TENNIS BALL SERVE SPEED ESTIMATION

An Undergraduate Research Scholars Thesis

by

JASON CHAMPAGNE

Submitted to the Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Scott Miller

May 2025

Major: Electrical Engineering

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Jason Champagne, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Faculty Research Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Tennis Ball Serve Speed Estimation

Jason Champagne
Department of Electrical and Computer Engineering
Texas A&M University

Faculty Research Advisor: Dr. Scott Miller
Department of Electrical and Computer Engineering
Texas A&M University

Serve speed analysis is a valuable tool for tennis players but is often inaccessible due to the high cost of radar-based systems. This research explores a cost-effective alternative, leveraging traditional image processing techniques and a single-camera setup to track tennis ball motion and estimate initial serve speed.

The methodology combines frame differencing and DBSCAN clustering to robustly detect the tennis ball, while trajectory formation and parabolic interpolation accurately identify the ball's ground intersection point. Player tracking is then used to determine the precise serve initiation frame, and court corner detection combined with homography mapping transforms 2D pixel coordinates into real-world 3D spatial data. An exponential correction model is applied to account for air resistance, yielding serve speed estimates that closely align with radar-based measurements.

Experimental validation on 45 video clips from professional matches demonstrates that the system can achieve speed estimates within approximately 4-5% of the ground truth. The study also discusses limitations such as sensitivity to camera movement and minimal trajectory

angle changes and outlines future improvements including advanced detection and integration of machine learning methods. These enhancements aim to improve tracking robustness and adaptability to diverse recording conditions, ultimately paving the way for accessible and reliable serve speed analysis in real-world tennis training environments.

ACKNOWLEDGEMENTS

Contributors

I would like to thank my faculty advisor, Dr. Scott Miller for his guidance and support throughout the course of this research. His insights and expertise have greatly shaped the direction and success of this project.

Funding Sources

I did not receive any external funding for this research.

1. INTRODUCTION

In tennis, accurately measuring serve speed is essential for player development, performance analysis, and strategic planning. Traditional methods, such as radar guns, or sophisticated multi-camera setups, are either expensive or inaccessible, creating a need for more affordable alternatives. Recent advancements in computer vision and image processing offer opportunities for a cost-effective solution using only a phone camera. This project focuses on refining the accuracy of initial ball speed estimation through post-processed video analysis, ultimately aiming to develop a system that tennis players and coaches can easily implement without specialized equipment.

1.1 Motivation

Tracking tennis ball serve speed plays a critical role in improving player performance. By understanding serve speed, players can refine their techniques, allowing them the tools to improve their skill. However, existing solutions such as radar guns or advanced camera setups remain impractical for many users due to cost and complexity. This project addresses this gap by leveraging traditional image processing techniques in a single-camera framework to offer a low-cost yet accurate alternative for serve speed estimation.

1.1.1 Challenges in Current Methods

Despite advancements in tennis analytics, accurately estimating serve speed remains challenging. Systems like SwingVision AI provide comprehensive match insights but typically do not prioritize precise ball speed tracking. Research systems using multiple cameras or motion sensors, such as Renò et al. [1] and Zhao et al. [2], have demonstrated impressive accuracy; however, these systems require expensive hardware setups. Even smartphone-based methods, such as those proposed by Fazio et al. [3], encounter limitations in positional accuracy, particularly when dealing with the high-speed dynamics of a tennis serve.

1.1.2 Scope of This Project

This project narrows its focus to the initial serve speed estimation using traditional image processing techniques. By concentrating on video data captured with a single camera, the methodology aims to provide a repeatable and explainable framework that accurately estimates serve speed without specialized equipment. Such a system could help a broad range of users, particularly at the amateur and college levels, to access advanced performance analytics.

1.2 Literature Review

This project builds on existing research in tennis ball tracking and vision-based techniques. Renò et al. [1] demonstrated that a four-camera setup could track a tennis ball during rallies within a margin of 0.1 meters 93.6% of the time, highlighting the benefits of enhanced spatial awareness with multi-camera configurations. In contrast, Yang et al. [4] implemented a more streamlined system using only two cameras combined with neural networks to predict ball trajectories for robotic gameplay, emphasizing that high levels of tracking accuracy are attainable with fewer hardware resources, allowing neural networks to predict the tennis ball location in real time.

Complementing these multi-camera approaches, Li, X. & Huang [5] introduced a marker-based methodology that uses color-coded markers on key joints alongside a single high-speed video camera to precisely track the motion of these joints, providing an in-depth biomechanical analysis of tennis serve mechanics. Their approach integrates traditional image processing techniques, such as background subtraction, optical flow, and inter-frame difference methods, to extract trajectories from the serve motion, effectively reducing noise and reliably tracking the players.

Another successful demonstration of single-camera systems, Huang et al. [6] proposed TrackNet, a marker-less, deep learning-based method for tennis ball tracking that directly supports serve speed estimation. By employing a lightweight convolutional neural network trained on consecutive video frames to generate detailed detection heatmaps, TrackNet achieves a precision of 99.7%, and a recall of 97.3%, where precision is defined as TP/(TP+FP) and recall as TP/(TP+FN), with TP, FP, and FN representing true positives, false positives, and false negatives, respectively. These results mean that there are very few missed tennis ball detections (false negatives) and even fewer incorrect detections (false positives). By eliminating the need for specialized markers and high-speed cameras, TrackNet provides a cost-effective solution that complements biomechanical insights from marker-based methods.

Another key advancement in single-camera approaches is provided by Kelly et al. [7], who developed an automated system for measuring tennis serve speed using a single high-speed camera. By validating their system against radar measurements, they achieved a mean error of approximately 4.5%, highlighting that framerate capture and precise camera calibration are crucial for accurate speed estimation.

Similarly, Chen et al. [8] showed that incorporating physics-based constraints by applying projectile motion equations can significantly refine trajectory reconstruction from single-camera video. Using a single camera setup, they extracted 2D ball trajectories and applied motion constraints to infer 3D flight paths. Their system correctly reconstructed over 90% of shooting trajectories, achieving a mean three-dimensional positional error of 0.14 meters across over 1,200 tracked ball frames. This approach demonstrates that a single camera, physics-informed approach can yield high accuracy in reconstructing ball motion.

In summary, these studies demonstrate the potential both multi-camera and single-camera systems for tennis ball tracking and serve analysis. While Reno et al. [1] highlight the enhanced spatial awareness of multi-camera setups, Yang et al. [4] and TrackNet [6] demonstrate that high levels of tracking accuracy can be achieved with more resource-efficient, neural network-based approaches. Similarly, Li and Huang's [5] marker-based methodology provide a detailed framework for capturing detailed biomechanical insights from a single high-speed camera. Together, these works show that advanced image processing techniques can effectively reduce background noise, reliably track motion, and extract precise spatial-temporal data. Building on these findings, the current project seeks to integrate these diverse techniques into a cost-effective, single-camera system aimed specifically at enhancing the accuracy of serve speed estimation.

2. METHODOLOGY

The goal of this research is to develop an accurate method for tracking tennis ball serve speed using only traditional image processing techniques and a standard phone camera. The project focuses on tracking the tennis ball and court corners in 2D space and calculating the ball's speed. This section outlines the key methods used for tennis ball tracking, court corner detection, and the 3D space calculations to determine the tennis ball serve speed. Figure 1 details a flowchart of the methodology that will be used.

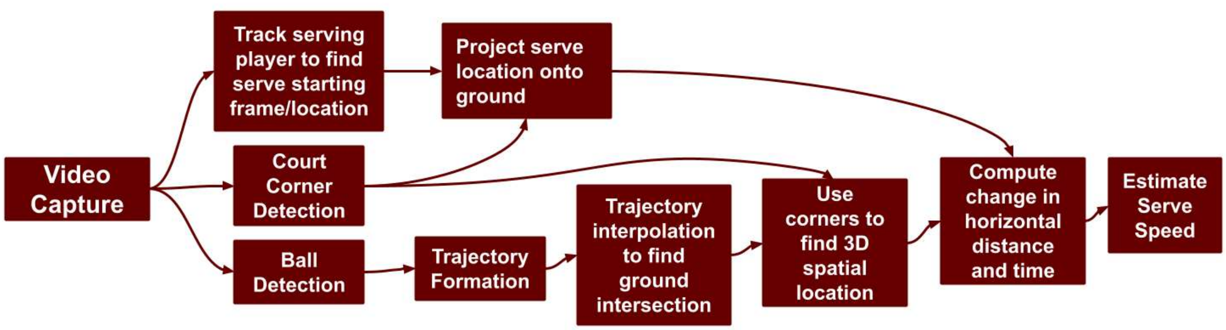

Figure 1: Flow Chart explaining methodology.

2.1 Video Capture

2.1.1 Overview and Research Goal

The long-term goal of this research is to develop a system that can accurately compute the initial speed of a tennis ball using only a standard phone camera. Ideally, this system would not require specialized equipment, such as radar-based tracking, making it accessible for players and coaches seeking to improve their skills. The videos chosen for this study serve as an initial step toward this goal, providing high-quality data to develop and refine the tracking methodology. While these videos come from professional matches with clear footage, the ultimate objective is to create a system that generalizes well to lower-quality recordings from phone cameras.

2.1.2 Selection of Tennis Match Videos

Professional tennis match recordings were obtained from publicly available sources on the internet. Videos were selected based on the availability of ground truth radar measurements, allowing for direct comparison with the estimated serve speeds. Videos with an unobstructed view of the court, player, and ball were chosen to ensure accurate tracking. High-definition videos with a stable, elevated camera position were preferred, as they provide clearer visual references for ball detection and trajectory estimation.

Since the focus of this study focuses on calculating serve speed, each video was trimmed to include only the first few seconds of a rally. This helped to reduce computational load while ensuring all necessary data was retained. Trimming was done manually by identifying the moment just before the ball toss and extending a couple of seconds after the first return. The trimming process used a program called "Lossless Cut," to ensure that the original frame rate and video quality remained unchanged, preserving the necessary details for accurate serve speed estimation. No additional image processing or filtering was applied at this stage to preserve the raw video data.

2.2 Tennis Ball Detection

Once the video has been selected and trimmed, it is processed frame by frame to detect the tennis ball's movement throughout the serve. The algorithm begins by reading the video as an input and iterating through each frame in sequence. For each frame, a function extracts the ball's position while filtering out irrelevant motion, such as player movement and background noise using image processing techniques. This function processes each frame independently but also incorporates information from the previous and the next frame for effective frame differencing, ensuring that only the moving objects remain. The following section details the steps involved in the initial process of isolating the tennis ball.

2.2.1 Frame Differencing

To detect the ball, the algorithm uses frame differencing which compares the previous and next frame to the current frame to extract regions of motion. Since the ball is one of the few objects in constant motion, this approach effectively isolates it from static background elements. Frame differencing works by taking the absolute difference between the previous frame and the current frame, and the current frame and the next frame, resulting in two motion masks. These masks are then converted to binary, retaining only pixels with an intensity difference above a certain threshold. This helps to filter out minor differences, likely caused by random noise. These masks are first combined using a logical AND operation to retain only consistent motion across the three frames. The remaining motion regions are then converted back into a grayscale mask by averaging the intensity values between the two masks, to enhance motion detail in the moving regions. This results in a grayscale mask that retains only major changes in intensity, effectively recording the moving elements in the image.

2.2.2 Thresholding and Smoothing

Once the motion mask is generated, a Gaussian filter is then used to smooth the mask, merging fragmented motion regions and further reducing noise. This step ensures the detected motion is more continuous and less affected by minor fluctuations. A second thresholding pass is then applied for further detection refinement. This threshold is adjusted as a percentage of the maximum intensity of the mask. This step further reduces the influence of small, fragmented noise artifacts while retaining the most significant regions of motion.

2.2.3 *Morphological Processing*

Since the tennis ball is a small object relative to the frame size, it is susceptible to being lost during thresholding. To prevent this, a morphological dilation operation is applied, expanding the detected motion regions to ensure the ball remains visible in the processed mask. A disk-shaped structuring element is used to enlarge smaller objects, including the ball. While this may expand some noise, later steps will refine the detection.

After dilation, Canny edge detection [9] is performed on the original grayscale input frame, extracting the edges of the original image. The detected edges are combined with the motion mask, highlighting the ball's contour while filtering out less structured noise. Another dilation step follows to further enlarge the ball. Finally, another thresholding operation is performed, resulting in a binary mask containing the ball, players, and a bit of remaining noise. These refinements set up the foundation for the next step, where the ball's location is identified using clustering techniques.

2.2.4 *Clustering with DBSCAN*

At this stage, multiple moving objects may still be present in the processed mask, including noise clusters. To reliably identify the tennis ball, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [10] algorithm is applied to group isolated pixels into clusters. The algorithm scans the processed mask and extracts all non-zero pixels, treating them as candidate points.

A clustering radius (eps = 10 pixels) is set to define how close the points must be to belong to the same cluster, and a minimum cluster size (minPts = 10) is used to eliminate small noise regions. In this implementation, a cluster is defined as a group of at least 10 pixels (minPts

= 10), each having at least one neighboring point within a 10-pixel radius (eps = 10). Clusters that do not meet these conditions are discarded as noise.

For each valid cluster, the algorithm computes:

- Mean (x,y) position: The estimated center of the object.
- Cluster size: Number of points in the cluster.
- Topmost point: The highest (x, y) coordinate in the cluster, useful for future analysis of the serve's starting frame.

This information is stored and analyzed for each frame, forming the foundation for both tennis ball trajectory formation and the identification of the player's peak jump. The player's peak jump will correspond to the starting serve frame, while the tennis ball trajectory will be used for detecting the ground intersection frame.

2.3 Trajectory Formation and Filtering

Once the tennis ball has been detected and tracked in each frame, the next step is to link these detections across consecutive frames to form a continuous trajectory. This trajectory represents the ball's motion throughout the video clip, ensuring each detection corresponds to the same physical object over time. Reliable trajectory formation is essential for future analysis, including parabolic interpolation for impact estimation.

To achieve this, detected pixel clusters are tracked frame by frame based on spatial proximity, ensuring consistent identification of the same object throughout the sequence.

2.3.1 Converting Frames into Clusters

Before trajectories can be formed, detected clusters are identified and stored in a structured format. Clusters detected in individual frames are associated across consecutive frames to form coherent trajectories. This process matches clusters based on spatial proximity and continuity of motion, creating a framework for reliable analysis.

2.3.1.1 Initialization

Each detected cluster is stored in a global list, tracked_clusters, with attributes including:

- Unique Identifier (ID)
- Start and end frame indices
- Frame-by-frame trajectory data, including position, size, and motion metrics (e.g., speed, angle).

2.3.1.2 Frame-by-Frame Cluster Matching

Clusters are matched across consecutive frames based on spatial proximity, ensuring continuity in tracking. A distance matrix is computed to measure the spatial proximity between clusters in the current frame and active clusters from the previous frame, provided the distance is within a predefined threshold (D_max = 80 pixels). Unmatched clusters are either assigned new IDs (if they appear for the first time) or marked as ended (if they disappear from the frame).

To extract current clusters, their x, y coordinates and sizes are recorded for each frame. In the first frame, all detected clusters are treated as new, assigned unique IDs, and added to tracked_clusters. These clusters are also marked as active to facilitate future matching.

For subsequent frames, each active cluster is compared to the clusters in the current frame using the distance matrix. Each active cluster is paired with the nearest unmatched cluster from the current frame, ensuring consistency in tracking.

Unmatched clusters in the current frame are treated as new, assigned unique IDs, and added to tracked_clusters. Similarly, active clusters that remain unmatched are marked as ended, and their final frame is recorded. This ensures that all clusters are accurately initialized, tracked, and finalized as their visibility changes over time.

2.3.1.3 Deriving Motion Metrics

To analyze cluster trajectories, the following metrics are calculated:

- Speed: The Euclidean distance traveled per frame, representing the ball's instantaneous velocity
- Angle: Direction of motion relative to the x-axis
- Change in Angle: The smoothness of trajectory, calculated as the difference in angles between consecutive segments.
- Average Speed: The average speed for the entire trajectory, excluding missing or noisy data.

2.3.2 Filtering Clusters

To refine the dataset, clusters are filtered based on lifespan, speed, and motion smoothness, isolating those most likely to correspond to the tennis ball. This step removes noise and irrelevant artifacts, ensuring reliable trajectory data for subsequent analysis.

Filtering is essential for removing irrelevant data while preserving clusters that exhibit the expected motion characteristics of a tennis ball. By focusing on clusters that demonstrate smooth, sustained motion at appropriate speeds, this stage enhances the reliability of trajectory modeling and ensures clean input for subsequent steps, including parabolic trajectory estimation.

2.3.2.1 Filtering Criteria

Each cluster in the tracked_clusters dataset is evaluated against three criteria:

- Lifespan: Clusters must persist for a minimum number of frames (min_frames) to be considered meaningful motion, rather than noise.

- Average Speed: Clusters must maintain an average speed above a defined threshold to eliminate slow-moving objects or artifacts
- Smoothness: Clusters with erratic motion, identified by high values of absolute angle change, are excluded. Excessive angular changes indicate non-linear motion, which should not be the case when identifying tennis serves.

2.3.2.2 Integration and Parameter Tuning

Parameter tuning is an integral part of the methodology to ensure the filtering criteria align with the expected motion of a tennis ball. The parameter max_angle_change was included to account for trajectory smoothness, with flexibility to adjust its threshold based on context. For initial testing, the criterion was temporarily relaxed to explore the relative impact of lifespan and speed on the filtering process. This iterative tuning helps ensure robust identification of relevant clusters without prematurely excluding meaningful data.

2.4 Trajectory Interpolation and Ground Intersection Calculation

Once the ball trajectory has been formed and refined, the next step is to interpolate the trajectory and compute the ball's intersection with the ground. This process is critical for determining when and where the ball lands, providing the data required for serve speed estimation.

Before performing parabolic interpolation, additional filtering is applied to remove short-lived, erratic, or non-smooth trajectories that could introduce inaccuracies. This step helps find locate the correct trajectories used for computing the ball's landing position.

2.4.1 Filtering Clusters for Stable Trajectories

To ensure the ball's trajectory is consistent and fast, the system first applies two filtering functions: Filtering short, slow-moving clusters, and filtering erratic motion.

2.4.1.1 Filtering Short and Slow-Moving Clusters

The filter_short_clusters function identifies and discards clusters that fail to meet the following criteria:

- Minimum Lifespan (min_frames = 6 frames): Clusters must persist across at least six frames to ensure their motion is deemed valid and not noise.
- Minimum Speed (min_average_speed = 10 pixels/frame): Clusters moving below this average speed are unlikely to represent the tennis ball and are therefore excluded.

By applying these criteria, only clusters exhibiting consistent and rapid motion are passed to the subsequent stages of processing.

2.4.1.2 Filtering Erratic Motion Using Angle Constraints (filter_angles)

Once the short lived and slow clusters have been removed, the filter_angles function applies additional constraints to ensure smooth trajectory motion.

- Maximum Angle Change (max_angle_change = 30 degrees): Any cluster exhibiting sharp, sudden directional changes exceeding this threshold is split into separate clusters before and after the abrupt turn.
- Minimum Consecutive Smooth Frames: Clusters must again maintain consistent motion across a predefined number of consecutive frames to remain valid.

These filters work together to isolate smooth, continuous trajectories to enhance downstream analysis.

2.4.2 Parabolic Interpolation and Intersection Computation

After filtering, the remaining valid trajectories are used to perform parabolic interpolation to identify the ball's ground intersection point. This analysis assumes that the first and second valid trajectories correspond to the ball's motion before and after ground contact, respectively.

The parabolic interpolation function processes these trajectories by fitting quadratic equations to the x and y components of each trajectory. This results in two parametric equations that describe the ball's motion in 2D space. The algorithm then calculates the intersection point by minimizing the sum of squared differences between the two parabolic equations, effectively pinpointing the ball's most probable landing position and the corresponding video frame.

One major advantage of this interpolation approach is its precision. It computes the exact intersection between parabolic trajectories, enabling the algorithm to compute a very precise frame estimate. Additionally, the method returns regression coefficients that indicate how well the data fits the quadratic model, along with the total sum of squared errors. These outputs provide useful confidence metrics that are useful in assessing the accuracy of the interpolation.

2.5 Tracking the Serving Player

Identifying the serving player is an important step in the trajectory analysis process as it helps determine the frame corresponding to the serve's initiation. While the tennis ball exhibits rapid, dynamic movement throughout the play, the serving player maintains a steadier pace, making their movement relatively easier to track. By isolating the serving player's trajectory, the algorithm can accurately find the frame associated with their peak jump. To achieve this, the filter_top_player_cluster function is used to extract the most likely player cluster from the list of tracked clusters. This function applies several filtering criteria to ensure that the selected cluster corresponds to the player rather than noise or other moving objects.

2.5.1 Filtering the Player Cluster

To effectively filter only the player, specific metrics related to the player are inputted into the function. For instance, the player should be visible for a minimum number of frames (min_frames) to ensure the detected cluster represents sustained motion rather than transient noise. This threshold is currently set to 60 frames, corresponding to 2 full seconds of continuous visibility. Additionally, the serving player cluster must have an average speed less than 10 pixels per frame. A height constraint is also enforced, meaning the player's average height must exceed a certain threshold. Once all filters are applied, the first cluster that meets all specified criteria is selected and returned as the serving player's trajectory.

2.5.2 Determining the Serve Initiation Frame

Once the serving player cluster has been identified, the next step is to determine the exact frame corresponding to the peak of their jump, which marks the initiation of the serve. To achieve this, the get_top_y_values function processes the trajectory of the filtered player cluster and extracts the frame in which the player reaches their highest position.

This function takes the trajectory of the identified serving player as input and applies the following operations:

2.5.2.1 Identifying Peak Jump Frame

The function retrieves all Top_Y values from the player's trajectory. Since Y-coordinates increase downward in this coordinate system, the smallest Top_Y value represents the player's highest jump position. This function searches for the minimum Top_Y value, corresponding to the highest vertical position reached by the player. The index of this minimum value is used to extract the corresponding frame number, X position, and Y position. The function outputs a structured result containing:

- Frame: The frame in which the player reached peak height.
- Y: The corresponding Top_Y value (player's highest detected position).
- X: The Top_X value, representing the average horizontal position of the player at the peak jump.

2.5.2.2 Role of Player Tracking in Serve Analysis

Determining the serve initiation frame is essential in computing serve speed accurately. The output will be used for determining the net time and spatial difference between the start and end of the serve, leading to more precise serve speed estimations and motion analysis.

2.6 Court Corner Detection

After the start and end pixel coordinates and times are estimated, the tennis court corners are calculated in the 2D video frame. This step is important for the subsequent steps that will pair this data with the real-world 3D spatial coordinates of a tennis court to convert 2D projections into three dimensions for ultimately calculating the horizontal displacement.

2.6.1 Preprocessing and Edge Detection

Before detecting court corners, the input video frame is converted to grayscale, and contrast stretching is applied to enhance the tennis court lines while suppressing noise. Next, Canny edge detection is used to detect the strong edges of the image, followed by a gaussian blur and then a thresholding function to retain the significant edge regions.

2.6.2 Angle-Based Morphological Filtering

Since tennis court lines are predominantly straight and oriented at specific angles, an angle-based opening technique is applied to extract these lines. Essentially a rotating line structuring element is used to retain long lines at each given angle. The processed results are combined across all angles to form a refined binary mask of the court, retaining lines above a certain length. This approach, although more computationally intensive, yielded much better results than a standard line detection algorithm such as the Hough Transform.

2.6.3 Court Intersection Analysis

After detecting the court lines, DBSCAN clustering is applied to filter out irrelevant structures. This ensures that the tennis court is retained while smaller, isolated line structures and noise get filtered out. A final morphological operation is performed, specifically an opening, which is a two-step process where an erosion removes small artifacts, and a subsequent dilation restores the shape of larger features using a defined structuring element. This process effectively cleans up the binary image by eliminating minor noise while preserving the court's main structure. Finally, the outermost corners of the remaining line segments are determined by taking the minimum and maximum sum and difference between the x and y position lists. This ensures the correct corners will be found and tracked as long as the camera is generally behind the returning player. These four corners are stored as a structure and used to project the serving and intersection pixel positions into spatial coordinates.

2.7 3D Spatial Position Estimation

To accurately analyze the ball's position relative to the court, the 2D pixel coordinates from the video frame must be transformed into real-world 3D spatial coordinates. This transformation allows for accurate distance estimation, ensuring the computed serve speed and ball movement are properly aligned with the actual court dimensions. The compute_distance_estimate function calculates the real-world horizontal distance between the ball's landing position and the player's peak jump position by:

- Computing the homography matrix (H), which maps pixel coordinates to real-world court dimensions.
- Transforming the ball's intersection point and player's peak position into real-world coordinates.

- Estimating the Euclidean distance between these points, assuming the court is a flat plane.

2.7.1 Computing Homography Matrix

The homography matrix relates pixel coordinates from the video frame into known real-world court dimensions. This function takes in the detected court corners in pixel coordinates and the corresponding real world coordinates, measured in meters, to compute a matrix detailing the camera position and angle. This transformation ensures that the points in the 2D image can be mapped to the correct positions on the real-world court.

2.7.2 Mapping Key Points to Real-World Coordinates and Computing Distance

After the homography matrix is computed, two key positions are transformed into real-world coordinates: The ball's landing position and the player's peak jump position (projected onto the ground).

In order to find the player's peak jump position, an interpolation function is created to find the corresponding y-position that maps the ball coordinates onto the ground. This is done because the final algorithm uses the horizontal displacement, so the point must be mapped onto the ground. After this is performed, the resulting coordinates are mapped into 3D spatial coordinates using the homography transformation. The z-coordinate is set to 0 meters since we are projecting onto the 2D plane of the court. Finally, Euclidean distance between these two coordinates is computed to find the 3D horizontal distance between the start and end of the serve.

2.8 Speed Calculation

The final step in the analysis pipeline is computing the initial serve speed based on the distance traveled by the ball and the time taken between the serve start and the ball's impact point. The speed is computed using an exponential correction model that accounts for air resistance, as shown in Equation 1.

$$V_0 = \frac{e^{kC_d x} - 1}{kC_d t} \tag{1}$$

Equation 1: Initial serve speed equation.

In this equation, V0 is the initial serve speed, Cd is the drag coefficient, k is a constant including mass, cross sectional area of the tennis ball, and density of the air, x is the change in horizontal distance in meters, and t is the time in seconds [11]. This exponential correction model, derived from the principles outlined by Cross in chapter 42, is crucial because it acknowledges the significant impact of air resistance on the ball's velocity during its flight. As Cross explains, even at moderate serve speeds, the drag force caused by air resistance can be a substantial fraction of the gravitational force acting on the ball. Without accounting for this deceleration, the calculated initial serve speed would be significantly underestimated.

3. RESULTS

3.1 Data Collection

3.1.1 Tennis match selection process

The dataset for this study consisted of 45 video clips extracted from three professional tennis matches held during the US Open in 2018 and 2023. This provided variation in both serve characteristics and visual conditions. The specific matches included:

- Daniil Medvedev vs. Alex Minaur (2023): Serves ranged in speed from 83 MPH to 128 MPH (9 clips).
- Serena Williams vs. Kaia Kanepi (2018): Serves ranged in speed from 80 MPH to 109 MPH (20 clips).
- Borna Gojo vs. Jiri Vesely (2023): Serves ranged in speed from 87 MPH to 130 MPH (16 clips).

Notably, the Serena Williams vs. Kaia Kanepi match was from 2018, while the other two were from 2023. This temporal difference introduces visual variations that contribute to the overall diversity of the dataset. The selection of these matches was intended to capture a broad sample of serve speeds, camera angles, and serve trajectories. Based on these considerations, specific criteria were established for selecting the individual clips used in this study.

3.1.2 Clip Selection Criteria

The selection of video clips for this study was guided by several criteria to ensure data quality and consistency:

- Serving Player Position: Clips were chosen only when the serving player appeared at the top of the screen. This allowed the algorithm to more reliably track the serving player's motion and the accurate detection of the serve initiation frame and location.
- Pronounced Angular Change: Only clips with a distinct directional change between the ball's pre- and post-bounce trajectories were included based on visual inspection. This clear directional change was essential, as the algorithm relies on identifying distinct parabolic trajectories before and after the bounce to accurately interpolate the ground intersection point.
- Camera Stability: Clips were selected from recordings with a stationary camera. This requirement was implemented because the motion detection process is very sensitive to camera movement, which introduces noise and reduces accuracy. Clips tested with camera movement resulted in the frame differencing algorithm picking up many false positives, resulting in slower performance, and generally incorrect trajectory selection.
- Frame Rate: All selected videos used were sampled at 30 frames per second (fps). This frame rate was chosen as a balance between accurate trajectory estimation and computational cost. Alternative frame rates (25 and 60 fps) were tested but typically resulted in increased false positives or false negatives, making 30 fps the optimal choice for this analysis.

3.1.3 Definition of Success and Overall Success Rate

To evaluate the generalizability of the algorithm, all parameters specified in the methodology were kept constant across all video clips. A case was defined as "successful" if it met all the following criteria:

- Trajectory Tracking: The tennis ball must be detected for at least six consecutive frames (min_frames = 6) to form a coherent trajectory. Additionally, the trajectory must be properly trimmed based on angular change, ensuring that the segments before and after ground impact are identified as the most probable paths (see Figure 2 for an example).
- Ground Intersection Point Estimation: The estimated landing point must visually correspond with the actual landing point and frame. Cases where the estimated intersection deviated significantly from the true landing spot were classified as failures.
- Serve Initiation Detection: The system must accurately capture the serve's starting time and location, which are essential for calculating the Euclidean distance and temporal difference required for the final serve velocity. Cases that failed to capture the serving player within a couple of frames of the start of the serve—or mistakenly tracked a different object—were discarded.

Figure 2: Trajectory plotted before and after hitting the ground.

Below, Figure 3 shows an example of a typical "pass" case. In this example, the top point was correctly selected, the court corners were accurately tracked, the ground intersection was located in the correct area, and both the initial serve time and the intersection time corresponded closely to the actual values.

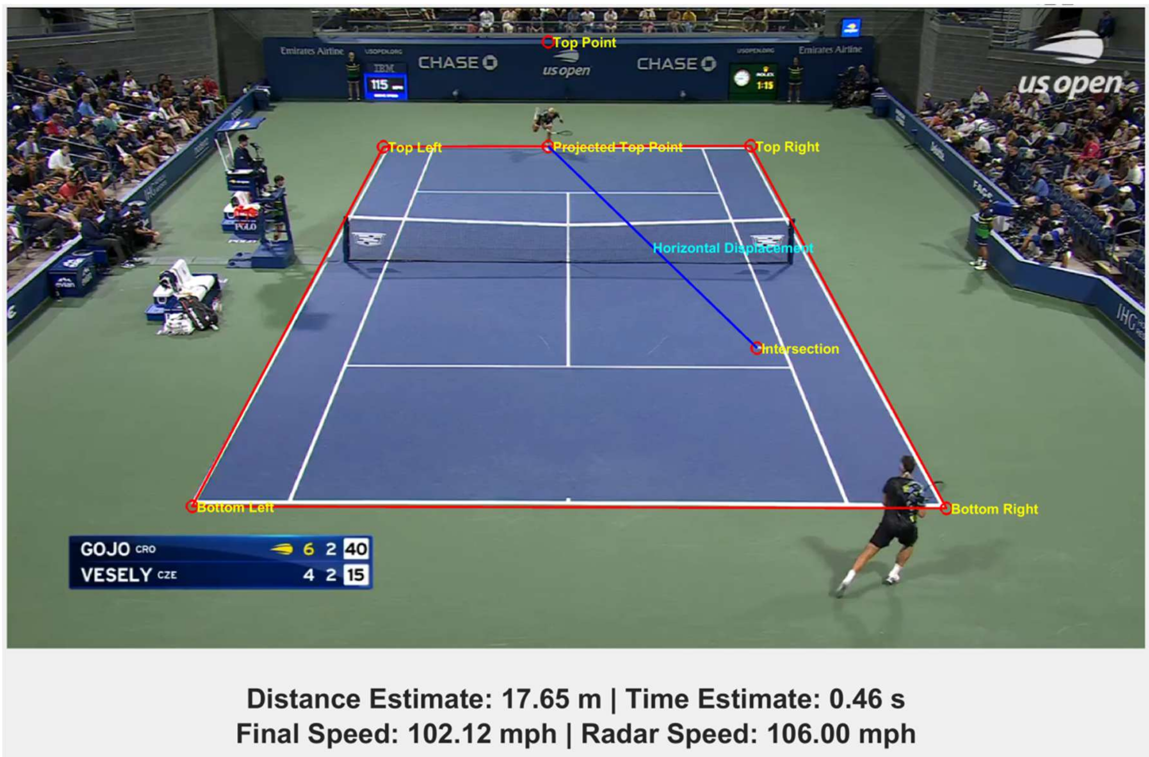

Figure 3: MATLAB final plot output for rally clip.

3.2 Summary of Results

3.2.1 Definition of Success and Overall Success Rate

In total, of the 45 clips tested, 28 of them passed all the previous criteria, resulting in an overall success rate of 62%. Among these successful cases, 18 (approximately 64%) produced serve speed estimates that fell within 5% of the radar-based ground truth. These findings indicate that when the algorithm functions as intended, it can deliver high quantitative accuracy in serve speed estimation. In addition, the results underscore the effectiveness of the exponential correction model (Equation 1) that accounts for air resistance in accurately estimating serve velocity.

In instances where the algorithm failed to identify the correct ground truth intersection point, it typically identified the correct pre-and post- bounce trajectories but did not select them as the optimal paths for interpolation. This error in trajectory selection ultimately led to incorrect coordinate and timing estimations.

3.2.2 Quantitative Analysis and Scatter Plot Evaluation

Figure 4 shows a scatter plot comparing the algorithm's serve speed estimates against the radar ground truth for the 28 successful cases. The analysis yielded a Root Mean Square Error (RMSE) of 4.91 MPH, indicating that, on average, the algorithm's estimates deviate by about 4.91 MPH from the radar measurements. The bias was calculated to be -0.56 MPH, a small value that suggests a slight tendency to underestimate serve speed. Overall, these quantitative metrics support the reliability of the proposed approach and suggest that further refinement may help reduce both the RMSE and any underlying bias.

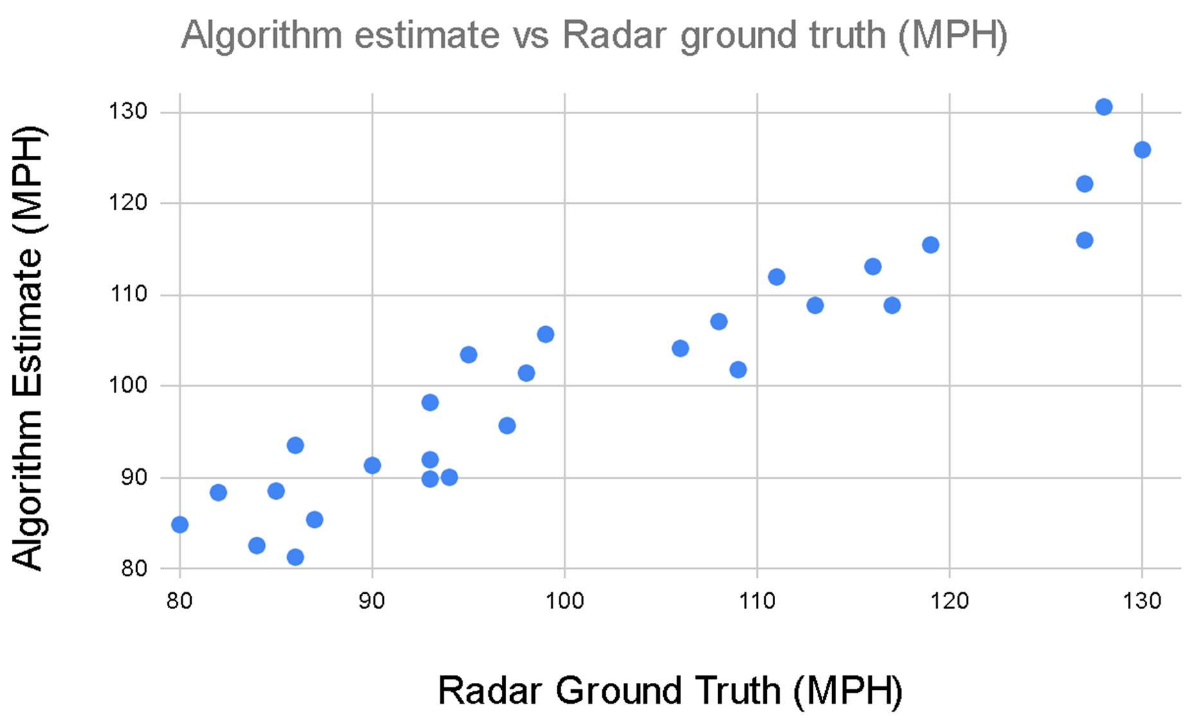

Figure 4: Scatter plot of the Radar Ground Truth vs Algorithm Estimate (MPH) for passing cases.

3.3 Limitations, and Observations

The experimental results demonstrate that a cost-effective, single-camera solution can deliver reasonably precise serve speed estimates. However, several factors contribute to the overall error observed in our system.

A primary source of error is the selection of the initial serve frame. Because the computation of serve speed depends critically on accurately determining the serve initiation, even a one- or two-frame misalignment can lead to significant errors in the calculated velocity. This sensitivity suggests that employing more robust or precise methods for serve initiation detection, such as incorporating audio cues from the tennis racket impact or leveraging machine-learning-based estimation, may help reduce these timing errors.

In addition to timing inaccuracies, the algorithm exhibits limitations in trajectory segmentation. Specifically, when the angular change between the ball's pre- and post-bounce trajectories is minimal, the system sometimes fails to distinguish the two segments, leading to incorrect interpolation of the ground intersection point. If either segment is not properly detected, the final serve speed estimate cannot be trusted.

The system is also sensitive to adverse recording conditions. Unintended camera movement can introduce background noise, causing false detections and the unintentional filtering of critical trajectory data.

Despite these limitations, the overall success rate and quantitative performance indicate that the proposed approach is a promising foundation for further development. Future work will focus on refining serve frame detection, enhancing segmentation algorithms—potentially through AI-based object detection—and improving filtering techniques to better accommodate less-than-ideal recording conditions. These improvements could help further reduce the RMSE and enhance the algorithm’s generalizability, paving the way for a more robust and accurate serve speed estimation tool.

4. CONCLUSION

4.1 Summary of Contributions and System Performance

This research successfully demonstrates the feasibility of estimating tennis serve-speed with reasonable accuracy using a single video camera and classical image processing techniques, offering a cost-effective alternative to more complex and expensive systems.

The methodology involves isolating the serve motion within video clips, tracking the ball's trajectory before and after the bounce, identifying the serve initiation frame using player tracking, and then mapping key points onto a 2D plane. This information is then used to make 3D projections, and the 3D horizontal distance is calculated. Finally, time differences and horizontal distance information is applied to an exponential correction model that is then used to calculate the final serve velocity. Based on this approach, we were able to reliably estimate serve speeds within a few percent of radar-based measurements without the need for expensive hardware or complex multi-camera setups.

The system was evaluated on a diverse set of video clips that, in successful cases, consistently tracked both the tennis ball and the serving player, identified the peak jump frame, and accurately computed the ball's ground intersection using fitted parabolic curves. The integration of traditional filtering methods proved effective in removing noise and maintaining consistent tracking over time, while sharp, directional changes in the ball's trajectory caused by the bounce served as natural markers for dividing pre and post contact motion segments.

4.2 Limitations, Challenges, and Future Work

While this research demonstrates that a cost-effective, single-camera system can accurately estimate tennis serve speed using classical image processing, several limitations and challenges remain. The algorithm performs optimally when the ball exhibits a distinct angle change between its pre- and post- bounce trajectories; however, when this change is minimal, the system struggles to differentiate the segments, resulting in inaccuracies in serve speed calculation. Furthermore, the approach is quite sensitive to adverse recording conditions such as unintended camera movement or low-contrast environments. This can lead to false detections and the unintended filtering out the desired trajectory data. These issues highlight opportunities for future improvement, such as more robust trajectory models, enhanced intersection analysis techniques, optimized filtering parameters, and the integration of machine-learning based ball detection methods. Future efforts will focus on addressing these issues and transitioning the post-processing pipeline into a mobile application, ultimately providing an accessible, practical, and user-friendly tool for tennis serve-speed analysis in training environments.

REFERENCES

[1] V. Reno, N. Mosca, M. Nitti, Cataldo Guaragnella, and E. Stella, "Real-time tracking of a tennis ball by combining 3D data and domain knowledge," *2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pp. 1–7, Dec. 2016, doi: https://doi.org/10.1109/TISHW.2016.7847774.

[2] H. Zhao, S. Wang, G. Zhou, and W. Jung, "TennisEye," *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, Apr. 2019, doi: https://doi.org/10.1145/3302506.3310404.

[3] M. Fazio, K. Fisher, and T. Fujinami, "Tennis Ball Tracking: 3D Trajectory Estimation using Smartphone Videos," 2018. https://stanford.edu/class/ee367/Winter2018/fazio_fisher_fujinami_ee367_win18_report.pdf

[4] Y. Yang, D. H. Kim, and D. Choi, "Ball tracking and trajectory prediction system for tennis robots," *Journal of Computational Design and Engineering*, vol. 10, no. 3, pp. 1176–1184, Apr. 2023, doi: https://doi.org/10.1093/jcde/qwad054.

[5] X. Li and P. Huang, "Simulation of tennis serve behavior based on video image processing and wireless sensor technology," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, Jun. 2020, doi: https://doi.org/10.1186/s13638-020-01746-w.

[6] Y.-C. Huang, I-No. Liao, C.-H. Chen, and W.-C. Peng, "TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications." Available: https://arxiv.org/pdf/1907.03698

[7] J. Kelley, S. B. Choppin, S. R. Goodwill, and S. J. Haake, "Validation of a live, automatic ball velocity and spin rate finder in tennis," *Procedia Engineering*, vol. 2, no. 2, pp. 2967–2972, Jun. 2010, doi: https://doi.org/10.1016/j.proeng.2010.04.096.

[8] H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, and S.-Y. Lee, "Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video," *Journal of Visual Communication and Image Representation*, vol. 20, no. 3, pp. 204–216, Apr. 2009, doi: https://doi.org/10.1016/j.jvcir.2008.11.008.

[9] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: https://doi.org/10.1109/tpami.1986.4767851.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," 1996. Available: https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf

[11] R. Cross and N. Tauziat, "Introduction Ball Trajectories Factors Influencing the Flight of the Ball 42." Available: https://www.physics.usyd.edu.au/~cross/TRAJECTORIES/42.%20Ball%20Trajectories.pdf