to keep the balls in place without having to add anything to secure them. We recorded 13 short videos (3 seconds each) from three camera viewpoints: rear (5 captures), side (4), and diagonal (4), each at varying distances and heights. Then we used an annotation tool we built to annotate all bounces from all camera images. To annotate the ball's positions, we have to click on the ball's position in the image, allowing for pixel-level annotation. The tool then uses a

[2] `https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html`

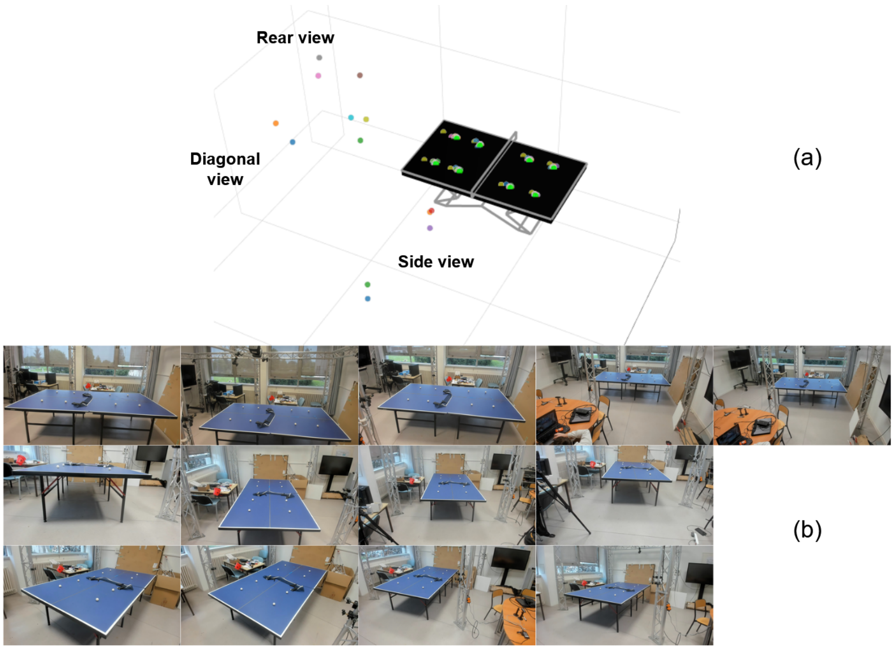
homography to calculate the ball's position on the table. For a point $p = [u, v]^T$ to compute the real position (x, y) on the table, we use the equation:

$$\tilde{\mathbf{p}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \Rightarrow \quad \tilde{\mathbf{P}} = \mathbf{H} \cdot \tilde{\mathbf{p}} = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} \quad \Rightarrow \quad (x, y) = \left(\frac{x'}{w}, \frac{y'}{w} \right) \tag{3}$$

,where $\mathbf{H}(3 \times 3)$ is the homography matrix computed with the Opencv functions[3] using at least 4 known points in the plane and their correspondences in the projected plane (we chose the table corners as references).

The annotation was done by the same person to ensure consistency in how positions were annotated.

Figure 3 (b) shows the setup of the experiment. We then annotated the ball positions and calculate the error rate with the ground truth. Figure 3 (a) shows the camera angles, ball ground truth and annotation results.



(a)

(b)

Figure 3: (a) Positions of cameras we used in our experiment and balls positions on the table; (b) view from the cameras.

Influence of the camera distance

To study the accuracy of annotations, we calculated the deviation of the annotated ball position with their ground truth. We found that deviation over $10cm$ from ground truth come from three of the 13 viewpoints present (two in rear view and one in diagonal view), which leads us to the observations: the rear view is the one with the greatest deviations from ground truth, and the side view shows little deviation.

A regression plot of the distance between annotation and ground truth as a function of distance between camera and bounce, shows a positive slope 11 times over 13 as distance increases. We used linear, exponential and power regressions and had the same result with almost or all positive slopes. Statistically, this shows that as the distance between camera and bounce increases, the distance between annotation and ground truth increases.

[3] `https://docs.opencv.org/4.x/d2/de8/group__core__array.html`

Influence of the camera angle

To study the effect of camera angle on annotation precision, we developed a second protocol where the camera's x and y positions were fixed, and only z position varied. We placed more balls on the table (40) to allow more comparisons. We performed 6 captures at different heights. We used the same process as for the previous protocol to capture the data.

The linear regression of absolute distance between each camera's annotation and the ground truth as a function of the camera height presents a negative slope 40 times out of 40, which means that the higher the height, the greater the precision. For each ball position on the table, the camera height therefore influences the annotation precision. The lowest camera is located at a height of $17cm$ relative to the table level, and it provides the most uncertainty in the annotation with an average of $27cm$ and up to more than $50cm$. From a height of $59cm$ (the second lowest position), the precision for all points is less than $10cm$. For the highest camera position at $130cm$, we obtain an average precision of $1.87cm$.

4 Empirical model

As we are interested in capturing or anticipating the error rate in ball bounce accuracy, we built an empirical model that will provide an accuracy estimation based on our experimental setup (Figure 4). Using the deviations between the annotation and the ground truth as a function of only the camera angle to the bounce, we performed an exponential regression. We chose to use exponential regression because we noticed in the data that for small angle values there was a high degree of variability in accuracy, and that above 30 degrees the accuracy stabilized and approached 0. Based on this observation, we eliminated polynomial regressions, which made it difficult to meet the stabilization constraints. We tested linear, logarithmic, exponential, and power regressions. We calculated the coefficient of determination to determine how well the regressions explain the data, and the exponential regression had the best value of 0.7 (Table 1). The regression is shown on Equation 4:

$$y(\alpha) = a \cdot e^{b\alpha} \tag{4}$$

With α the camera angle, a initial value (when $\alpha = 0$), and b the slope.

Regression	coefficient of determination
Linear	0.52
Exponential	0.70
Logarithmic	0.68
Power	0.66

Table 1

Figure 4 shows that the smaller the camera angle, the greater the variation in accuracy, as is the case with the camera position with the smallest angle, which has an accuracy of between $5cm$ and $40cm$ depending on the table zone. We also note that from a certain angle between the camera and the table, annotated bounces for the same camera view have angles that vary greatly from one another, giving overlapping annotation groupings for different views with different angles.

This model confirms our initial hypothesis that camera angle impacts ball bounce tracking precision. Still it has several limits beyond the few samples we used for our model. First, a camera does not have the same angle of incidence with all points on the table, so points closer to the camera have a larger angle than points further away. Also, our accuracy estimation is calculated on a discretized table using a $1cm$ by $1cm$ grid, to limit calculation times. Still, this work paves the way for further investigations and applications. For instance, Figure 5 shows 3 examples of accuracy predictions made on 3 different views from 3 matches using the model.

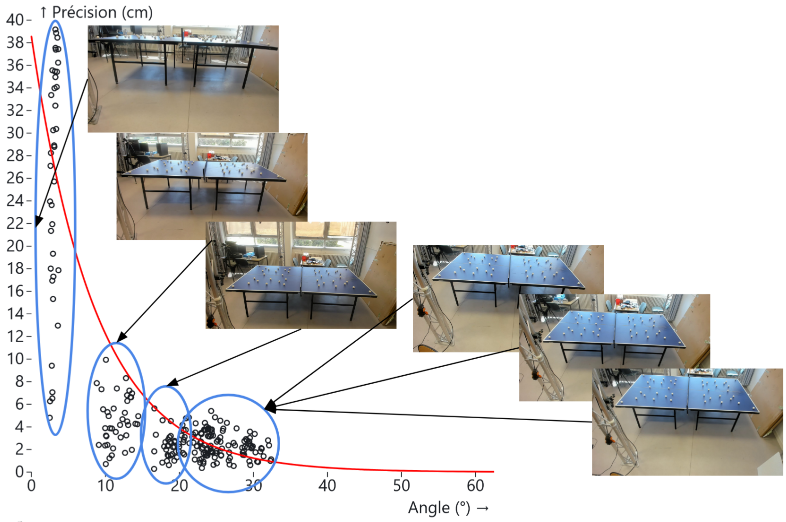


Figure 4: The dots represent deviations between the annotations and the ground truth. On average, as the angle between the annotated bounce and the camera increases, so does the accuracy of the annotation. An exponential regression can model this trend (red curve), with the initial value $a = 38.56$ and the slope $b = -0.115$.

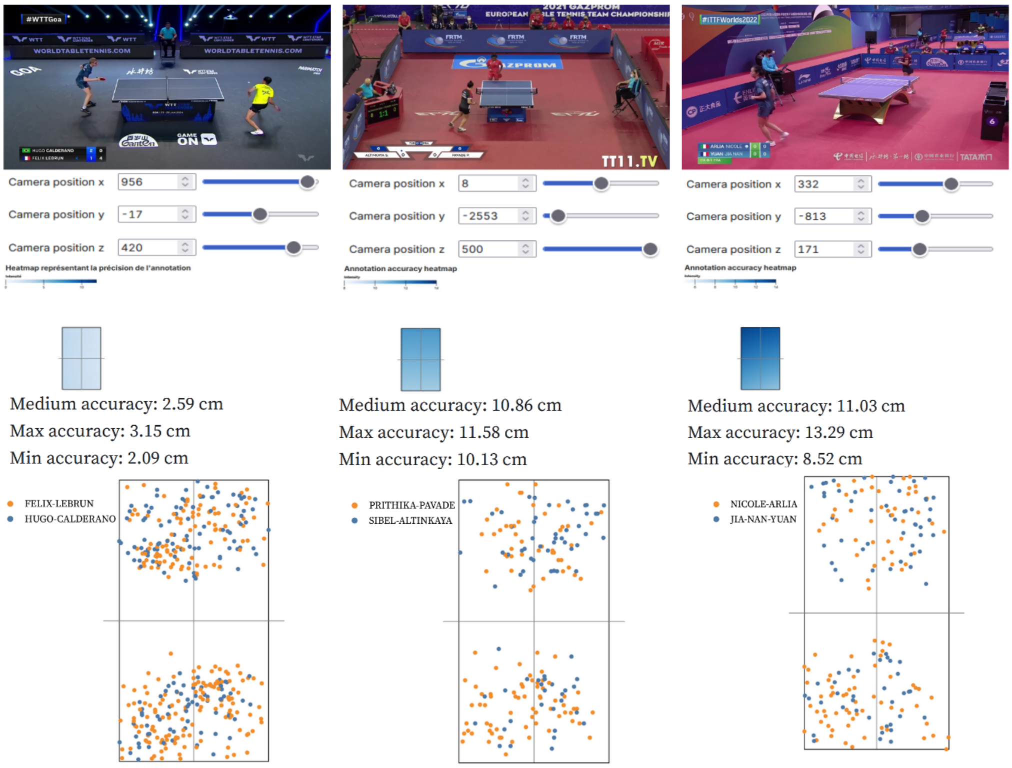
Using our model, we can already estimate the confidence on collected data and eventually provide an error of margin either visually or for calculated statistics. We can see that, depending on the precision of the annotations, there are certain trends: on (b), which has a precision of over $10cm$ for the whole table, the top-down view of all bounces shows a certain trend, with balls close to the net being closer for the bottom half of the table than for the top half of the table. Whereas on (a), which has a precision of less than $3.5cm$, no clear trend emerges. More work research is needed on how to effectively communicate this confidence, either visually or statistically. Also more work is needed to model the impact of other factors such as frame rate, image resolution and camera distance on annotation accuracy.

5 Conclusion, Limits and Perspectives

In this work, we proposed a protocol for evaluating how camera angle impacts table tennis ball bounce tracking. This approach is based on calculating the accuracy of bounce annotations on a table tennis table according to the position of the camera and its angle relative to the table, comparing the ground truth with the annotations. Using the results obtained during the experiments, we developed an empirical model that predicts the accuracy of ball bounce tracking based on the position of the camera relative to the table.

This approach has certain limitations. The first is that we used a single camera to take measurements with 4K resolution (4096×2160 pixels), whereas cameras for professional matches often use a lower resolution. In addition, the camera position was closer than during competitions, where they need to film the playing area in addition to the table.

We have identified several perspectives that we would like to explore in order to develop a more robust prediction model. The first perspective is how the annotations are made. We plan to take more measurements by placing more balls on the table, which will allow us to achieve a greater coverage density and take more captures in order to cover camera positions closer to those used in professional competitions. The second perspective concerns the model itself. We have only taken into account the position of the camera, and we would like to investigate whether the number of pixels making up the table influences accuracy. Finally, the last perspective concerns the correction of rebound positions. By studying the direction of the differences between



(a) (b) (c)

Figure 5: Example of precision estimates for different views taken from broadcasts and table bounces we identified. (a) The side view, the area on the table with the lowest annotation accuracy is $3.15cm$. (b) The rear view, annotation accuracy on the table remains almost constant over the whole table, with a difference between the worst and best accuracy of less than $1.5cm$. (c) The diagonal view, the accuracy of the annotation on the table varies greatly depending on the zone, with a gap of almost $5cm$ between the two extremes.

the actual field data and the values obtained, we would like to study the feasibility of developing a model to correct rebound positions based on the camera position and the position of rebounds on the table.

References

[1] Roman Voeikov, Nikolay Falaleev, and Ruslan Baikulov. TTNet: Real-time temporal and spatial video analysis of table tennis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 884–885, 2020.

[2] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale. State of the Art of Sports Data Visualization. *Computer Graphics Forum (EuroVis'18)*, 37(3):663–686, 2018.

[3] Jiachen Wang, Jiang Wu, Anqi Cao, Zheng Zhou, Hui Zhang, and Yingcai Wu. Tac-Miner: Visual Tactic Mining for Multiple Table Tennis Matches. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):2770–2782, June 2021. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[4] Jiachen Wang, Yihong Wu, Xiaolong Zhang, Yixin Zeng, Zheng Zhou, Hui Zhang, Xiao Xie, and Yingcai Wu. Tac-Anticipator: Visual Analytics of Anticipation Behaviors in Table Tennis Matches. *Computer Graphics Forum*, 42(3):223–234, June 2023.

[5] Yingcai Wu, Ji Lan, Xinhuan Shu, Chenyang Ji, Kejian Zhao, Jiachen Wang, and Hui Zhang. iTTVis: Interactive Visualization of Table Tennis Data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):709–718, January 2018. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[6] Gabin Calmet, Aymeric Erades, and Romain Vuillemot. Exploring Table Tennis Analytics: Domination, Expected Score and Shot Diversity. In Springer Link, editor, *Machine Learning and Data Mining for Sports Analytics*, Communications in Computer and Information Science, Turin, Italy, September 2023.

[7] Aymeric Erades and Romain Vuillemot. Player-Centric Shot Maps in Table Tennis. *Computer graphics Forum (Proc. Eurovis)*, page 10, June 2025.

[8] Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete. Real-Time Crowdsourcing of Detailed Soccer Data. *IEEE VIS*, 2013.

[9] Aymeric Erades, Pierre-Etienne Martin, Romain Vuillemot, Boris Mansencal, Renaud Peteri, Julien Morlier, Stefan Duffner, and Jenny Benois-Pineau. SportsVideo: A Multimedia Dataset for Event and Position Detection in Table Tennis and Swimming. *MediaEval Workshop 2023*, 2023.

[10] Yu-Chuan Huang, I.-No Liao, Ching-Hsuan Chen, Tsì-Uí İk, and Wen-Chih Peng. TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications. Technical Report arXiv:1907.03698, arXiv, July 2019. arXiv:1907.03698 [cs, stat] type: article.

[11] Tianjian Zou, Jiangning Wei, Bo Yu, Xinzhu Qiu, Hao Zhang, Xu Du, and Jun Liu. Fast moving table tennis ball tracking algorithm based on graph neural network. *Scientific Reports*, 14(1):29320, November 2024.

[12] K. Wong. Tracking Table Tennis Balls in Real Match Scenes for Umpiring Applications. *British Journal of Mathematics & Computer Science*, 1(4):228–241, January 2011.

[13] Hnin Myint, Patrick Wong, Laurence Dooley, and Adrian Hopgood. Tracking a table tennis ball for umpiring purposes. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 170–173, Tokyo, Japan, May 2015. IEEE.

[14] Zhewen Zhang, Fuliang Wu, Yuming Qiu, Jingdong Liang, and Shuiwang Li. Tracking Small and Fast Moving Objects: A Benchmark. In Lei Wang, Juergen Gall, Tat-Jun Chin, Imari Sato, and Rama Chellappa, editors, *Computer Vision – ACCV 2022*, volume 13847, pages 552–569. Springer Nature Switzerland, Cham, 2023. Series Title: Lecture Notes in Computer Science.

[15] Haoxuan Li, Saba Ghazanfar Ali, Junhao Zhang, Bin Sheng, Ping Li, Younhyun Jung, Jihong Wang, Po Yang, Ping Lu, Khan Muhammad, and Lijiuan Mao. Video-Based Table Tennis Tracking and Trajectory Prediction using Convolutional Neural Networks. *Fractals*, 30(05):2240156, August 2022.

[16] Jota Ito. The Filming of Table Tennis.

[17] Aymeric Erades, Martin Pierre-Etienne, Romain Vuillemot, and Stefan Duffner. Fine Grained Action Classification and Position of Table Tennis and Swimming Videos for MediaEval 2023. *MediaEval Workshop 2023*, 2023.