POLITECNICO
MILANO 1863

Detection of tennis ball bounces from a single view video of a match

IACV project 2023-2024

Gloria Desideri and Lorenzo Aicardi

June 22, 2024

Table of contents

Introduction

Gloria Desideri and Lorenzo Aicardi

POLITECNICO MILANO 1863

Project Scope and Motivation

- **Scope:**
  - Detect bounce points of a tennis ball from a single-view video taken by a top camera.
  - Determine the true 3D coordinates of the points detected from the image.
- **Motivation:**
  - Enhance performance analysis and assist coaches and players in strategy building.
  - Improve umpiring accuracy with advanced line calling technology.
  - Enrich broadcasts with engaging visualizations.

Other Approaches

- Pure machine learning techniques, such as TrackNetV2, are commonly used.
- Bounce detection and 3D trajectory estimation often utilize stereo vision for triangulation.
- Some studies reduce the uncertainty of the ball's path by assuming a parabolic trajectory.
- Analytical methods leverage Y coordinate changes, effective for detecting bounces on the racket but less effective for field bounces.

Our Approach

- Propose a hybrid method combining Deep Learning and analytical techniques.
- Use Deep Learning to detect the ball trajectory and players.
- Estimate bounce coordinates with analytical techniques to improve player detection.
- Employ numerical methods to find 3D bounce points despite the single-view limitation.
- Model the trajectory using the RDP algorithm and apply filters to detected changes for accurate bounce coordinates.

Feature Extraction

Gloria Desideri and Lorenzo Aicardi

POLITECNICO MILANO 1863

Ball Detection Using TrackNetV2

- Used a pretrained TrackNetV2 model for ball detection.
- TrackNetV2 is designed for tracking high-speed, tiny objects like tennis balls.
- Combines features of Convolutional Neural Networks (CNNs) and Deconvolutional Neural Networks (DeconvNets).
- Utilizes a pre-trained VGG16 model for feature extraction.
- DeconvNet refines features for spatial localization, producing a heatmap indicating object presence likelihood.
- Softmax activation applied to DeconvNet output, yielding a probability distribution over 256 grayscale values per pixel.

Tracking Process

- Grayscale prediction transformed into a binary heatmap using a threshold.
- Channel with highest probability indicates object location.
- Hough Gradient Method applied to validate frames, ensuring the object appears only once.
- Object location represented by a central point, enabling continuous tracking.

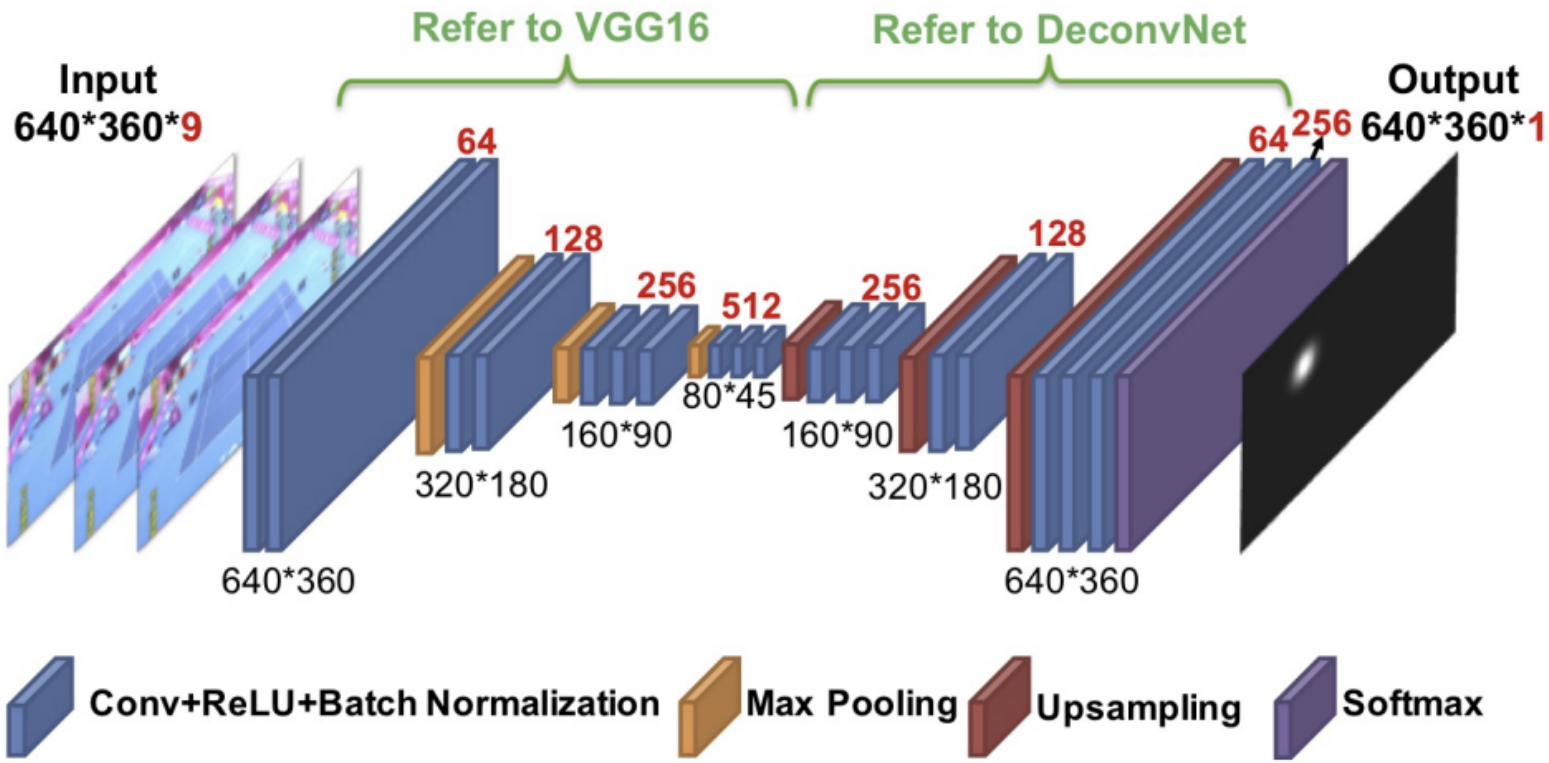


Figure: Tracknet

Court Detection: Binary Threshold and Pixel Filtering

- Convert input video frame to a grayscale image.
- Apply a binary threshold to highlight possible court lines based on intensity.
- Use pixel filtering to improve line identification accuracy.
  - Examine each pixel to see if it is part of a court line.
  - Compare the intensity of each white pixel to its neighboring pixels at a predetermined distance.
  - Keep the pixel if the intensity difference is large enough in both horizontal and vertical axes.

Court Detection: Hough Transform

- Use the Hough Transform to find lines in the binary image.
- Convert points in the image space to curves or lines in a parameter space.
- Each point votes for a possible line that might cross it.
- Employ the Probabilistic Hough Line Transform.
  - Picks points and, if they align with enough other points, extends them into line segments.
  - Effective at finding straight lines even with noise and gaps present.

Hough Transform Example

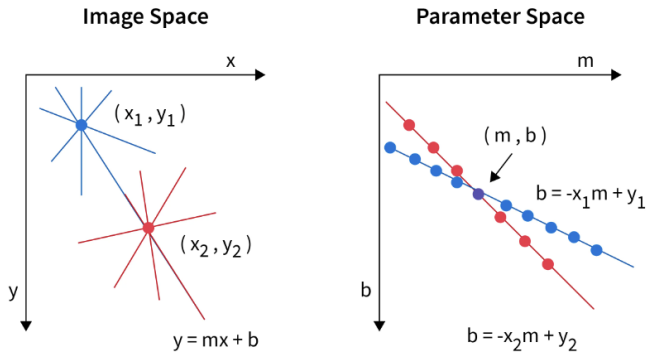


Figure: Example of the Hough Transform general idea

- Detected lines are divided into vertical and horizontal lines based on their orientation.
- Combine lines that are close to one another and probably correspond to the same physical line.
- Merging process ensures each court line is represented by a single, continuous line.

Players

- **Player Detection with ResNet**
  - Utilizes ResNet-50, a deep convolutional neural network with 50 layers.
  - Designed for image recognition and feature extraction.
  - Includes residual blocks for learning residual functions and facilitating deeper network training.
  - Shortcut connections in residual blocks help mitigate the vanishing gradient problem.
  - Architecture: initial convolutional layer followed by four stages of residual blocks.
- **Bounding Box Filtering for Fast-Moving Players**
  - Necessary due to the fast movement of players.
  - Differentiation between bottom and top player detection.

Players

- **Bottom Player Detection**
  - Leverages the detected court; the bottom part of the court is used as the Region Of Interest (ROI).
  - The box with the largest area detected by ResNet is assigned to the bottom player.
  - After the initial detection, previously found boxes determine a new restricted ROI.
  - A margin is added to the previously found box to account for movement.
  - The center of the previous box is calculated, and the box closest to the previous one is chosen as the new player box.

Players

- **Top Player Detection**
  - More difficult due to the presence of multiple people in the top part of the frame.
  - Player 2 is identified by detecting all persons in the top half of the court.
  - Movements are tracked across frames to calculate movement distances.
  - The person with the most consistent and significant movement is selected as player 2.
  - The SORT (Simple Online and Realtime Tracking) algorithm is used for tracking.
  - The object with the largest movement distance is detected as player 2.

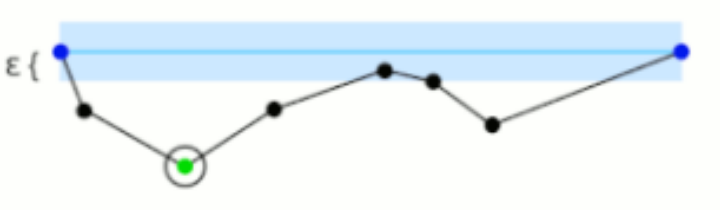
Bounce Detection

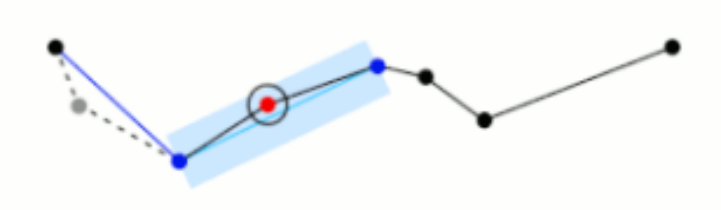
Gloria Desideri and Lorenzo Aicardi

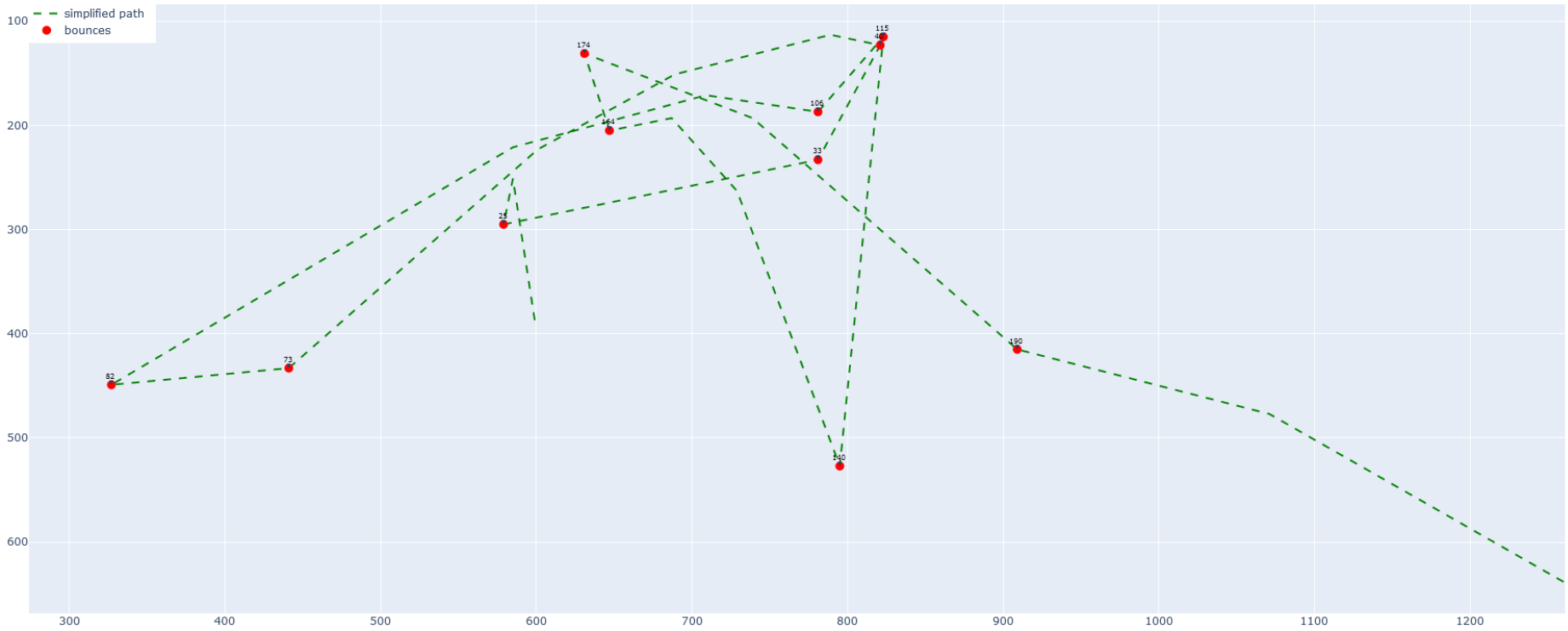
POLITECNICO MILANO 1863

RDP

- Recursively divide trajectory line, by taking the first and last points
- Take the furthest point from the segment connecting the 2 dots
- If the dot is further than a certain distance epsilon, keep it, otherwise discard it
- Iterate until all points have been analyzed.





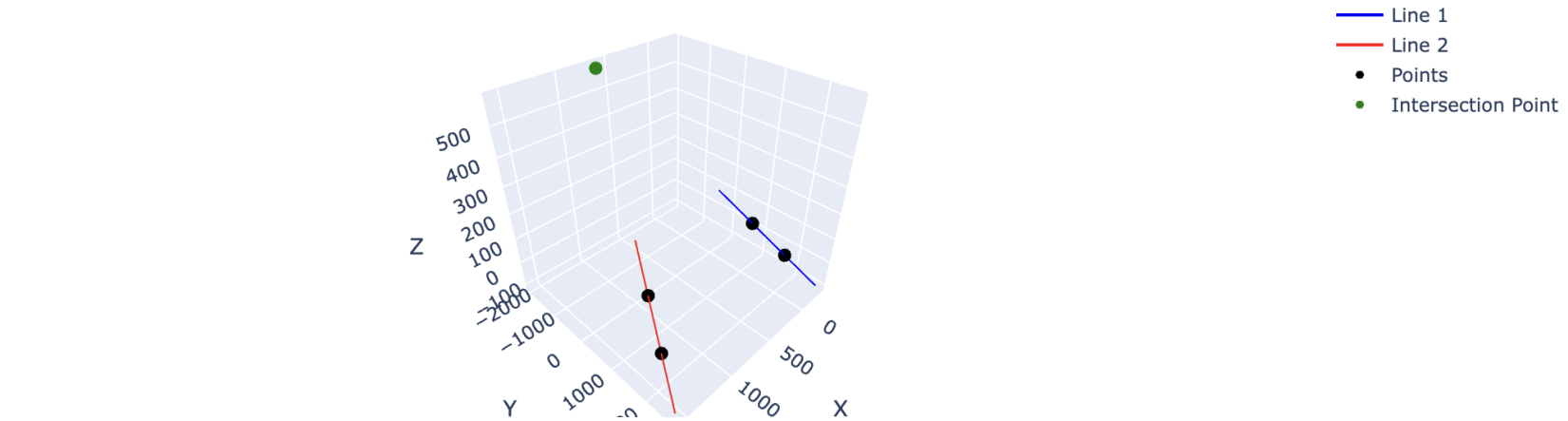

simplified path
bounces
100
200
300
400
500
600
300
400
500
600
700
800
900
1000
1100
1200
82
73
174
164
106
115
33
190
140

Filter trajectory changes

- Vectors in first and second quadrant $\rightarrow$ either peak of trajectory or player further from the camera hit the ball.
- Vectors in third and fourth quadrant $\rightarrow$ either bounce on the ground or player closer to the camera struck the ball (automatic bounce).
- If the first vector appeared in a quadrant on the opposite side of the second (so for example, the first in the first quadrant and the second in the third quadrant) we decided to confirm the bounce if the angle formed was less than 180° (i.e., it was "convex").
- Case of ball being served.

Camera center

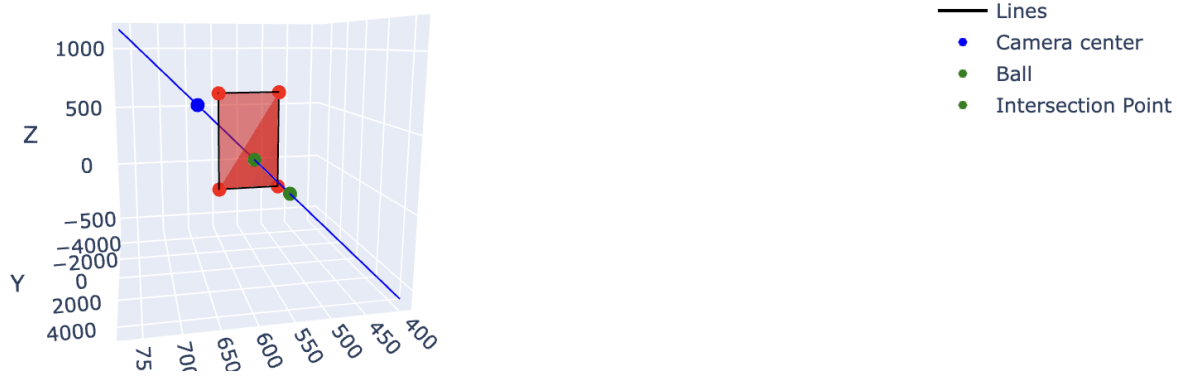
- The homography was estimated by taking the 4 corners of the tennis court.
- The top of the poles were known; we computed the inverse homography from the image to the plane to obtain their projection.
- The camera center was then found as the intersection between lines EE' and FF'.

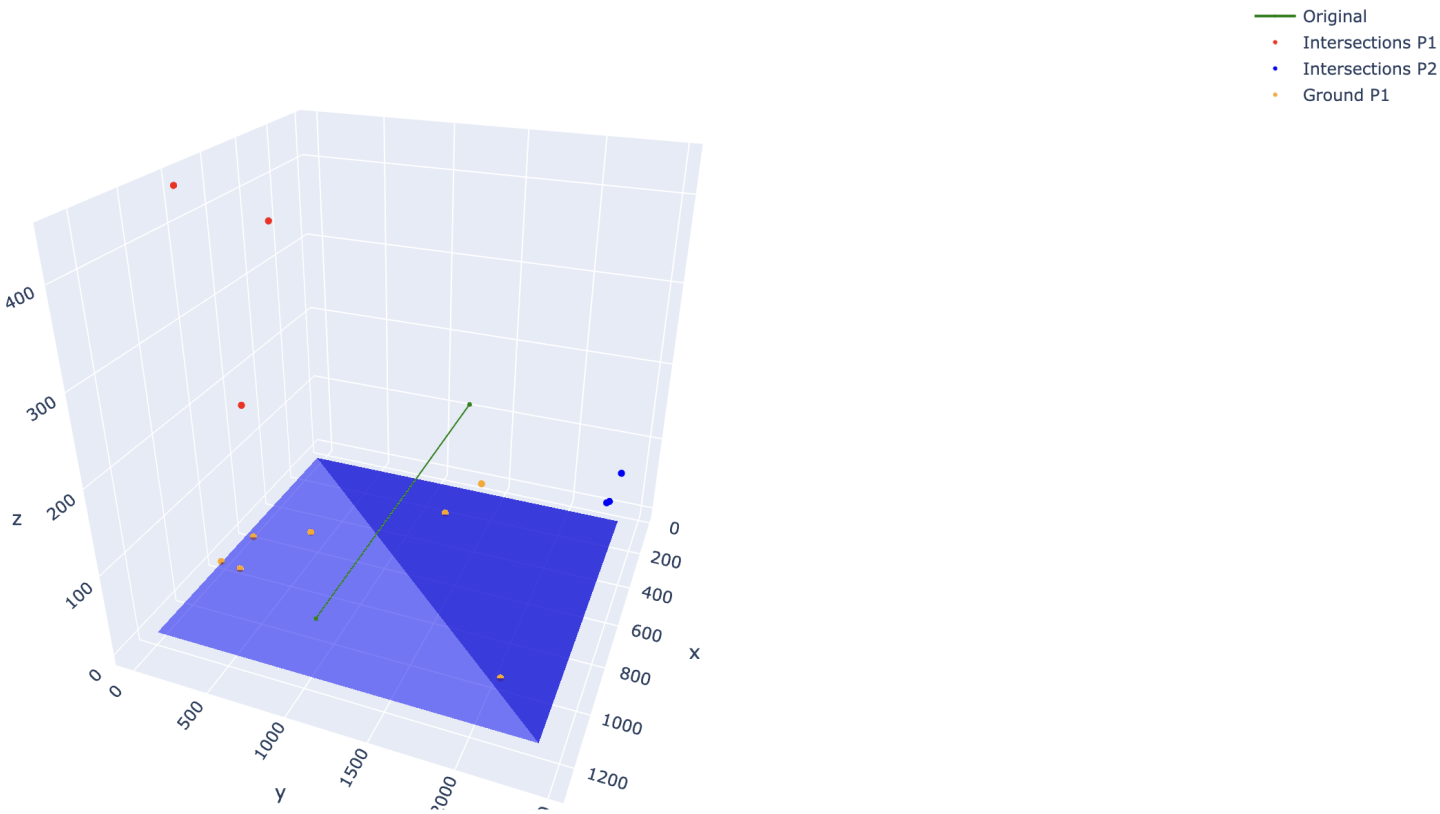

Lines, Points, and Intersection Point in 3D
Line 1
Line 2
Points
Intersection Point
Z
500
400
300
200
100
0
−100
−2000
−1000
0
1000
Y
0
500
1000
X

3D bounce coordinates

- Ball bounces on player's plane when it bounces on the racket.
- Retrieve player's feet position on the ground using homography, and build the plane perpendicular to the tennis court plane to identify player plane.
- The bouncing point is the intersection of the ray of view from the camera center to the image of the ball through the player's plane.
- For ground bounces, an homography is enough.

Lines, Points, and Bounding Box in 3D



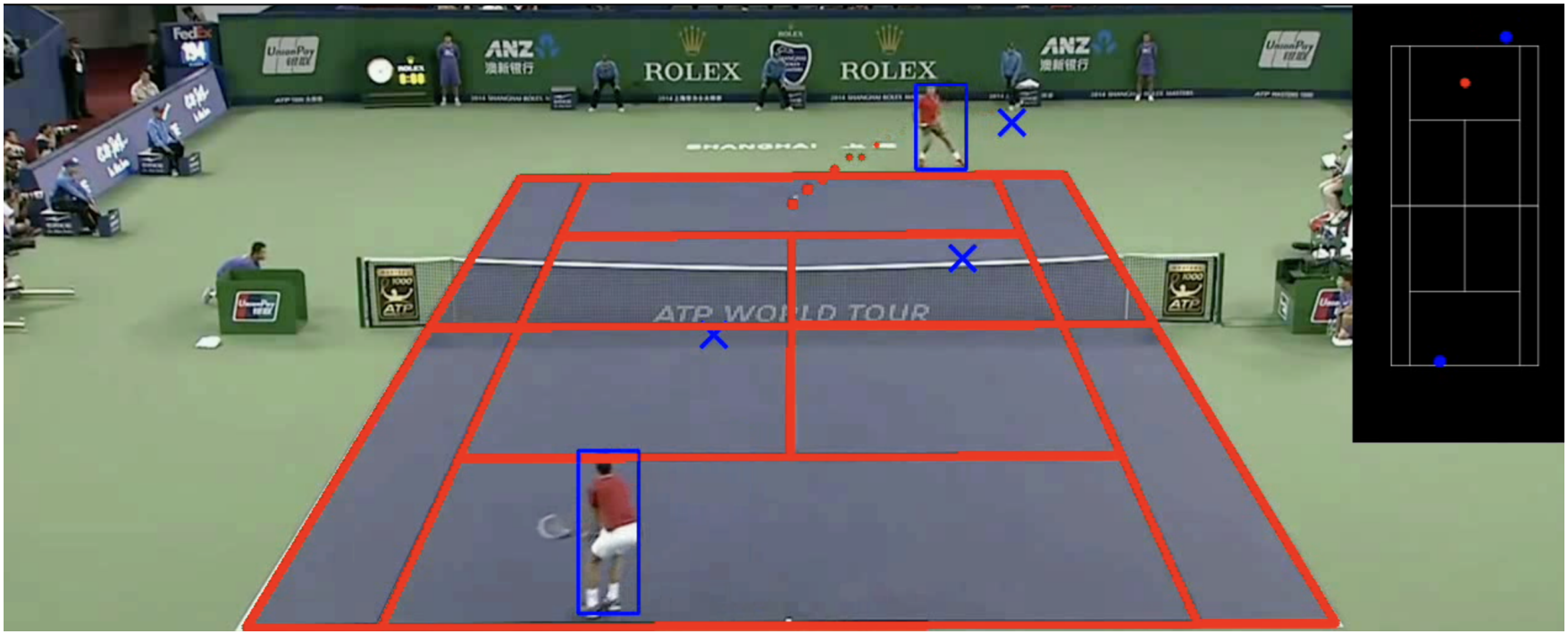

Original
Intersections P1
Intersections P2
Ground P1
z
400
300
200
100
0
0
500
1000
1500
y
0
200
400
600
800
1000
1200
x

Experimentation

Gloria Desideri and Lorenzo Aicardi

POLITECNICO MILANO 1863

- Tennis videos from WASB dataset and high resolution videos extracted from long matches.
- Gradual identification of encountered problems (neural network noise, peaks of trajectory marked as bounces).
- Finding a good tolerance parameter as compromise.


ANZ
ROLEX
ROLEX
ANZ
ATP WORLD TOUR

Conclusions

- Player detection is highly dependent on the court detection.
- 3D accuracy depends on the points chosen for the homography. Detecting automatically the points of the court and the poles would make for better accuracy.
- Further reducing the noise of the neural network would make for better trajectory analysis.

Repository URL:
https://github.com/LorenzoAicardi/TennisBounceDetector/tree/main