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

A Semantic Tree-Based Fast-Moving Object Trajectory Tracking Algorithm for Table Tennis

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

Tianjian Zou, Beijing University of Posts and Telecommunications, China

Dazhuang Sun, Beijing University of Posts and Telecommunications, China

Yu Yang, Beijing Sport University, China*

Jun Liu, Beijing University of Posts and Telecommunications, China

ABSTRACT

Table tennis is a popular sport around the world. A key technology in table tennis education and analysis system is reconstructing the trajectory of the fast-moving ball from videos. Typically the table tennis ball is too small and barely visible in the video, making it difficult to be recognized directly by detection models like YOLO. However, table tennis balls usually has obvious motion features, which are usually not found in similar false targets. It inspired the authors to first find all candidate targets and then use the motion features of table tennis ball to select them out. In this article, the authors propose a tree-based algorithm named T-FORT to track the ball and reconstruct its trajectory. Specifically, they consider all the possible objects in a tree-framework, and identify the real target by integrating visual features and moving patterns. The authors conduct a set of experiments on three datasets to evaluate the effectiveness and performance of the proposed algorithm. The experimental results show that the proposed method is more precise than existing algorithms, and is robust in various scenarios.

KEYWORDS

Object Tracking, Table Tennis, Video Analysis

INTRODUCTION

In sports like table tennis, badminton, or baseball, tracking the fast-moving ball is a key technical problem for following analysis tasks. With the results of ball tracking, the analysis program could understand the process of a game, a round, or even a stoke. For example, with the tracking result, the analysis program could evaluate if the hit-point is proper in a training or competition. In addition, the tracking result can also help estimate the flying speed of the ball to evaluate the training effect of a professional athlete. In a number of professional situations, researches usually use radar or high-

DOI: 10.4018/IJSWIS.337320
*Corresponding Author

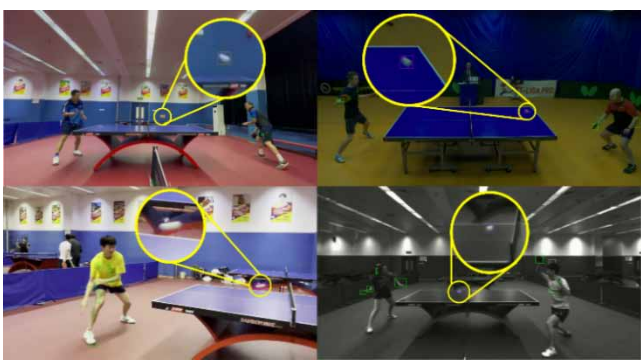
speed camera to track the fast-moving balls. However, these kinds of methods are difficult to widely use because of high cost and complexity. Towards this end, we aim to propose an algorithm to track the fast-moving table tennis ball based on images captured by commodity video camara, such as mobile phones. Such algorithms enable the development of web services and features for table tennis analysis, allowing users to enjoy accurate and professional analysis and guidance using mobile clients.

The basic idea of existing mainstream object tracking methods is finding matched objects after object detection in each frame. However, it is almost impossible to directly apply this kind of method for detecting the table tennis ball because it doesn't have stable appearance and features, owing to its fast-moving state. On the other hand, the table tennis ball exhibits typical moving patterns like a shooting star in a video. As it is moving rapidly, the algorithm can preliminarily separate the table tennis ball, such as balls in Fig. 1, from the background by computing the difference of frame images (Rozumnyi et al., 2017). While not all the connected regions from separation are the table tennis ball, some of them are noises caused by movement of people or other objects. A proper algorithm should consider all the situations to distinguish the real trajectory of a ball.

To solve this problem, we propose a tree-based fast-moving object trajectory tracking (T-FORT) algorithm based on tree pruning and tracking strategies. The first step of the proposed algorithm is searching candidate regions based on differential images. Then, a tracking tree is formulated by the candidate regions. Because the candidate regions may contain a number of fake balls, we design a tree pruning process to filter fake objects to identify real balls. At last, the trajectory is restored based on a growing tracking tree process. To evaluate the effectiveness and performance of the proposed algorithm, we perform the proposed algorithm and other baseline methods on a self-built dataset and two open datasets. The experimental result demonstrates the excellent performance and robustness of the proposed algorithm. Our algorithm creatively combines tree structures and pruning algorithms to analyze all possible tracking results, greatly improving tracking recall and tracking performance.

The rest of this paper is structured as follows. Section 2 introduces the related work. In Section 3, we introduce the proposed method in details. In Section 4, we evaluate the proposed algorithm by experiments, based on a self-built dataset and two open source datasets, and explain the evaluation results. In Section 5, we conclude the work and point out future works.

Figure 1. Example of table tennis ball to be tracked



RELATED WORKS

In 2017, the problem of tracking the "Fast Moving Object" in a table tennis scenario was first addressed in Rozumnyi et al. In this paper, the authors used binary differential images to search for possible candidate regions and then analyze each candidate region to identify table tennis balls. Their work can track table tennis balls in videos with a clean background. In this situation, the number of easily confused targets in differential images is relatively small. In addition, this algorithm cannot tolerate losing targets, which often happens in reality. In their following research works (Rozumnyi et al., 2021; Zita & Šroubek 2021), they try to introduce deep learning methods to improve the accuracy of detectors rather than improving the performance of tracking algorithms. To some extent, our work is inspired by the above works. However, we think that the fast-moving object tracking task, a very special object tracking problem, cannot be solved by the straightforward approach of applying deep learning method just using image features and semantic information.

In the previous research works on processing of images with moving objects, the first problem to be solved was to remove the blurring appearance or other noise of the moving objects in the video (Rozumnyi et al., 2021; Sagawa et al., 2009; Lu, 2018), (Qian et al., 2022; Chu et al., 2022; Zheng et al., 2022; Chopra et al., 2022; Li et al., 2019)(Almomani et al., 2022). Most of these methods are valid for tracking objects with normal size (Lee, 2015; Xiao et al., 2015; Al-Jilani et al., 2019). However, it is very hard to solve the blurring problem of a fast-moving table tennis ball because of its tiny scale and unnotable appearance. In this kind of situation, a number of studies have proposed target tracking methods based on correlation filters (Kotera et al., 2020; Wang & Zhang, 2019). Some researchers have combined correlation filters with Kalman filters to estimate the target's position, thereby improving tracking performance and accelerating the algorithm's speed (Adhikari, 2016). These methods work well when the target has features that are clearly different from the background. However, for a fast-moving table tennis ball, owing to its small size and irregular features, it is difficult to establish a stable model for correlation filters.

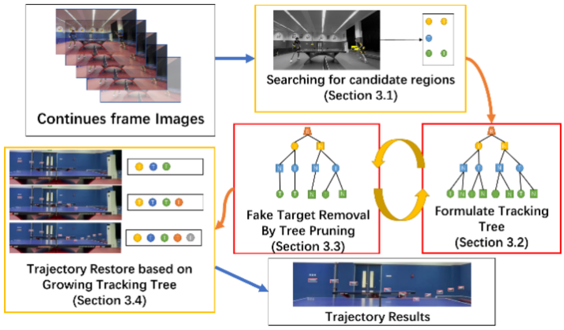
In the tracking problem of fast-moving targets, the image background remains relatively stable compared to the fast moving target, while the pixels around the fast-moving target in the image tend to have significant changes. Therefore, some reasearchers proposed to use differential images to solve the target tracking problem. In Sun et al., (2017) differential images are used to obtain the boundary region of a target. By fusing the above regions with the depth image, the entire target can be segmented. However, this kind of method may fail when directly applied on small target tracking because of the small size of the target. The differential image may contain fake objects because of noise and other factors, which are extremely similar to the target, making it difficult to distinguish the true target.

In recent years, significant progress has been made in object detection methods and other image processing tasks based on deep learning (Sun et al., 2020; He et al., 2021; Dai et al., 2021; Dong et al., 2019) (Wang et al., 2022; Shen & Saab 2021; Alsmirat et al., 2019; Yu et al., 2018; Saab & Jaafar, 2021; Al-Sobbahi & Tekli, 2022), making it less difficult to directly detect fast-moving targets. However, detecting the target directly from the entire image still poses a great challenge. Some researchers choose to crop part of the image and detect the target from the cropped small image. In Zhang et al., (2020), researchers use a Kalman filter based on moving informations to predict the most likely regions of the target. In this way, the algorithm can avoid the interference of noise from other regions on the detection results and achieve accuracy improvement (Voeikov, 2020; Lu, 2020). However, this method cannot handle situations with low contrast between background and target such as indoor environments. Meanwhile, the unpredictable trajectory and non-distinctive image features of table tennis balls still pose a huge challenge to methods based on deep learning. Inspired by works (Benrazek et al., 2023; Nhi & Le, 2022), we propose to leverage the tree-framework to integrate the visual features and moving patterns of table tennis balls to solve this prolem.

DESIGN OF THE PROPOSED ALGORITHM

The proposed algorithm has four parts, and we will introduce it step by step in the following sections. The overall framework and process of the proposed algorithm are shown in Fig. 2.

Figure 2. The overall framework and process of the proposed algorithm



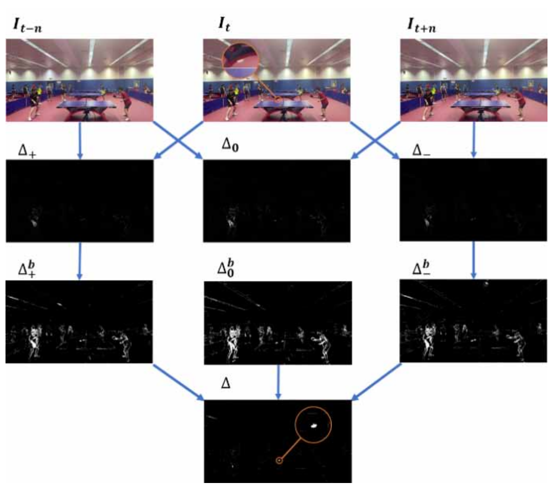
Searching Candidate Regions Based on Differential Images

Owing to the rapid movement of table tennis balls, it tends to leave complete features and regular connected regions in differential images, while other slowly moving targets often leave irregular connected regions. These kind of phenomenons inspire us to design the first step of searching candidate regions based on deffential images.

To identify the candidate connected regions of table tennis in video frame It, we take consecutive image frames It−1, It, It+1 as input. Then, three binary images $\Delta_+^b, \Delta_-^b, \Delta_0^b$ are computed, where $\Delta_+ = |I_t - I_{t-1}|$, $\Delta_0 = |I_{t+1} - I_{t-1}|$, $\Delta_- = |I_t - I_{t+1}|$. The threshold to generate binary images is defined as $\mu + k\sigma$, where μ, σ and k are the average of $|I_t - I_{t-1}|$, the variance of $|I_t - I_{t-1}|$, and a configurable parameter, respectively. At last, we compute Δ according to $\Delta = \Delta_+^b \wedge \Delta_-^b \wedge \neg\Delta_0^b$. The binary image Δ contains the connected regions of all the moving objects, which are present in the frame It, but not in the frames It−1 and It+1 (i.e. moving objects in It). In this way, we get the binary image Δ.

In some situations, the table tennis ball may move slowly, or the camera capturing video uses a rolling shutter. It would result in incomplete connected regions in image Δ. This problem can be solved by using It−n, It, It+n to replace It−1, It, It+1 where n could be 2, 3 or even 4 in high frame rate videos. The complete computing process is shown in Fig. 3.

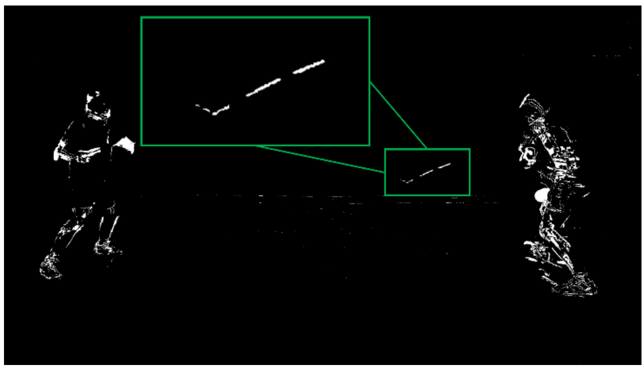
Figure 3. The process of computing Δ



Formulating the Tracking Tree

The step in section 3.1 produce a set of candidate regions that would be a table tennis ball. However, it is hard to judge if a candidate region is a ball by the information in a single frame. With the careful study of numerous videos, we found that the fast-moving table tennis balls exhibit consistent trajectories in continous video frames. An example is shown in Fig. 4. We can leverage this feature to identify balls with higher accuracy.

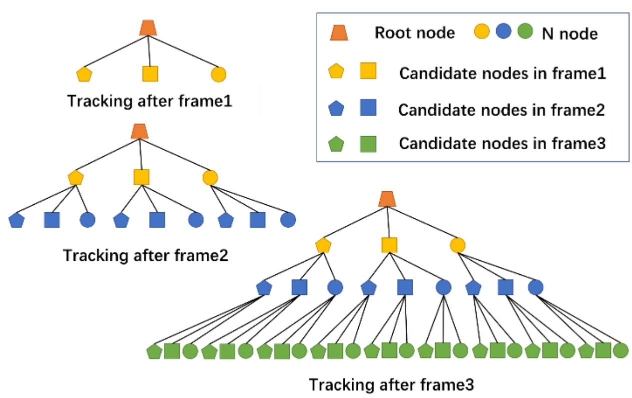
Figure 4. Ball trajectory may be clear after adding $\Delta_{t-1}, \Delta_t, \Delta_{t+1}$, which are from video frames



To identify all possible ball trajectories by leveraging interrelated information in multiple video frames, we design a tree structure named Tracking Tree. An example is shown in Fig. 5. Each deepest node in the Tracking Tree represents a trajectory at its layer *t*. For each node in the tree, it may have a candidate, which can be called Cand-node, or just vacant, which can be called N-node. The nodes represents a trajectory have the corresponding candidate or have no object at t_i, where t_i is the time represented by node's layer.

Specifically, a completed Tracking Tree can be formulated using the following steps, shown in Fig. 5:

Figure 5. The process of formulating a tracking tree



(1) Start with a root Node;
(2) Analyze a new frame at time *t* by the process of section 3.1, and get candidates-list that have *n* targets;
(3) For every node in the deepest layer, add *(n+1)* nodes as its child. *n* nodes (these n nodes are Cand-node) are one-to-one correspondence to *n* nodes and the one more vacant node (this is a N-node);
(4) After Step 3, the tree will have one more layer, and its depth add one. Then, continue to do step 2 and 3 until the end of the video.

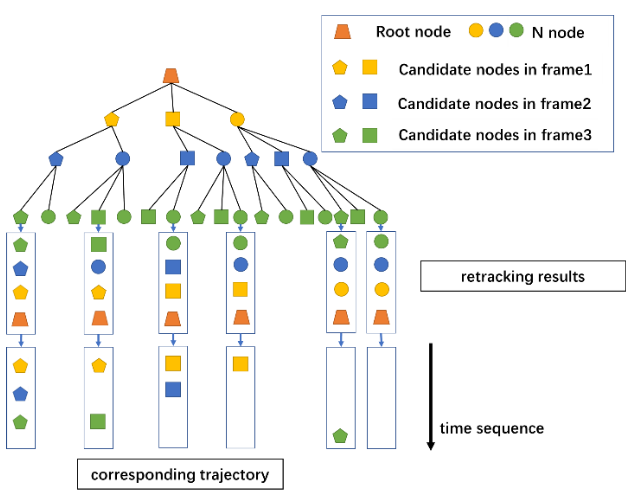
Based on the formulating process of a tracking tree, we can have a formal definition of trajectory: A trajectory should be described as a list of candidates, in which candidates belong to different frames. Each candidate represents a connected region in Fig. 4, or it can be regarded as a single target in a frame. After sorting by time order of frames, the list of candidates is a trajectory.

Suppose we have a trajectory u with length k between time t_m and t_n : $\left[a_{t_{u_1}}^{i_{u_1}}, a_{t_{u_2}}^{i_{u_2}}, \ldots, a_{t_{u_k}}^{i_{u_k}}\right]$, then the best trajectory v can be described as $v = \underset{u}{\operatorname{argmin}}\left(\frac{\sum_{j=1}^{k-2} f\left(a_{t_{u_j}}^{i_{u_j}}, a_{t_{u_{j+1}}}^{i_{u_{j+1}}}, a_{t_{u_{j+2}}}^{i_{u_{j+2}}}\right)}{k-2}\right)$, where f is a manually set function based on moving features that have smaller values when more features are matched. By computing f, the algorithm could determine which trajectory is the moving table tennis ball, while others consist of noise.

If we track back each node in the last layers, and record each Cand-node's candidate, and record no information when meeting a N-node, we can reverse and transform it to a trajectory as the ans of track-back. By the participant of N-node, each possible trajectory can find a node that has the same track-back ans. Note that only one N-node child in each node and its other children are different from each other because they represent different candidate targets in a frame. Each node has different track-back ans. Thus, by time t, all trajectories have one-to-one correspondence to node in the last layer of the Tracking Tree.

In Fig. 6, we select six nodes as an example to show their tracking results and corresponding trajectory. Note that the tracking tree in the figure has been pruned, which will be introduced in the next section.

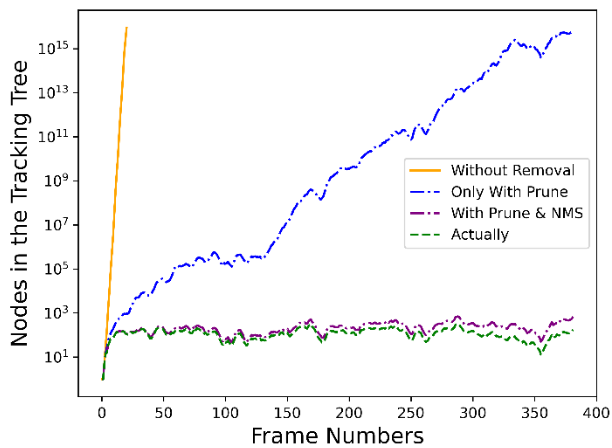
Figure 6. Interpretation of the tracking tree



Fake Target Removal by Tree Pruning

We can compute the length t of the video processed, and get the number of average candidates per frame, n. Thus, the spatio-temporal complexity of the proposed algorithm could be estimated as about $o\left(n^t\right)$. It is too high to be acceptable for processing a video. The reason for this high complexity is too many fake targets causing nodes in the tracking tree. In addition, similar trajectory, e.g. $[a_{t-2},b_{t-1},c_t,d_{t+1},e_{t+2}]$, $[a_{t-2},c_t,d_{t+1},e_{t+2}]$ and $[a_{t-2},b_{t-1},c_t,d_{t+1}]$, would exist at the same time. For a long trajectory with length l, the number of similar trajectory could be 2^l or $o\left(2^l\right)$, which is a huge number. In Fig. 7, the yellow solid line shows the estimated numbers of nodes of a Tracking Tree produced by increased number frames of an example video.

Figure 7. The number of nodes in the tracking tree



To reduce the complexity, we introduce a tree pruning and non-maximum suppression (NMS) strategy to control the scale of the Tracking Tree in a reasonable range and reserve the node for representing the correct trajectory.

To decrease the scale of the Tracking Tree, we define two pruning rules:

(1) If unreasonable pattern exists in the trajectory, the node that it belongs to will be pruned;
(2) Restrict the numbers of grandchildren that belongs to a node.

To execute the two rules, we design a number of indicators and thresholds for judging every trajectory, which are shown in Table 1. When conducting rule 2, we rank its grandchildren by some indicators and reserve nodes that look most like a trajectory of a table tennis ball.

In NMS process, we firstly use indicators and thresholds to get a "good node," then delete other nodes that have the same candidate. For a "good node" in the last layer that has no candidate (N-node), it will be cut if it has the same great-grandfather and grandfather. Also, we will limit the number of nodes with the same candidiates. For example, nodes that have corresponding trajectory [a, b, c], [a, c], [b, c] have same candidates c, while only 1 or 2 of them could be reserved.

Table 1. Typical indicators and its significance

Indicators	Significance	Range & Typical Threshold
two_area_change	$\frac{\left\|a.area - b.area\right\|}{\max\left(a.area - b.area\right)}$	[0,1] & 0.6
two_distance_max	Pixel distance between a.pos and b.pos	[0,frame_size] & 200 (for 720P 60fps)
three_angle_change	$norm\left(a.pos - b.pos\right) - norm\left(b.pos - c.pos\right)$ Where pos is 2d-vector (x,y)	[0,2] & 1.0
three_distance_change	$\frac{\left\|a.pos - b.pos - b.pos - c.pos\right\|}{max(a.pos - b.pos, b.pos - c.pos)}$	[0,1] & 0.8
continue_null_number	Max continuous frame gap between candidates (b.t – a.t or max(b.t-a.t,c.t-b.t))	$\left[1, +\infty\right]$ & 4
three_nodes_reserve	Maximum numbers of grandchildren belongs to node	$\left[1, +\infty\right]$ & 6

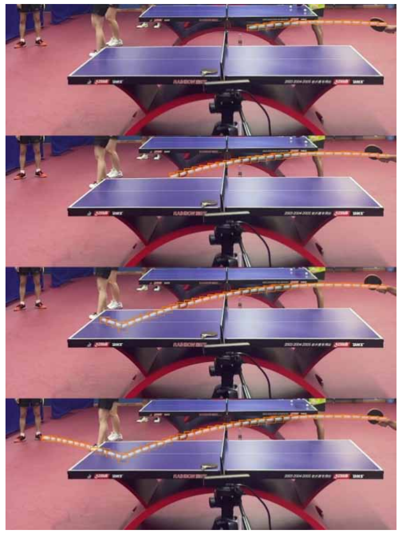
Trajectory Restoration Based on Growing Tracking Tree

In the last section, we could select good nodes by using computed indicators. These good nodes and their trajectory should be regarded as partial trajectory results. However, as the tree grows, the results will also change accordingly. In this period, many similar trajectory and their nodes will constantly exist in the Tracking Tree, serving as different alternative trajectories. By specifying one trajectory as the result trajectory and updating it, we could clarify the output results and control the scale of the Tracking Tree.

Specifically, when we find a “good node,” we firstly compare it with all the updating result trajectories to avoid repeatedly tracking. If we confirm that it is a new trajectory result, we create a new result trajectory, and then set its correspondent node to the found “good node,” or trajectory’s favorite node. As video frame input, the “good node” will change according to the timeline and the trajectory will update synchronously. When no “good node” is found, the state of the result trajectory will change from “updating” to “end.” After that the algorithm can output it.

When selecting “good node” and updating result trajectories, we also introduce a number of indicators and thresholds, which are shown in Table 2. They can help to filter some false trajectory. Some of them can sharply raise the precision while not decreasing the recall. Taking the parameter “ball_min_length” as an example, when the threshold of this parameter is increased, only more stable and longer-lasting table tennis balls in the video stream can be selected, resulting in a significant increase in the algorithm’s accuracy. Another similar example is “ball_delete_null_number,” where increasing the threshold causes the algorithm to wait longer for several frames when no ball is detected. This enhances the algorithm’s ability to resist occlusion and noise, improving recall rate, but also makes the algorithm more susceptible to interference from other targets, leading to a decrease in accuracy.

Figure 8. An updating trajectory in different frames



Most of the time, the algorithm may not be fast enough for real time running. Outputting all of the results after tracking the whole video is preferred. To make the result more orderly and easy to read, it is recommended to save the candidates and their properties from the same frames to independent files, an example is shown in Fig. 9. In this file, we use IDs to distinguish different candidates. The tracking results will be saved in a json file to specify which candidates in which frame consist of the trajectory.

EXPERIMENTAL EVALUATION

To evaluate the effectiveness and performance of the proposed algorithm, we conduct a set of experiments based on open source datasets including OpenTTGames Dataset (Guo et al., 2019) and FMO dataset (Rozumnyi et al., 2017), and a self-build dataset.

Dataset

OpenTTGames dataset is the most popular table tennis dataset that is used by related researchers. It includes a number of videos captured from side and top view with 120 FPS frame rate. We randomly choose a subset from the whole OpenTTGames dataset with 446 balls appearing in 2000 frames. The FMO dataset has three kinds of views, which are side-top view, side-top view close to table, and side view close to table. There are a total of 784 balls appearing in 915 frames. To make the evaluation more comprehensive, we build an additional dataset with more views and playing situations. The detail of the dataset we built is shown in Table 3. All video data in our dataset are basically collected from field recordings, and there is basically no difference from real application scenarios.

The dataset we built has three views. Most of them are on "side view," in which the camera is located at the side of the table, and its optical center usually on the extension line of the table tennis net. The camera's optical axis is nearly parallel to the table tennis net. In this view, the table tennis ball's trajectory is like a shooting star, and has a clear moving feature. It is also the most commonly used view for table tennis analysis. In addition to "side view," we also add some videos in other views and analyze the performance, though it is unsuited to use this algorithm to track because its motion doesn't have a regular pattern.

Table 2. Typical indicators, suppose we have a trajectory [a_1,a_2,a_3,…, a_n] to compute indicators

Indicators	Significance	Range & Typical Threshold
ball_delete_null_number	A trajectory will set to final when its favorite node has param continuous N-node	[1, continue-_null_-number*] & 4
ball_min_length	A trajectory must have at least param candidates $\left(\mathrm{n} > \left(\mathrm{param} - 1\right)\right)$	$\left[2, +\infty\right]$ & 3
delete_similiar_ratio	If more than param ratio in the node's trajectory have same candidates with "good node," the node will be regarded as similar trajectory and removed	$\left[0,1\right]$ &0.4
good_none_empty_ratio	To select(confirm) it is a ball, N-nodes ratio in the trajectory of "good node" must be lower than this param	$\left(0,1\right]$ & 0.5
good_max_average_slope_error	$\frac{1}{n-2} \times \sum_{i=1}^{n-2} norm\left(a_{i+2}.pos - a_{i+1}.pos\right) - norm\left(a_{i+1}.pos - a_i.pos\right)$ Where pos is 2d-vector	[0, three_ angle_change*] & 0.5
good_max_average_distance_error	$\frac{1}{n-2} \times \sum_{i=1}^{n-2} \frac{\left\lvert a_{i+2}.pos - a_{i+1}.pos - a_{i+1}.pos - a_i.pos \right\rvert}{max(a_{i+2}.pos - a_{i+1}.pos, a_{i+1}.pos - a_i.pos)}$	[0, three_ distance_change*] & 0.8

* Please refer to Table 1.

Figure 9. Organizational form of results

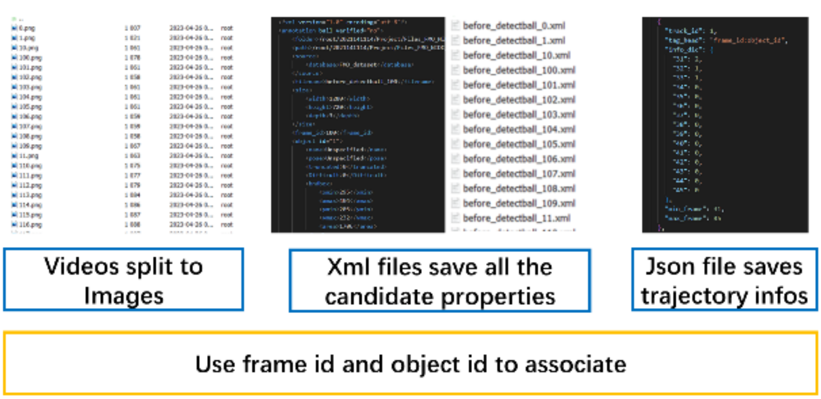


Table 3. Video information and characteristic of our dataset

Subset	View	FPS	Device	Notes
FreePlay30	Side	30	Iphone13	Free play by amateur
Attack60	Side	60	Huawei mate40	Attack action by beginner
DriveTraining	Squint-side	60	Iphone13	Training of professional athletes
Competition	Side	120	LeadSence Dual-camara	Training competition of professional athletes
FreePlay60	Side	120	Iphone13	Free play by professional athletes

The quality and frame rate of the videos varies. We have grayscale frames and color frames. They are from different kinds of devices like mobile phones, professional cameras, and binocular cameras. Our datasets include variations in ball speed, spin, lighting conditions, and background. Furthermore, environmental variables such as lighting conditions, table surface, and background details have also been taken into account. It is worth noting that, in a specific video pattern, the length or frame number of video usually doesn't matter, but the pattern of balls is important. That is to say, if the algorithm works well in the first 10% percent of the video, it will also work well in the rest of the 90%. In fast-moving object datasets, it is different from category and moving patterns, but not frame numbers or data scales. Thus, when testing on a long video, we set maximum number of frames to be processed as 2000, and only analyze this part. Fig. 10 shows example frames of the dataset we built.

Figure 10. Example frames in our dataset

Evaluation Results

As discussed in Section 3, there are a number of details and parameters that can be set in our algorithm, which may help improve the performance. We firstly use a couple of parameters that have low threshold, then set different parameters based on video properties in order to make a simple improvement on the previous results. Table 4 shows the final results. The performance criteria are precision, recall, and F1-score, which are calculated by TP/(TP + FP), TP/(TP + FN), and 2TP/(2TP + FN + FP), respectively. TP, FP, and FN are the number of true positives, false positives and false negatives, respectively. We can see that the proposed algorithm can achieve acceptable precision and recall performance for tracking fast-moving table tennis balls.

Our algorithm takes differential detection as the core, which is pixel-level detection. Therefore, it outperforms deep learning methods represented by CNNs in terms of detecting small targets like table tennis balls. Deep learning and CNN-based methods may have poor sensitivity to small targets

Table 4. Result of the algorithm on various datasets

Dataset	TP	FN	FP	Precision	Recall	F1-Score
FreePlay30	216	34	31	0.874	0.864	0.869
Attack60	552	53	171	0.763	0.912	0.831
DriveTraining	209	10	115	0.645	0.954	0.770
Competition	733	232	7	0.990	0.759	0.859
FreePlay60	650	45	110	0.855	0.935	0.893
openTTGame4	315	5	100	0.759	0.984	0.857
Average				0.81	0.90	0.853

*We mainly focus on the situation when people are playing, thus we set frame boundary in "openTTGame4" (1260 to 1700).

due to pooling and down sampling. We then conducted a detailed analysis of the detected targets by using local image information, such as local gradients and connected domain shapes. Therefore, the detection method has good stability under conditions of rapid table tennis ball movement, partial occlusion, and complex lighting. Its comprehensive performance is superior to most popular object detection algorithms.

To show the excellent performance of the proposed algorithm more clearly, we make a comparsion with the SOTA method, the FMO algorithm proposed in Rozumnyi et al., (2017), based on their dataset. In this dataset, there are three views; "pingpong_top" requires the camera to be set right up the table, which is rarely practically used. In "pingpong_paint," the camera is set close to the desk from the side, which can be used to monitor the training status. And the "pingpong_side" is the common view like our side view. Therefore, we compare the proposed T-FORT algorithm with the FMO method based on the latter two datasets. The results are shown in Table 5. We can see that the proposed algorithm outperforms the FMO method on both precision and recall indicators.

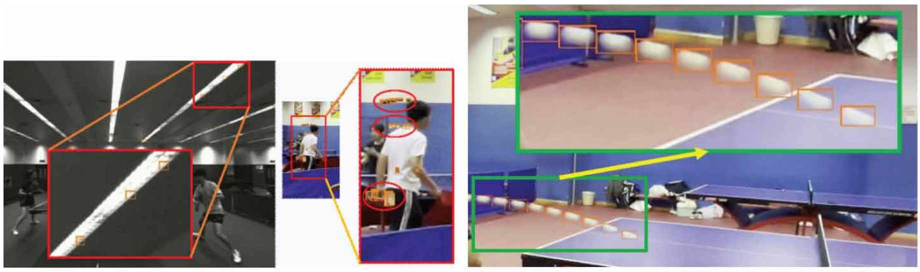
Compared with the state-of-the-art method FMO, our algorithm can continue tracking the ball even when the object detection model fails. The reason for this is that the proposed method can learn moving patterns in different scenarios and combine possible targets from consecutive frames to reconstruct the trajectory with the learned information. Similarly, the occasional noise in a single frame can also be filtered by considering the moving patterns in a set of consecutive frames. That is why our method can achieve better recall and accuracy performance compared with the FMO algorithm.

In addition, based on the analysis of the above results, we found the key point to improve precision performace of the proposed algorithm is filtering fake targets. These fake targets mainly consist of background noises and player movements, some examples are shown in Fig. 11. However, they usually move slowly and are located in a small area. In practice, we can estimate the ideal "pixel speed" for a moving table tennis ball. Combining with some extra information, such as the position of the table and player, it is easy to filter the fake trajectory precicely.

Table 5. Comparasion of the proposed method with FMO

Dataset Name	Number of Frame	Precision(%) FMO/Ours	Recall(%) FMO/Ours
pingpang_paint	120	100/98.6	88.7/95.8
pingpong_side	445	12.0/43.1	7.3/58.9
Average		56.0/70.9	48.0/77.4

Figure 11. Typical false targets. Fake target from background noise (upper left) and people movement (upper right, marked by red ellipses). The difference between table tennis ball (right) and people movement or background noise is clear.



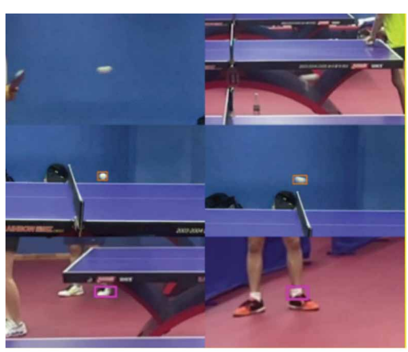
Finally, we compared the differences between our proposed algorithm and the representative deep learning based object detection scheme YOLOv8. The YOLOv8 model we used was trained by labeled images in our dataset. Experimental results in Table 6 demonstrate that the performance of YOLOv8 is acceptable but not as good as the results of our algorithm in Table 4. In most cases, YOLOv8 works well. But it encounters challenges in detecting table tennis balls for the frames with blurry and feature-deficient ball appearance. That is why the F1-Score and overall performance of YOLOv8 are inferior to our proposed algorithm.

Table 6. Result of YOLO-v8 on various datasets (conf=0.1)

Dataset	TP	FN	FP	Precision	Recall	F1-Score
FreePlay30	213	46	33	0.866	0.822	0.844
Attack60	599	376	265	0.693	0.614	0.651
DriveTraining	506	199	71	0.877	0.718	0.789
Competition	324	122	212	0.604	0.726	0.660
FreePlay60	494	113	111	0.817	0.814	0.815
openTTGame4	179	44	16	0.918	0.803	0.856
Average				0.766	0.720	0.742

*Same as Table 4 in the previous text, we mainly focus on the situation when people are playing, thus we set frame boundary in "openTTGame4" (1260 to 1700).

Figure 12. The Results of YOLOv8 detector. Result images of YOLOv8, false negative (top), true positive (middle) and false positive (bottom) examples. Unstable appearance and similar image features can all cause recognition errors.



In summary, our algorithm achieves more advanced results in the problem of table tennis ball tracking by applying tree structures and pruning algorithms to search for possible targets under the tracking framework, outperforming existing target detection algorithms.

CONCLUSION AND FUTURE WORK

In this paper, we have proposed a tracking tree based fast-moving object trajectory tracking algorithm for table tennis. The proposed algorithm is composed of four steps, searching candidate, formulating the tracking tree, fake target removal by tree pruning, and trajectory restore based on tracking tree growing. To evaluate the effectiveness of the proposed algorithm, we collected all existing open-source data and built a more comprehensive dataset. The experimental results demonstrated that the proposed algorithm got a good result with high precision and recall indicators and achieved remarkable performance improvement compared with baseline method. However, the current algorithm can only run offline, and its running speed is about 2-3 FPS without performance optimization. After a large performance optimization, the algorithm is expected to achieve real-time operation. The accuracy of the algorithm still has considerable room for improvement. For future work, we will improve the performance of the proposed algorithm and handle the irregular motion patterns which can occur in real-world scenarios.

AUTHOR NOTE

Correspondence concerning this article should be addressed to Yang Yu, Sports Coaching College, Beijing Sport University, Beijing 100084, China. Email: yuyang@bsu.edu.cn

DATA AVAILABILITY

The data used to support the findings of this study are included within the article.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

FUNDING STATEMENT

This work was partly supported by Humanities and Social Sciences Research Planning Fund, Ministry of Education, China (23YJAZH186), and partly supported by the Fundamental Research Funds for the Central Universities (2112027).

REFERENCES

Adhikari, G., Sanjay, K., Sahani, M. S., & Chauhan, B. K., & Das. (2016). Fast real time object tracking based on normalized cross correlation and importance of thresholding segmentation. in *Proc. ICRTIT*, Chennai, India. doi:10.1109/ICRTIT.2016.7569517

Al-Jilani, B., Rabie, T., & Baziyad, M. (2019). Autonomous motion tracking for dynamic objects using a temporal quad-tree algorithm. *2019 Advances in Science and Engineering Technology International Conferences (ASET).* IEEE.

Al-Sobbahi, R., & Tekli, J. (2022). Comparing deep learning models for low-light natural scene image enhancement and their impact on object detection and classification: Overview, empirical evaluation, and challenges. *Signal Processing Image Communication*, *109*, 116848. doi:10.1016/j.image.2022.116848

Almomani, A., Alauthman, M., Shatnawi, M. T., Alweshah, M., Alrosan, A., Alomoush, W., Gupta, B. B., Gupta, B. B., & Gupta, B. B. (2022). Phishing website detection with semantic features based on machine learning classifiers: A comparative study. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–24. doi:10.4018/IJSWIS.297032

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

Benrazek, A., Kouahla, Z., Farou, B., Seridi, H., Allele, I., & Ferrag, M. A. (2023). Tree-based indexing technique for efficient and real-time label retrieval in the object tracking system. *The Journal of Supercomputing*, *79*(18), 1–38. doi:10.1007/s11227-023-05478-8

Chopra, M., Singh, S. K., Sharma, A., & Gill, S. S. (2022). A comparative study of generative adversarial networks for text-to-image synthesis. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–12. doi:10.4018/IJSSCI.300364

Chu, J., Zhao, X., Song, D., Li, W., Zhang, S., Li, X., & Liu, A. A. (2022). Improved semantic representation learning by multiple clustering for image-based 3D model retrieval. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–20. doi:10.4018/IJSWIS.297033

Dai, P., Weng, R., Choi, W., Zhang, C., He, Z., & Ding, W. (2021). Learning a proposal classifier for multiple object tracking. *Proc. CVPR*, (pp. 2443–2452). IEEE. doi:10.1109/CVPR46437.2021.00247

Dong, X., Shen, J., Wu, D., Guo, K., Jin, X., & Porikli, F. M. (2019, July). Quadruplet network with one-shot learning for fast visual object tracking. *IEEE Transactions on Image Processing*, *28*(7), 3516–3527. doi:10.1109/TIP.2019.2898567 PMID:30762546

Guo, Y., Yang, D., & Chen, Z. (2019). Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by Kalman filter. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(9), 3538–3551. doi:10.1109/JSTARS.2019.2933488

He, J., Huang, Z., Wang, N., & Zhang, Z. (2021). Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. *Proc. CVPR*, (pp. 5299–5309). IEEE. doi:10.1109/CVPR46437.2021.00526

Jelušić, P. B., Poljičak, A., Donevski, D., & Cigula, T. (2022). Low-Frequency data embedding for DFT-Based image steganography. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–11. doi:10.4018/IJSSCI.312558

Kotera, J., Matas, J., & Šroubek, F. (2020, August). Restoration of fast moving objects. *IEEE Transactions on Image Processing*, *29*, 8577–8589. doi:10.1109/TIP.2020.3016490 PMID:32813657

Lee, G. (2015). A fast moving object tracking method by the combination of covariance matrix and Kalman filter algorithm. *Journal of the Korea Institute of Information and Communication Engineering*, *19*(6), 1477–1484. doi:10.6109/jkiice.2015.19.6.1477

Li, D., Deng, L., Gupta, B. 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

Lu, C. (2020, April). Kalman tracking algorithm of ping-pong robot based on fuzzy realtime image. *Journal of Intelligent & Fuzzy Systems*, *38*(4), 3585–3594. doi:10.3233/JIFS-179581

Lu, F., Zhou, B., Zhang, Y., & Zhao, Q. (2018). Real-time 3D scene reconstruction with dynamically moving object using a single depth camera. *The Visual Computer*, *34*(6-8), 753–763. doi:10.1007/s00371-018-1540-8

Nhi, N. T. U., & Le, T. M. (2022). A model of semantic-based image retrieval using C-tree and neighbor graph. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–23. doi:10.4018/IJSWIS.295551

Qian, W., Li, H., & Mu, H. (2022). Circular LBP prior-based enhanced GAN for image style transfer. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(2), 1–15. doi:10.4018/IJSWIS.315601

Rozumnyi, D., Matas, J., Šroubek, F., Novotný, L., & Matas, J. (2017). The world of fast moving objects. *Proc. CVPR, Hawaii, HI, USA,* (pp. 5203–5211). IEEE.

Rozumnyi, D., Matas, J., Šroubek, F., Pollefeys, M., & Oswald, M. R. (2021). Fmodetect: Robust detection of fast moving objects. *Proc. ICCV*, (pp. 3541–3549). IEEE. doi:10.1109/ICCV48922.2021.00352

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

Saab, S. S., & Jaafar, R. H. (2021). A proportional-derivative-double derivative controller for robot manipulators. *International Journal of Control*, *94*(5), 1273–1285. doi:10.1080/00207179.2019.1642518

Sagawa, R., Ota, Y., Yagi, Y., Ryo, F., Asada, N., & Kawasaki, H. (2009). Dense 3D reconstruction method using a single pattern for fast moving object. *Proc. ICCV*, Kyoto, Japan. doi:10.1109/ICCV.2009.5459397

Sayour, M. H., Kozhaya, S. E., & Saab, S. S. (2022). Autonomous robotic manipulation: Real-time, deep-learning approach for grasping of unknown objects. *Journal of Robotics*, *2022*, 2022. doi:10.1155/2022/2585656

Shen, D., & Saab, S. S. (2021). Noisy-output-based direct learning tracking control with Markov nonuniform trial lengths using adaptive gains. *IEEE Transactions on Automatic Control*, *67*(8), 4123–4130. doi:10.1109/TAC.2021.3106860

Sun, C., Wang, Y., & Sheu, M. (2017, September). Fast motion object detection algorithm using complementary depth image on an RGB-D camera. *IEEE Sensors Journal*, *17*(17), 5728–5734. doi:10.1109/JSEN.2017.2723599

Voeikov, R., Falaleev, N. S., & Baikulov, R. (2020). *TTNet: Real-time temporal and spatial video analysis of table tennis*. Proc. CVPR, SeattlE, WA, USA. doi:10.1109/CVPRW50498.2020.00450

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, H., & Zhang, S. (2019, May). Robust and fast object tracking via co-trained adaptive correlation filter. *Optik (Stuttgart)*, *185*, 20–29. doi:10.1016/j.ijleo.2019.02.053

Xiao, J., Stolkin, R., & Leonardis, A. (2015). Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* (pp. 4978-4987). IEEE. doi:10.1109/CVPR.2015.7299132

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

Zheng, Z., Zhou, J., Gan, J., Luo, S., & Gao, W. (2022). Fine-Grained image classification based on cross-attention network. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–12. doi:10.4018/IJSWIS.315747

Zita, A., & Šroubek, F. (2021). Tracking fast moving objects by segmentation network. *Proc. ICPR, Milan, Italy*. doi:10.1109/ICPR48806.2021.9413129

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.

Tianjian Zou is currently working toward the master's degree in Information and Communication Engineering at Beijing University of Posts and Telecommunications (BUPT), and received his B.E. degree in automotive engineering, Tsinghua University in 2021. His research interests include artificial intelligence, big data analysis, and machine learning.

Dazhuang Sun is currently a student working toward the bachelor's degree in e-Commerce Engineering with Law at Beijing University of Posts and Telecommunications (BUPT). His research interests include deep learning, computer vision, and intelligent sports.

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.

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.