International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

A Semantic Web-Based Approach for Bat Trajectory Reconstruction With Human Keypoint Information

Zechen Jin, Beijing Sport University, China
https://orcid.org/0009-0009-2592-2726

Yida Zheng, Beijing University of Posts and Telecommunications, China

Jun Liu, Beijing University of Posts and Telecommunications, China

Yang Yu, Beijing Sport University, China*

ABSTRACT

Restoring the trajectory of a bat from a table tennis match video is critical in analyzing a table tennis technique and conducting statistical analysis. However, directly bat location detection in each frame is challenging due to changing shapes caused by varying movement directions and speeds, leading to ambiguity. This paper develops a novel two-stage method. The first stage utilizes YOLO for bat detection in each frame, followed by filtering out erroneous candidate boxes. In the second stage, the authors use a temporal prediction model that integrating human keypoint information and interpolation to reconstruct a complete bat trajectory with minimal errors. The method's effectiveness and performance are evaluated on our video datasets. The evaluation results demonstrate that the proposed method outperforms traditional methods on precision performance metrics. The error screening algorithm improves precision score to nearly 1. In addition, the method has the recall score 22.3% higher than YOLO 's and also 1.4% higher than that of YOLO with cubic spline interpolation.

KEYWORDS

Bat Trajectory, Object Detection, Semantic Segmentation, Table Tennis Sports Analysis, Time Series Forecasting Model

Table tennis attracts many people worldwide owing to its competition and entertainment features (Voeikov et al., 2020). With the development of computer vision technologies, several researchers have investigated the motion of table tennis bats during the stroke execution (G. Chen et al., 2013). Table tennis bat trajectories reconstruction is a pivotal analysis component in this field, as it can reveal the technical intricacies and movement patterns of table tennis players. Based on the bat trajectories, various metrics can be measured, such as hitting quality, speed, and impact point accuracy, which can provide valuable feedback to the players. Furthermore, when these metrics are combined with

DOI: 10.4018/IJSWIS.338999 *Corresponding Author

other information, multiple follow-up high-value analyses are possible, such as hit point analysis, action normative analysis, and serve action guidance.

Getting a complete and accurate bat trajectory is challenging for three reasons. First, training the bat detection model requires a large-scale dataset containing table tennis bats with different shapes and angles. Creating such a dataset requires annotating many images. Unfortunately, such a dataset does not currently exist. Second, object detection models cannot fully detect the racket in each frame, as there are missed and misdirected detection cases. Therefore, reconstructing the bat trajectories is necessary. Third, due to the fluctuating speed of the bat during the swing, it is difficult to accurately fill in or predict missing values based on interpolation methods or temporal prediction models. Besides, current works have investigated well-defined objects with a clear movement pattern and in limited conditions (Rozumnyi et al., 2020a; Rozumnyi et al., 2021a; Tao et al., 2016).

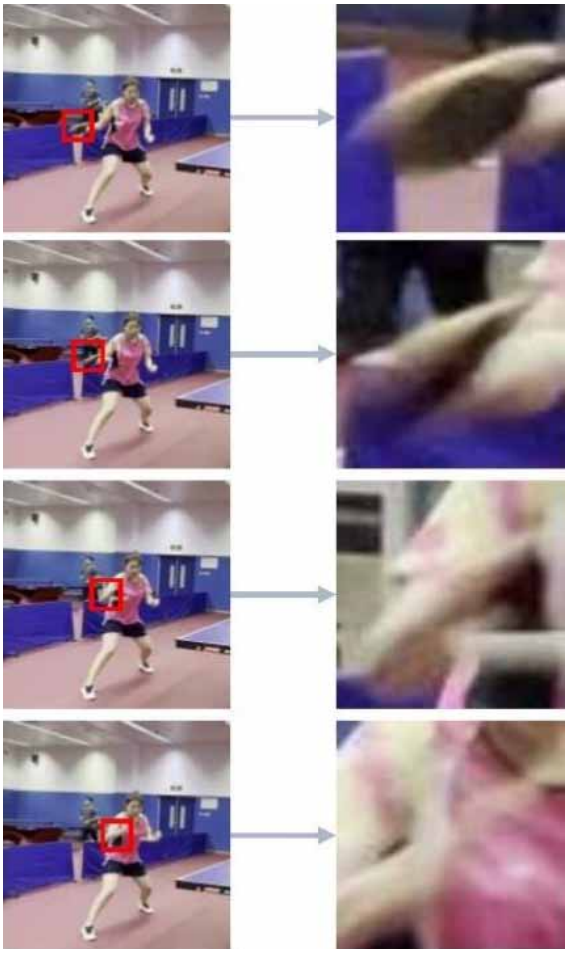
The literature offers several studies on sports analysis using AI technology. For instance, Llanos et al. (2022) used Openpose for table tennis pose detection and four machine learning models for pose analysis. Unlike the authors' work, they categorized different key points of the human body to assess an athlete's posture. AlShami et al. (2023) applied a Transformer to predict a tennis player's future trajectory as a sequence derived from their body joints' data and ball position. Although the trajectory tracking task is similar, the figure subject in the picture occupies a large proportion, and the detection is relatively easy. However, the figure's running distance in the tennis court is large, so the human body's keypoint and the ball's information can be well captured. J. Wu et al. (2021) and G. L. Cai et al. (2022) [ball trajectories2] predicted the flight trajectory of a table tennis ball based on LSTM. However, the problems of miss detection, motion blurring, and shape changes racket trajectories face during restoration significantly affect the performance of their method. Unlike current methods, the authors' is much more novel, especially for table tennis bats.

The authors' method is to detect or predict the position of the bat from the table tennis video frame and reconstruct the complete bat trajectory. However, due to the fast moving bat in swing actions, there are many frames in which the bat cannot be detected. The authors' method predicts and complements these missing bat positions. Recall score is the top priority. It measures the accuracy of bat trajectory reconstruction.

This paper proposes a generalized and robust method to produce and estimate the whole bat trajectories from sports videos. The concept is to use human body keypoints and confidence scores as extra information to decompose the problem into two stages: bat object detection from videos and prediction and complementary of the missing bats from the results of bat detection. A player's arm information is crucial for most swinging actions as it drives the bat's motion, making arm keypoints excellent features for bat trajectories reconstruction. More importantly, the authors' method applies to other scenarios involving trajectory reconstruction of handheld objects. Specifically, based on a pre-trained object detection model specified for bat class, the proposed method first detects bat locations in all frames from the input video. Besides, the authors feed the video into a preset human pose recognition model (Bazarevsky et al., 2020) to obtain human body keypoint information and eliminate erroneous bat candidate boxes. Second, the authors classify the video frames into forward swing frames and others. Then, the authors generate the full bat trajectory based on a temporal prediction model that leverages human body keypoint information and an interpolation method. Finally, to train and evaluate this method, the authors build a video dataset with location information of labeled bats, which is also useful for future work on bat analysis. This paper's main contributions are summarized as follows:

1. The authors propose a temporal prediction model utilizing human body keypoints about bat movement trends. This facilitates the prediction of bat positions under varying velocities.
2. The authors develop a novel bat trajectory reconstruction system that simultaneously addresses trajectory prediction and missing target imputation to obtain an accurate bat trajectory.

Figure 1. Sketch of racket trajectory reconstruction method



3. The authors build a large-scale table tennis video dataset with labeled bat locations and human keypoint information. Experiments on this dataset reveal that this method outperforms traditional methods by a large margin.

RELATED WORK

Optical Flow Method

The optical flow method exploits the temporal variations of pixels in an image sequence and the correlation between consecutive frames to calculate the motion information of objects between adjacent frames, which is based on the correspondence between the previous frame and the current frame. Optical flow can be used to estimate and analyze the motion of objects in a sequence, with the existing methods divided into traditional and deep learning algorithms. Lucas et al. (1981) proposed the Lucas-Kanade sparse optical flow algorithm (Bruhn et al., 2005), which exploits brightness constancy, temporal persistence, and spatial consistency. Bouguet introduced an improved Lucas-Kanade algorithm (Bouguet et al., 2001) based on pyramid hierarchies, overcoming the issues of tracking fast-moving objects and affine transformations. Another traditional approach is the dense

optical flow, such as Farnebäck's method (Farnebäck, 2003), which approximates the neighborhood information of each pixel using polynomials and calculates the displacement for all points in the image. However, the trade-off between the accuracy and speed of this method limits its practical application. Deep learning (Behera et al., 2023; Li et al., 2022; Tembhurne et al., 2022; Zhou, 2022) has yielded promising results in optical flow estimation in recent years. For instance, FlowNets and RAFT (Boyer et al., 2009; Dosovitskiy et al., 2015; Ilg et al., 2017) utilized convolutional neural networks to predict optical flow for each pixel in the image and achieved significant advancements in real-time estimation algorithms. However, for the bats in table tennis training or competition videos, their positions constantly change during the motion, causing rapid transformations and resulting in much motion blur. Therefore, extracting features from such scenarios is too challenging.

Instance Segmentation

Instance segmentation (Hafiz & Bhat, 2020) has witnessed significant research interest in recent years, owing to the rapid progress in deep learning methodologies (Alsmirat et al., 2019; Hu et al., 2022; D. Li et al., 2019; Ling & Hao, 2022; H. Wang et al., 2020; Yu et al., 2018). Generally, instance segmentation combines pixel-level semantic segmentation (L. C. Chen et al., 2017; Ronneberger et al., 2015) and object detection. Instance segmentation methods can be divided into single-stage and two-stage approaches (Gu et al., 2022). Influenced by the single-stage object detection research (C. Li et al., 2022; Lin et al., 2017a; Redmon et al., 2016; Tian et al., 2019), single shot instance segmentation algorithms are categorized into anchor-based methods (Bolya et al., 2019; H. Chen et al., 2020; C. Y. Wang et al., 2023; X. Wang et al., 2020) and anchor-free methods (Dai et al., 2016; X. Chen et al., 2019; Kirillov et al., 2020; Y. Li et al., 2017; Xie et al., 2020). Specifically, YOLCAT (Bolya et al., 2019) employed ResNet101 with FPN, the same as RetinaNet (Lin et al., 2017a), and designed two branch networks to generate data of each candidate frame and prototype mask for each picture. YOLCAT can predict the mask coefficients to generate the instance mask through linear combination. Besides, SOLO (X. Wang et al., 2020) classified each pixel into the corresponding instance category and returned the center of the object pixel by pixel. Regarding size processing, SOLO utilized FPN to assign objects of different sizes to feature maps at different levels used as the size categories of the objects. BlendMask (H. Chen et al., 2020) added a Bottom module to extract low-level detailed features based on FCOS (Tian et al., 2019) and proposed a Blender module to integrate these two features drawing on the fusion methods of (Bolya et al., 2019; Tian et al., 2019). Other anchor-free methods generate the position and shape information of the target directly at each pixel position in the image. PolarMask (Xie et al., 2020) conducted instance segmentation and object detection using the same modeling method. This network is as easy as FCOS (Tian et al., 2019), especially involving Polar CenterNess instead of bounding box centerdness. PointRend (Kirillov et al., 2020) treated image segmentation as a rendering problem. The two-stage methods (Z. Cai & Vasconcelos, 2018; He et al., 2017; Kirillov et al., 2019; K. Wang et al., 2019) first identify objects in the image and then perform pixel-level segmentation inside each box. Mask-RCNN (He et al., 2017) and Panoptic FPN (Kirillov et al., 2019) introduced additional segmentation branches based on (Ren et al., 2015). Z. Cai and Vasconcelos (2018) enhanced Mask r-cnn (He et al., 2017) by cascading multiple stages of the network, gradually improving target detection and instance segmentation performance. However, instance segmentation algorithms require extensive additional annotated data and pre-training. In the bat tracking task, pixel-level segmentation accuracy is not so crucial, opposing the position and motion of the bat. Therefore, this paper employs object detection algorithms.

Small Target Detection

How to detect and localize small-sized targets in complex scenes has been extensively studied in computer vision (Benmoussa et al., 2022; Dwivedi, 2022; Pan et al., 2022). Deep learning-based target detection algorithms can be classified into two categories: two-stage and one-stage. The former first extracts features, generates region proposals, and classifies the samples using a convolutional neural

network. R-CNN (R. Girshick et al., 2014), Fast-RCNN (R. B. Girshick, 2015), and Faster-RCNN (Ren et al., 2015), which rely on VGG as the backbone. These methods have achieved remarkable results in this task. In one-stage detection, features are extracted directly in the network to predict object classification and location without region proposal. YOLO (Redmon et al., 2016) is a typical and representative algorithm based on region extraction. Some methods design various network architectures and feature pyramid structures (J. Cao et al., 2020; Lin et al., 2016a) to capture feature information at different scales and levels. Several recent research works utilized attention mechanism (W. Li et al., 2020), null convolution (Kim et al., 2021), and multi-scale feature fusion (Guo et al., 2020) to capture small objects' details and contextual information, thus improving the accuracy and robustness of target detection. This paper adopts YOLOv5 for bat target detection to efficiently and accurately detect fast-moving bats.

Missing Value Imputation

Currently, several approaches address missing values in sequences. Interpolation methods, including linear interpolation, polynomial interpolation, and spline interpolation (Hagan & West, 2006), are commonly employed. In recent years, recurrent neural network models, such as LSTM (Hochreiter & Schmidhuber, 1997), have been developed to generate sequences by leveraging historical data features. For example, Alahi et al. (2016) utilized information about the interactions among nearby pedestrians to predict pedestrian trajectories. This work introduces a novel method by leveraging an LSTM model fused with human keypoints information to generate the whole bat trajectory in a video.

Bat Trajectory Prediction and Reconstruction

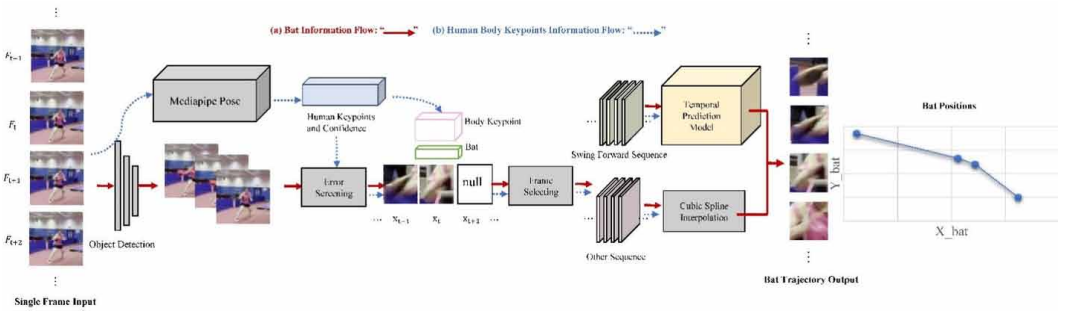
There are few research studies on bat trajectory prediction and reconstruction. During the authors' investigation, they did not see other bat trajectory reconstruction methods under similar conditions and scenarios. There are some studies on baseball (Shibata et al., 2023) and tennis bat trajectory estimation (Furuya et al., 2021) or some table tennis robot (Ji et al., 2021). There are also some studies on the ball trajectory reconstruction like badminton ball (Liu & Wang, 2022) and table tennis (H. Li et al., 2022). Among these similar methods, most of them apply physical sensors, and also their bats are easier to identify in images than table tennis bats. Compared with them, the authors' method does not need complicated equipment. It has strong versatility, and its application scenarios are relatively novel.

METHOD

Figure 2 illustrates the overall framework of the proposed method. The key concept is to leverage human body keypoints and confidence scores. Specifically, first, detect the bats and obtain the human keypoint information from the video, and then predict the missed bat locations by fusing human body keypoints and confidence scores. By extensively studying the process of table tennis strokes, the authors found a strong correlation between the movement patterns of certain points of the arm and the movement of the bat. Inspired by Social-LSTM (Alahi et al., 2016), the authors incorporate this information as surrounding contextual points of the bat location sequence into the temporal prediction model. In addition, they combine an interpolation method to fill the missed bats in other conditions. Next, they describe the proposed method in detail.

It first takes frames of a sports video as input and detects the bat location using YOLOv5. The human body keypoint information is extracted and combined with bat information. The sequences are then fed to a block that separates them into two types and utilizes two models to reconstruct the bat trajectory.

Figure 2. Proposed architecture



Notations

Let a full bat trajectory be $X \in \mathbb{R}^{T\times D}$ and table tennis images captured from a phone or other cameras as $F \in \mathbb{R}^{T\times h\times w\times 3}$, where T is the frame sequence length, D is the dimension of the bat position, including coordinate (x, y) and confidence score c, and h*w is the image size. The authors introduce human body keypoints $P \in \mathbb{R}^{T\times J\times D}$ as additional information to produce the bat trajectory from the input table tennis video, where J is the number of human joints employed. All of the notations are shown in Table 1.

Bat Object Detection

In order to reconstruct the trajectory of a moving object, the authors first perform object detection on each frame and then use the detected bounding boxes to predict and fill missed bats. Therefore, choosing an object detection model that offers both high speed and accuracy is important. However, the fast motions of bats lead to variable shapes in frames and introduce motion blur, making it challenging to detect bats accurately.

Based on these observations, the authors build a bat dataset with multiple poses (refer to Section 4.1 for further details) and use it to train a bat object detection model using the classic YOLOv5 model. The YOLO model performs very well in terms of speed and accuracy. In order to achieve better detection, the authors keep labeling and adding bat images for training. In order to optimize the detection effect of YOLO, they preprocess each frame to make the bat account for a larger proportion of the image. However, the detection results of YOLO are not perfect. The authors design an error screening algorithm for the bat bounding boxes to retain the real part of the frames with bat.

During image preprocessing, the authors employ the yolov5s.pt pre-trained model to detect the left side player in the first video frame and determine the center position of the player $C_1\left(c_{1,x}, c_{1,y}\right)$ and size of the bounding box h_0*w_0 around the player. Note that the authors focus on the player who hits the ball with their right hand and sits on the left side of the screen. Then, the authors crop the original image into smaller images containing only the players according to a certain proportion to standardize the size of the images. The scale ratio is defined as $s = \max\left(\frac{h_0}{h}, \frac{w_0}{w}\right)$. Then, the authors take

Table 1. Summary for the notation

Notation	Description
X	a full bat trajectory
F	table tennis images
T	frame sequence length
D	dimension of the bat position
h	height of an image
w	width of an image
P	human body keypoints
J	number of human joints employed
C_1	the center position of the player
h_0, w_0	size of the bounding box around the player
s	scale ratio of original image
v	vertices to crop the image
l	the line connecting the keypoints of the right elbow and the right wrist of the human body
C_t	relative origin of the t^{th} frame
(x,y)	position of the bat
c	confidence score
t	frame ID of all frames

$$\begin{aligned} V_1 &= \left(c_{1,x} - 0.55 \times s \times w, c_{1,y} - \frac{h_0}{2} \times 0.55\right), \\ V_2 &= \left(c_{1,x} + 0.45 \times s \times w, c_{1,y} + \frac{h_0}{2} \times 0.45\right), \end{aligned} \tag{1}$$

as top left and bottom right vertices to crop the image. According to the actual situation, they need to extend the range of cropped image up and left more than down and right. Where 0.55 and 0.45 represent the scale of expansion from the center towards the upper left and lower right sides of the frame, respectively. This ensures a large proportion of the figure and the bat in the image, which allows for capturing complete information on the human body keypoints and bat movements. After the initial processing of the images, the authors used the trained YOLO model for bat detection.

The authors convert the bat detection results into pixel values and use the absolute coordinates of bat center X to denote a sequence of raw bat trajectory $X_1, X_2, \ldots, X_T$. Among all of the frames from the input video, suppose that the bats are detected in M frames and there are also N frames that

contain falsely detected bats. During the experiments, the authors observed that there are generally two types of false detections: one is where the detection result is located far from the arm, and another is where a part of the human body is detected as a bat. Based on experience, the authors believe the racket's location should be near the elbow-to-wrist line extension. Therefore, they formulate a screening rule accordingly. Specifically, they first extract the relevant human wrist and elbow keypoints and compute the two-point connecting line l_t. Considering the majority of right-handed hits, they calculate the distance d_t between the center point of each frame of the bat $\left(x_t, y_t\right)$ and the line l_t connecting the keypoints of the right elbow P_t^e and the right wrist P_t^w of the human body, where t indicates the serial number in T frames. The distance is formulated using line representation parameters as:

$$d_t = \frac{ax_t + by_t + c}{\sqrt{a^2 + b^2}}, \tag{2}$$

where a, b, and c are the parameters of line l_t. The points that exceed the threshold are considered false candidate bounding boxes. The authors also calculate the angle between the elbow-wrist direction vector and the wrist-bat direction vector. If the angle is larger than the threshold, the position of the bat detection frame is not in the direction of the elbow-wrist extension line, posing a false detection. For the left-handed hits, the authors replace all of the key points on the right side with the left side and perform the same calculation. Here, they experimentally set the threshold value to 60. Besides, they adopt the Mediapipe Pose Model (Bazarevsky et al., 2020) to extract keypoints information from the human body. They choose keypoints 13-16 in this stage and 11, 12, 23, and 34 in the next stage. The human body figure with the human keypoints considered is illustrated in Figure 3. There are T – M + N frames F_{fault} where the bat was undetected or falsely detected, and the authors set the bat data in these frames to null values. The output of the bat detection process is denoted as:

$$X' \in \mathbb{R}^{T*2} = \begin{cases} null, & if\ F_t \in F_{fault} \\ X_t, & otherwise \end{cases} \tag{3}$$

Temporal Prediction Model With the Fusion of Human Body Information

During trials, the authors observed a continuous variation in bat speed throughout the forward swing, transitioning from a slow pace to a rapid one and then returning to a slower pace. This dynamic behavior presents a significant challenge for conventional interpolation methods to accurately fill in the missing data points for the bat's motion during this specific phase. Certain keypoints of the human body, such as the shoulders, elbows, and wrists, exhibit movement patterns similar to the bat. Therefore, the authors leverage these keypoints to predict the bat trajectories. Inspired by the remarkable success of LSTM (Hochreiter & Schmidhuber, 1997) in temporal forecasting tasks, the authors consider the bat trajectories as time series data. Thus, they employ LSTM to generate a complete bat trajectory by fusing human body keypoints. Next, the authors introduce how their approach represents the data and then explain how they employ LSTM to achieve this information fusion and generate the complete bat trajectory.

The authors adopt the widely recognized MediaPipe Pose landmarks comprising 33 key points outlining the human body's skeleton. Mediapipe is a multimedia machine learning application framework developed and open-sourced by Google Research. One of its productions, MediaPipe Pose, is a machine-learning scheme designed for high-fidelity human pose tracking. It leverages research findings from BlazePose (Bazarevsky et al., 2020) and obtains the entire 33 2D landmarks

Figure 3. Human body keypoints figure with the considered points in green

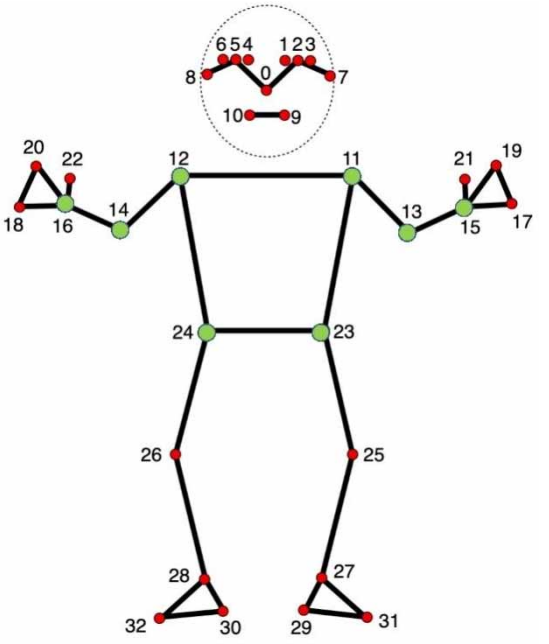
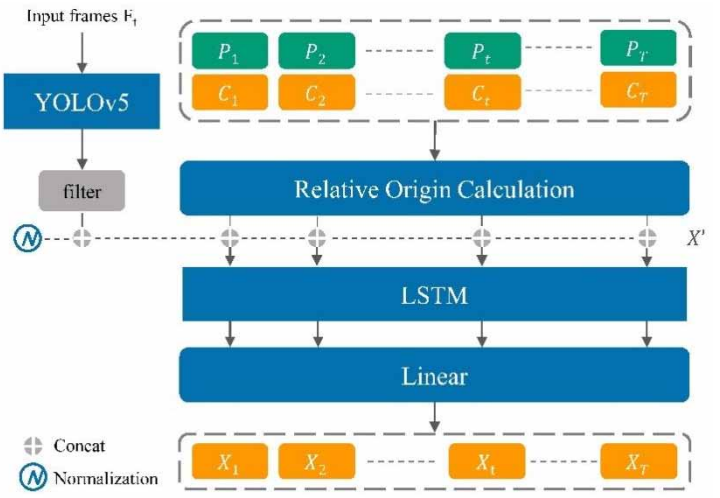


Figure 4. Model architecture of the temporal prediction network involving human body information fusion



of RGB video frames from the ML Kit Pose Detection API. Compared to Openpose (Z. Cao et al., 2017), MediaPipe is more lightweight and thus is well suited for mobile and embedded applications. It can handle multiple data types and easily integrates into different platforms. Hence, the authors employ Mediapipe Pose to output an image's 2D human body keypoints (pose landmarks) rather than the 3D keypoints (pose world landmarks) since depth information is not required. The output coordinates are absolute values in (0,1) and are correct. Depending on whether a player is a left-hander or right-hander, the authors utilize three additional points whose numbers are 11, 13, 15, or 12; 14;

and 16, which are particularly relevant to the movements of a table tennis bat. The authors integrate them into the bat location sequence generated by YOLOv5. Specifically, they utilize each point's information, including its coordinate and confidence score, which ranges from 0 to 1. Owning to possible drifting and discontinuous changes in the positions of human keypoints, they incorporate confidence scores for each keypoint to balance the model's learning of the bat's movement trends without excessively relying on position variations from keypoints with lower confidence. All confidence scores are produced by BlazePose and described as visibility. The human keypoints information is represented as $P_t \in \mathbb{R}^{J*3}$.

After removing the erroneous bats, the authors set the confidence scores of all true bats to 1 and then concatenate the human keypoint information P into X′. In order to normalize the coordinates of various points, the authors use the midpoint of the human body as the relative origin to calculate the coordinates of each point, which is formulated as:

$$C_t = \frac{1}{4}\sum_{j\in S} P_t^j, t \in T, \tag{4}$$

where C_t represents the relative origin of the t^{th} frame, and S represents the set {11, 12, 23, 24} of J human body keypoints. The coordinates of each point are calculated as $X_t^{'} - C_t$ and $P_t - C_t$ in both x-axis and y-axis, respectively. Then, the authors further scale the coordinates by dividing them by 100 to obtain smaller numerical values.

The authors use LSTM to predict and fill missing values in a continuous series of swing motions. Specifically, they consider a time step of four and leverage the bat information from the previous four frames to predict the missed bat in the current frame. Furthermore, they incorporate the keypoint information of the human body from the subsequent frame of each bat frame as additional input. Figure 4 depicts the model architecture of the temporal prediction network that fuses human body information. This method suggests a new composition of input data (i.e., the authors utilize the bat information from $i-4$ to $i-1$ frames and the keypoint information from $i-3$ to i frames to predict the bat's position in the i^{th} frame, as illustrated in Figure 5). The human body information of the current frame is added as an aid to the bat position sequence of the previous frames to help predict the missing bat position of the current frame. Since the human keypoints on the arm are moving in the same trend as the bat, the authors treat their trajectories as time series. These points share the same motion pattern as the bat in a LSTM model. When the authors have more relevant information than just the previous bat sequence, they can better predict and complete the bat for the current frame. The confidence of human keypoints provides complementary information that tells the model which points are important to consider in this frame and which points are less valuable. The authors fill the missing values with the predicted results and perform the same procedure for the whole sequence.

The input dimension of this LSTM model is 12, which has four pairs of coordinates and confidence scores. The output comprises only the coordinates of the bat center, which are obtained through a fully-connected layer. The authors use the Mean Squared Error (MSE) loss as the loss function, defined as follows:

$$\mathcal{L}_{cor} = \frac{1}{N}\sum_{i=1}^{N}\left(\sqrt{x_{out}^2 + y_{out}^2} - \sqrt{\hat{x}^2 + \hat{y}^2}\right)^2, \tag{5}$$

where x_{out} and y_{out} are the output of this temporal prediction model, and $\hat{x}$ and $\hat{y}$ are the ground truth of the bat location. Considering different speed of bat movements, the video frame rate of this training set for the LSTM model is 60 fps, while the video frame rate of normal use of mobile phones is generally 30 fps and 60 fps, so this model meets the requirement of analysis in daily training.

Trajectory Completion and Reconstruction

After comprehensive experiments, the authors compare the effectiveness of traditional interpolation methods (Hagan & West, 2006) with their proposed temporal prediction model. The authors find that the interpolation methods have satisfactory completion results overall. Still, they perform poorly in capturing certain forward swing frames because of the fast-moving speed of bats in these frames. Therefore, they introduce a novel method that combines the cubic spline interpolation with their temporal prediction model to reconstruct complete bat trajectories.

The authors smooth and filter the wrist point horizontal coordinate curve for all sequence frames with a Butterworth filter in lowpass mode. The normalized cut-off frequency $_{ùn}$ is set to 0.3. This step allows the authors to find multiple wrist poles in the hitting phase. Subsequently, they define the forward swing sequences as consecutive and monotonically increasing series between each pair of neighboring local minima and maxima points. Multiple forward swing frames supply sequences of human body keypoints and bat positions, which are input into the time series prediction model and used as training data. The information from sequential frames of the forward swing is entered into the authors' trained missing frame prediction model for the bats. The input data consists of three components: the coordinates of the key points on the human body, the coordinates of the bats detected by YOLOv5, and the confidence scores of every point. The authors use the model to estimate the bat's position in the frame of the missing section and reconstruct its trajectory in the forward swing frame.

Cubic spline interpolation is a mathematical technique that estimates the values of a function between known data points. It is particularly useful when there is a set of discrete data points for creating a smooth curve that passes through those points. This method is widely used in various fields, including computer graphics, engineering, and numerical analysis. Regarding the remaining frames, the authors directly apply this cubic spline interpolation. Figure 6 depicts two different sequences. The whole trajectory of the table tennis bat is represented as $X \in \mathbb{R}^{T\times 2}$.

Occlusion is present but rare in a whole swing motion. When the bat is occluded, YOLO detection is impossible. But the keypoints of the human body can still be extracted. At this time, the time series prediction model or interpolation method is required to predict the bat position based on the previous frames.

Figure 5. Input and output data composition of the LSTM model

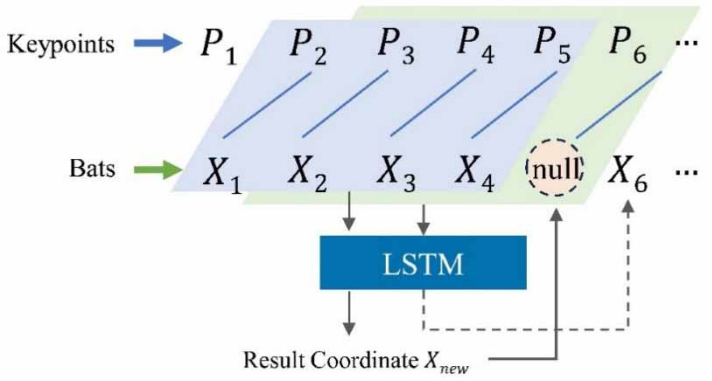


Figure 6. Two different sequences from a 30fps video clip

"Swing forward" represents the frames that comprise a complete forward swing action, and "Other" represents the remaining frames.

The developed method reduces the processing time and saves computational resources but also accurately reconstructs and fills the missing values in bat trajectories. The proposed method simplifies the computational workload and guarantees that the produced bat trajectories accurately reflect the original data, closely matching real-world observations.

EXPERIMENTAL EVALUATIONS

Datasets

The authors construct three table tennis video datasets with different types of bat postures and movements to evaluate the developed scheme. The following experiments utilize these datasets.

Table Tennis Sports Video Dataset

The authors collect daily training videos from table tennis teams, courses, and representative staff teams at the Beijing University of Posts and Telecommunications. They construct a collection of table tennis sports videos with various frame rates and camera views. This dataset contains 1,009 short videos that are approximately two seconds long, and 26 long videos are from 22 male and 9 female players. The videos are captured at different frame rates using mobile phones (30fps, 60fps, 240fps) and a stereo camera (120fps). The camera is set on one side of the table to capture motion videos of the players on their bat-holding side. The camera angle is either perpendicular to the table's long side, capturing both sides of the players' movements, or set at a predetermined angle relative to the table's long side, capturing the motion of one single player.

Bat Target Detection Dataset

The authors aim to build a table tennis bat object detection dataset involving multiple views and poses. Therefore, they selected segments from their table tennis video dataset and annotated a substantial number of bats using the RoboFlow Annotate tool. The dataset includes 3,754 images of bats with different shapes and sizes. This dataset provides ample and effective raw data for model training and optimization. Figure 7 depicts some examples from the bat dataset.

Swing Sequences Dataset

This dataset contains 1,596 frame sequences of forward swing actions with labeled bat locations and human body keypoint information in each frame. In all generated sequences, the coordinates and confidence scores are normalized. Specifically, the authors extract multiple consecutive frames

of forward swing from the above table tennis sports video dataset, annotate the bat by RoboFlow, and extract human keypoints using Mediapipe (Bazarevsky et al., 2020). This dataset is created for training and optimizing the LSTM-based bat trajectory temporal prediction model. As mentioned in Section 3.3, the authors' approach involves using the temporal prediction model within the swing-forward frames and employing an interpolation algorithm in other frames. Consequently, this dataset is solely utilized for the training phase. The authors apply their method to the entire swing sequence during the testing stage.

The yellow bounding boxes target bats from various angles and sights.

Baselines

Current literature does not study bat trajectory recognition and tracking in sports videos. The few existing methods focus on table tennis robots filmed from the front, which differs from this application scenario. For the bat identification problem addressed in this paper, the authors compare three classic approaches as baselines, namely the Lucas-Kanade sparse optical flow algorithm as the traditional optical flow method, YOLOv8-seg as the instance segmentation method, and the object detection method YOLOv5. These models are fast to execute and possess a certain object-tracking capability. Next, the authors briefly introduce these baseline methods.

Figure 7. Bat target detection dataset annotated in Roboflow

Gunnar Farneback Optical Flow Algorithm

The Gunnar Farneback optical flow algorithm (Farnebäck, 2003) is a classic two-frame differential optical flow estimation algorithm that calculates an object's trajectory through optical flow. It uses polynomial expansion to estimate the optical flow field by fitting a local quadratic polynomial to the image brightness. In sports video scenes with people as the main subjects, the movement of the bat is more obvious. Thus, the intensity information of the bat's optical flow is relatively strong. Specifically, the authors filter out 10 points with the highest brightness in each frame and take the point closest to the wrist as the center of the bat's surface.

YOLOv8-seg

The YOLOv8-seg model is an instance segmentation model based on YOLACT (Bolya et al., 2019). It is simple to deploy and easy to train using self-defined data and attains excellent performance. The authors use the Labelme tool to label more than 1,100 segmented images of the bat manually. Then, the authors train the segmentation model on the pre-trained YOLOv8m-seg.pt weights using their labeled images. They calculate the midpoint of the coordinates of the segmentation results in each frame as the complete bat trajectory.

YOLOv5

YOLOv5 is a single-stage object detection algorithm with a certain tracking and recognition ability. Using the bat target detection dataset, the authors train their bat object detection model on the pre-trained YOLOv5s.pt weights. The authors set the batch size to 16 and trained the model for 500 epochs.

Comparison Methods

Spurred by the evaluation results shown in Table 1, the authors choose YOLOv5 as the baseline of their approach. They build several methods on it, each optimized based on the previous one. Next, they briefly introduce each method.

YOLOv5S + Interpolation

Based on the YOLOv5 results, the authors design a filtering algorithm to find the wrong detection using the two rules presented in Section 3.1 and elevate the detection precision to nearly one. The authors name this combined method as YOLOv5S. They apply only cubic spline interpolation for bat trajectory reconstruction, but this method yields unsatisfactory results for certain frames in swing-forward sequences.

YOLOv5S + Social-LSTM

Social-LSTM (Alahi et al., 2016) is a trajectory prediction model that leverages surrounding pedestrians' information to share movement patterns and predict all pedestrian trajectories. The authors simplify the model architecture and train it on their dataset with human keypoint information in each frame for 300 epochs. They attempt to use this model alone to predict all the missing trajectories without the interpolation method.

YOLOv5S + LSTM

Considering the complexity of Social-LSTM (Alahi et al., 2016) and the limitations of its application scenarios, the authors design an extension based on it to leverage the key idea of using surrounding related information on classic LSTM (Hochreiter & Schmidhuber, 1997). They introduce human body keypoints coordinates from the shoulder, elbow, and wrist (12, 14, 16 points) and the bat location as input and train on a table tennis sports video dataset for 300 epochs.

Performance Metrics and Evaluation Results

Performance Metrics

- Precision (P) represents the proportion of true positive bats in YOLO detection results, calculated as $P = \frac{TP}{TP + FP}$, where TP is the number of true bat frames and TP + FP is the number of frames in segmentation and detection results. This metric is used only on the baselines. After error filtering, the authors successfully elevated it to 1.
- Recall (R) represents the proportion of correct bat positions predicted in a trajectory, which is calculated as $R = \frac{TP}{TP + FN}$, where TP + FN is the number of frames in a video clip because each frame exactly has a bat in it.
- Intersection Over Union (IoU) represents the overlap ratio between the predicted and the real bounding box. This better represents the degree of overlap between the two boxes. The size of both boxes is 100*100, centered on the prediction and ground truth coordinates. Given the occurrences of undetected bats in YOLOv8-seg and YOLOv5 results, frames with undetected objects are treated as having zero IoU. Finally, the authors compute the average of the IOU of all frames.
- Distance From the Center Point (D) represents the Euclidean distance between the prediction and ground truth centers. The authors only consider the frames where bats are detected, including the ones with false bat detection, and take the average distance results of these frames.

Evaluation Results

The authors select five videos from their table tennis sports dataset, including professional and amateur players' forehand attacks from two scenarios. To standardize the measurement, the authors extracted two seconds from each video segment that consisted of approximately 120-140 frames. The authors first compare the baselines on the test dataset, with the corresponding results reported in Table 1.

Optical flow requires relatively modest computational resources, but its performance is the least satisfactory. For each subsequent frame, the authors calculate the optical flow relative to the previous frame, including the magnitude and angle of the flow. Next, they identify the locations closest to the wrist keypoints with the highest optical flow intensity. They observe that the optical flow of the pixels near the bat in sports videos is not prominent and cannot be tracked effectively. Indeed, at times, the optical flow intensity around the bat is weaker than that of the more noticeable motion of the table tennis ball. Thus, determining the bat's position by searching for the bright points may lead to significant errors. Although YOLOv8-seg is significantly better than optical flow, it still underperforms. This is because, first, both precision and recall scores are relatively low, as there

Table 2. Trajectory estimation and reconstruction results of baselines on test datasets

Baselines	Farneback				YOLOv8-seg				YOLOv5			
	P	R	IoU	D	P	R	IoU	D	P	R	IoU	D
set1	0.812	0.605	0.254	131.895	0.963	0.684	0.568	17.278	0.938	0.789	0.755	4.515
set2	0.893	0.739	0.353	81.529	0.948	0.791	0.658	16.234	1	0.739	0.75	3.521
set3	0.598	0.528	0.209	134.044	0.713	0.685	0.581	291.097	1	0.85	0.791	4.735
set4	0.857	0.537	0.187	125.69	1	0.822	0.613	12.197	0.974	0.843	0.731	8.25
set5	0.912	0.796	0.41	51.399	0.954	0.71	0.552	18.539	1	0.741	0.651	5.328
total	0.814	0.641	0.283	104.911	0.916	0.738	0.594	71.069	0.982	0.792	0.736	5.27

are some false and missed detections during the segmentation process, hindering the subsequent experiments. Second, the model has limited robustness, leading to considerable errors, especially on test set 3. Therefore, additional annotations of the bat's surface from different sizes and angles are required to enhance the model's effectiveness. Unlike object detection bounding boxes, segmentation annotations are more time-consuming and labor-intensive. In conclusion, YOLOv5 is more suitable as a baseline. However, since the precision score in YOLOv5 is not always 1, the filtering algorithm is essential to eliminate the wrong detections and increase the precision score to 1.

The authors evaluate the compared methods and their method on the test dataset. For convenience in observing the experimental results, they center each image around the bat and crop it to a size of $100*100$. Regarding metrics P and R, if the bat appears fully within the image, they consider the bat center position correct. The corresponding results are reported in Table 2.

The precision of YOLOv5 is 0.982, highlighting the requirement for error filtering. After setting up rules to filter out false detection results, the precision of YOLOv5 was raised to 1. The results in Table 2 indicate that the proposed method that fuses human body keypoint information achieves excellent performance. The interpolation method performs exceptionally well, so the authors apply it to their work. After incorporating confidence scores, the authors' method exhibits superior performance across diverse test sets compared to other approaches, notably outperforming the interpolation method on average in the R metric. The IoU and D scores are comparable to the cubic spline interpolation method.

The authors compare the results of the proposed method to the ground truth value and draw a comparative line graph of bat position for 10 consecutive forward swing frames from these five test videos. Figure 8 illustrates the corresponding chart, highlighting that the bat trajectory restored by the proposed method is closer to the real trajectory, but there is still some error. Video 1 has the smallest error, while video 5 has the largest. This is because the bat target detection in the first stage of video 1 is good, and the shooting angle is 45° facing the athlete, which is consistent with the angle of the previously trained temporal prediction model, so the trajectory reconstruction is satisfied. The shooting viewpoint of video 5 is on the athlete's side, so the bat's trajectory during the swing differs from the training model. Although video 1 and video 5 have the same frame rate of 60fps, the swinging speed of the person in video 5 is faster than that in video 1. This produces more blurred bat images and poorer quality, leading to less effective tracking. Figure 9 depicts the bat bounding boxes of this method and ground truth for 10 consecutive forward swing frames of test video 1 and video 2, which suggests that training the model utilizing multi-view swing forward bat time series could improve the effectiveness and performance of the bat object detection model.

The idea of the authors' method can be applied to other similar swing motion scenes (J. Wu et al., 2021). It is necessary to train different detection models and time series prediction models for different types of bats. However, the integration of human body keypoints is similar. The authors shoot the table tennis videos of athletes wearing different color jackets, different positions, and different illumination angles. These videos are used in different training sessions. The five test videos are

Table 3. Trajectory estimation and reconstruction result of the authors' approach and compared methods on test datasets

Methods	YOLOv5S+I			YOLOv5S+SL			YOLOv5S+L			Ours		
	R	IoU	D	R	IoU	D	R	IoU	D	R	IoU	D
set1	0.921	0.866	6.325	0.877	0.711	40.727	0.86	0.666	46.946	1	0.866	6.213
set2	0.957	0.884	5.267	0.878	0.801	19.301	0.922	0.842	11.185	0.983	0.889	5.648
set3	1	0.878	5.338	0.929	0.793	84.371	0.937	0.806	20.582	0.976	0.853	7.609
set4	1	0.829	8.953	0.935	0.806	89.236	0.94	0.783	42.615	0.993	0.821	9.025
set5	0.898	0.882	10.219	0.834	0.724	30.254	0.86	0.751	25.47	0.891	0.818	10.721
total	0.955	0.868	7.22	0.891	0.767	52.778	0.904	0.77	29.36	0.969	0.849	7.843

Figure 8. Comparative line graph of bat position of the authors' method and ground truth for 10 consecutive forward swing frames from five test videos

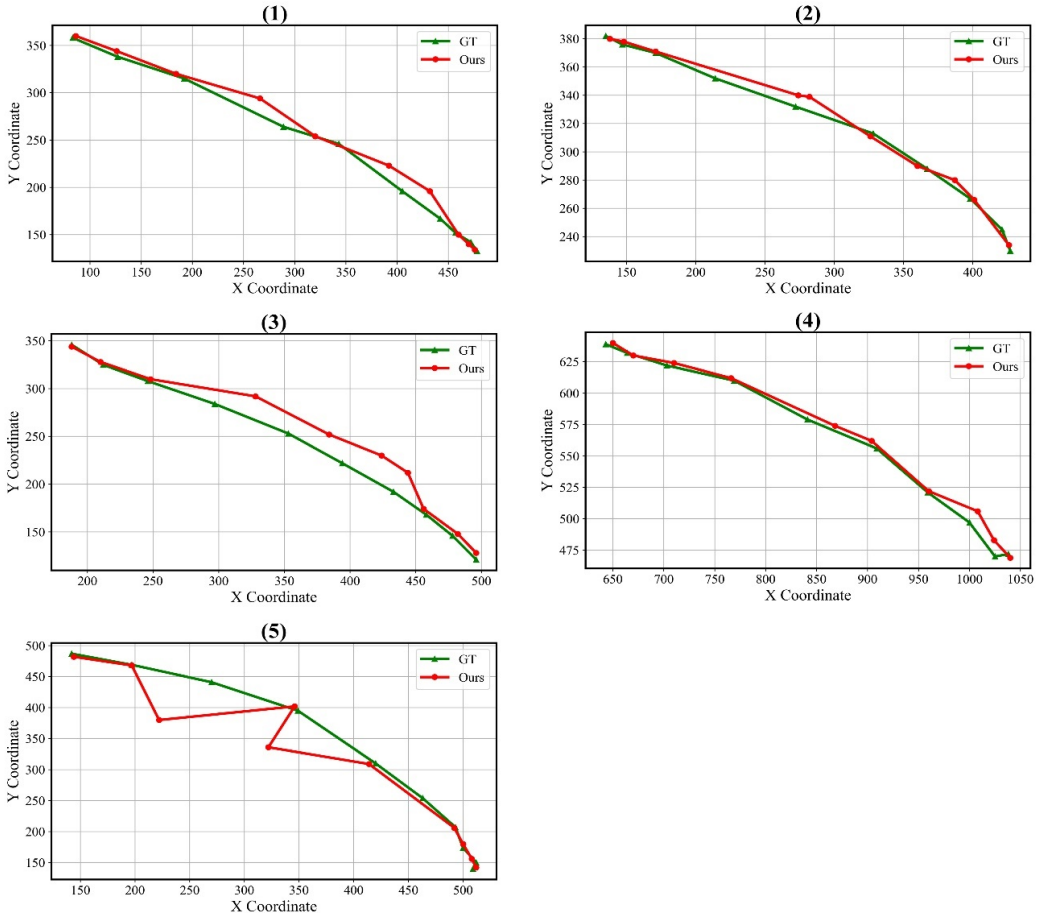
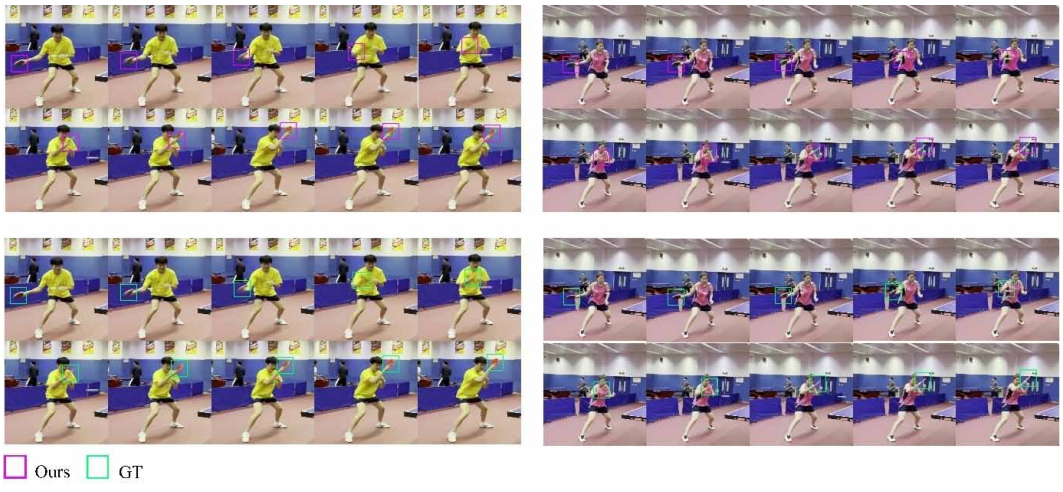


Figure 9. Bat bounding boxes of the authors' method and ground truth (abbreviated as GT) for 10 consecutive forward swing frames in Test Video 1 and 2



different in terms of characters, environment, and lighting, and the results on these test sets show that this method has a good performance under varied lighting and background conditions typical in table tennis match environments. The real-time processing ability of the method is a very important aspect. The method contains two models, where YOLO's inference time for each image is as fast as 0.007 seconds, meaning 140 frames per second (FPS). Mediapipe Pose can reach the speed of 30 FPS with only CPU. Other processing of the time series prediction model and other operations are consistent and extremely rapid. Therefore, this method has better real-time processing performance.

CONCLUSION

This paper proposes a novel table tennis bat trajectory reconstruction method that fuses human body keypoints and confidence scores. Based on object detection, it combines a temporal prediction model and an interpolation method. In addition, the authors built multiple table tennis bat-related datasets, providing a rich resource for their training and testing. The comprehensive experimental evaluation results demonstrated the excellent performance of the proposed method in multiple scenarios compared to existing classic methods. Based on the proposed method, the output bat trajectories of table tennis training and competition videos can assist subsequent analysis tasks, such as hitting point evaluation and serving as a skill guide.

Considering the ease of implementation and use for coaches and sports analysts, this method can be further optimized and designed as a sports guidance app for coaches. They only need to shoot or upload a video from a specific angle to visualize the bat trajectory. This work can be mainly combined with the tracking trajectory of table tennis analyze the relevant indicators of hitting points. In future work, the authors are willing to try in this area. In terms of usage scenarios and functions, this approach has the potential for adapting the method for trajectory reconstruction in other sports that involve fast-moving objects. Though the scenes of these two sports are more open, so the angle and position of the recorded video need to be reconsidered. The idea of this method can be used for reference.

GCN models are approaches that can be applied to human critical point detection. Models such as (Jang & Lee, 2021; Yan et al., 2018) have extended graph neural networks to spatial-temporal graph models to obtain spatial and temporal characteristics of keypoints in the human body. At the present stage, the authors' primary research focuses on the motion trajectory of the bat object in sports videos rather than the human body. Solving the identification and tracking of the bat is of utmost importance for the scenario examined in this paper. However, GCN can effectively capture complex spatial structures and temporal dependencies in action sequences by constructing temporal graphs. For future work, the authors will introduce advanced models like Graph Convolutional Networks to improve the capability of trajectory construction and validate and improve the proposed method on datasets with more videos.

REFERENCES

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human trajectory prediction in crowded spaces. *Proceedings of the CVPR*, 961-971. doi:10.1109/CVPR.2016.110

AlShami, A., Boult, T., & Kalita, J. (2023). Pose2Trajectory: Using transformers on body pose to predict tennis player's trajectory. *Journal of Visual Communication and Image Representation*, *97*, 103954. doi:10.1016/j.jvcir.2023.103954

Alsmirat, M. A., Al-Alem, F., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2019). Impact of digital fingerprint image quality on the fingerprint recognition accuracy. *Multimedia Tools and Applications*, *78*(3), 3649–3688. doi:10.1007/s11042-017-5537-5

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). BlazePose: On-device real-time body pose tracking. *CoRR 2020, abs/2006.10204*.

Behera, T. K., Bakshi, S., Sa, P. K., Nappi, M., Castiglione, A., Vijayakumar, P., & Gupta, B. B. (2023). The NITRDrone dataset to address the challenges for road extraction from aerial images. *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, *95*(2-3), 197–209. doi:10.1007/s11265-022-01777-0

Benmoussa, K., Hamdadou, D., & Roukh, Z. E. A. (2022). GIS-based multi-criteria decision-support system and machine learning for hospital site selection: Case study Oran, Algeria. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–19. doi:10.4018/IJSSCI.285592

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9157-9166.

Bouguet, J. Y. (2001). Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. *Intel Corporation*, *5*, 4.

Boyer, C., Bulmus, V., Davis, T. P., Ladmiral, V., Liu, J., & Perrier, S. (2009). Bioapplications of RAFT polymerization. *Chemical Reviews*, *109*(11), 5402–5436. doi:10.1021/cr9001403 PMID:19764725

Bruhn, A., Weickert, J., & Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, *61*(3), 211–231. doi:10.1023/B:VISI.0000045324.43199.43

Cai, G. L. (2022). A method for prediction the trajectory of table tennis in multirotation state based on binocular vision. *Computational Intelligence and Neuroscience*, *2022*, 1–10. doi:10.1155/2022/8274202 PMID:35463267

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6154–6162. doi:10.1109/CVPR.2018.00644

Cao, J., Chen, Q., Guo, J., & Shi, R. (2020). Attention-guided context feature pyramid network for object detection. *CoRR 2020, abs/2005.11475*.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299. doi:10.1109/CVPR.2017.143

Chen, G., Xu, D., Fang, Z., Jiang, Z., & Tan, M. (2013). Visual measurement of the racket trajectory in spinning ball striking for table tennis player. *IEEE Transactions on Instrumentation and Measurement*, *62*(11), 2901–2911. doi:10.1109/TIM.2013.2265471

Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Yan, Y. (2020). Blendmask: Top-down meets bottom-up for instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8573–8581. doi:10.1109/CVPR42600.2020.00860

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. doi:10.1109/TPAMI.2017.2699184 PMID:28463186

Chen, X., Girshick, R., He, K., & Dollár, P. (2019). Tensormask: A foundation for dense object segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2061–2069. doi:10.1109/ICCV.2019.00215

Dai, J., He, K., Li, Y., Ren, S., & Sun, J. (2016). Instance-sensitive fully convolutional networks. *Proceedings of the Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, 534–549.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2758-7766. doi:10.1109/ICCV.2015.316

Dwivedi, R. K. (2022). Density-based machine learning scheme for outlier detection in smart forest fire monitoring sensor cloud. *International Journal of Cloud Applications and Computing*, *12*(1), 1–16. doi:10.4018/IJCAC.305218

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. *Proceedings of the Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, 363–370. doi:10.1007/3-540-45103-X_50

Furuya, R., Yokoyama, H., Dimic, M., Yanai, T., Vogt, T., & Kanosue, K. (2021). Difference in racket head trajectory and muscle activity between the standard volley and the drop volley in tennis. *PLoS One*, *16*(9), e0257295. doi:10.1371/journal.pone.0257295 PMID:34520488

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. doi:10.1109/CVPR.2014.81

Girshick, R. B. (2015). Fast R-CNN. *Proceedings of the ICCV*, 1440–1448.

Gu, W., Bai, S., & Kong, L. (2022). A review on 2D instance segmentation based on deep neural networks. *Image and Vision Computing*, *120*, 104401. doi:10.1016/j.imavis.2022.104401

Guo, C., Fan, B., Zhang, Q., Xiang, S., & Pan, C. (2020). Augfpn: Improving multi-scale feature learning for object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12595–12604. doi:10.1109/CVPR42600.2020.01261

Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: State of the art. *International Journal of Multimedia Information Retrieval*, *9*(3), 171–189. doi:10.1007/s13735-020-00195-x

Hagan, P. S., & West, G. (2006). Interpolation methods for curve construction. *Applied Mathematical Finance*, 13.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9. PMID:9377276

Hu, B., Gaurav, A., Choi, C., & Almomani, A. (2022). Evaluation and comparative analysis of semantic web-based strategies for enhancing educational system development. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–14. doi:10.4018/IJSWIS.302895

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2462–2470. doi:10.1109/CVPR.2017.179

Jang, H. B., & Lee, C. W. (2021). ST-GCN based human action recognition with abstracted three features of optical flow and image gradient. *Proceedings of the Frontiers of Computer Vision: 27th International Workshop, IW-FCV 2021, Daegu, South Korea, February 22–23, 2021, Revised Selected Papers 27*, 203–217. doi:10.1007/978-3-030-81638-4_17

Ji, Y., Hu, X., Chen, Y., Mao, Y., Wang, G., Li, Q., & Zhang, J. (2021). Model-based trajectory prediction and hitting velocity control for a new table tennis robot. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2728-2734. doi:10.1109/IROS51168.2021.9636000

Kim, W., Son, B., & Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. *Proceedings of the International Conference on Machine Learning. PMLR*, 5583–5594.

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019a). Panoptic feature pyramid networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6399–6408.

Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020a). Pointrend: Image segmentation as rendering. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9799–9808.

Li, D., Deng, L., Bhooshan Gupta, B., Wang, H., & Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, *479*, 432–447. doi:10.1016/j.ins.2018.02.060

Li, H., Ali, S. G., Zhang, J., Sheng, B., Li, P., Jung, Y., Wang, J., Yang, P., Lu, P., Muhammad, K., & Mao, L. (2022). Video-based table tennis tracking and trajectory prediction using convolutional neural networks. *Fractals*, *30*(05), 2240156. doi:10.1142/S0218348X22401569

Li, S., Qin, D., Wu, X., Li, J., Li, B., & Han, W. (2022). False alert detection based on deep learning and machine learning. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–21. doi:10.4018/IJSWIS.313190

Li, W., Liu, K., Zhang, L., & Cheng, F. (2020). Object detection based on an adaptive attention mechanism. *Scientific Reports*, *10*(1), 11307. doi:10.1038/s41598-020-67529-x PMID:32647299

Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017). Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359–2367.

Lin, T. Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., & Belongie, S. J. (2016a). Feature pyramid networks for object detection. *CoRR 2016, abs/1612.03144*.

Lin, T. Y., Goyal, P., Girshick, R. B., He, K., & Dollár, P. (2017a). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

Ling, Z., & Hao, Z. J. (2022). An intrusion detection system based on normalized mutual information antibodies feature selection and adaptive quantum artificial immune system. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–25. doi:10.4018/IJSWIS.308469

Liu, P., & Wang, J. H. (2022). MonoTrack: Shuttle trajectory reconstruction from monocular badminton video. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3513-3522. doi:10.1109/CVPRW56347.2022.00395

Llanos, M. J., Obrero, J. R., Alvarez, L. M., Yang, C. H., & Aliac, C. J. (2022). Computer-assisted table tennis posture analysis using machine learning. *Proceedings of the 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 1–6. doi:10.1109/IICAIET55139.2022.9936806

Pan, X., Yamaguchi, S., Kageyama, T., & Kamilin, M. H. B. (2022). Machine-learning-based white-hat worm launcher in botnet defense system. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–14. doi:10.4018/IJSSCI.291713

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the CVPR*, 779–788. doi:10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference,* Munich, Germany*,* October 5-9, 2015*, Proceedings*, 234–241.

Rozumnyi, D., Matas, J., Sroubek, F., Pollefeys, M., & Oswald, M. R. (2020a). FMODetect: Robust detection and trajectory estimation of fast moving objects. *CoRR 2020, abs/2012.08216.*

Rozumnyi, D., Oswald, M. R., Ferrari, V., Matas, J., & Pollefeys, M. (2021a). DeFMO: Deblurring and shape recovery of fast moving objects. *Proceedings of the CVPR*, 3456–3465. doi:10.1109/CVPR46437.2021.00346

Shibata, S., Hirose, K., Naruo, T., & Shimizu, Y. (2023). Estimation of baseball bat trajectory during a practice swing using a Kalman filter for velocity compensation. *Proceedings of the Institution of Mechanical Engineers. Part P, Journal of Sports Engineering and Technology*, *237*(2), 96–101. doi:10.1177/1754337119871436

Tao, R., Gavves, E., & Smeulders, A. W. M. (2016). Siamese instance search for tracking. *Proceedings of the CVPR*, 1420–1429.

Tembhurne, J. V., Almin, M. M., & Diwan, T. (2022). Mc-DNN: Fake news detection using multi-channel deep neural networks. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–20. doi:10.4018/IJSWIS.295553

Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9627–9636.

Voeikov, R., Falaleev, N., & Baikulov, R. (2020). TTNet: Real-time temporal and spatial video analysis of table tennis. *Proceedings of the CVPR*, 3866–3874. doi:10.1109/CVPRW50498.2020.00450

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475. doi:10.1109/CVPR52729.2023.00721

Wang, H., Li, Z., Li, Y., Gupta, B. B., & Choi, C. (2020). Visual saliency guided complex image retrieval. *Pattern Recognition Letters*, *130*, 64–72. doi:10.1016/j.patrec.2018.08.010

Wang, K., Liew, J. H., Zou, Y., Zhou, D., & Feng, J. (2019). Panet: Few-shot image semantic segmentation with prototype alignment. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9197–9206. doi:10.1109/ICCV.2019.00929

Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). Solo: Segmenting objects by locations. *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference,* Glasgow, UK*,* August 23–28, 2020*, Proceedings*, 649–665.

Wu, E., & Koike, H. (2020). Futurepong: Real-time table tennis trajectory forecasting using pose prediction network. *Proceedings of the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–8. doi:10.1145/3334480.3382853

Wu, J., Liu, D., Guo, Z., Xu, Q., & Wu, Y. (2021). TacticFlow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics*, *28*(1), 835–845. doi:10.1109/TVCG.2021.3114832 PMID:34587062

Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., & Luo, P. (2020). Polarmask: Single shot instance segmentation with polar representation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12193–12202. doi:10.1109/CVPR42600.2020.01221

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence,* 32. doi:10.1609/aaai.v32i1.12328

Yu, C., Li, J., Li, X., Ren, X., & Gupta, B. B. (2018). Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimedia Tools and Applications*, *77*(4), 4585–4608. doi:10.1007/s11042-017-4637-6

Zhou, Q. (2022). A study on human transiting based on big data and web semantics: Distinguishment and detection. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–18. doi:10.4018/IJSWIS.310055

Zechen Jin is currently a student working toward the master's degree in Physical Education and Training (PEET) at Beijing Sport University (BSU). His research interests include analysis of table tennis techniques and tactics, and intelligent sports. Jun Liu, an associate professor, is the director of the Center for Data Science, Beijing University of Posts and Telecommunications (BUPT). He received his B.E and Ph.D. degrees from Department of Information Engineering, BUPT in 1998 and 2003, respectively. His research interests include artificial intelligence, big data analysis, and stream data algorithms.

Yang Yu, an associate professor of Beijing Sport University, is former coach of the national table tennis men's team, master's tutor, doctor of sports training of Beijing Sport University. His research direction is table tennis theory and practice.