Visual Analysis of Sport Events

Video analysis and ball tracking of a volleyball action

Matteo D’Oria (10611067) – Lorenzo Castiglia (10578631)

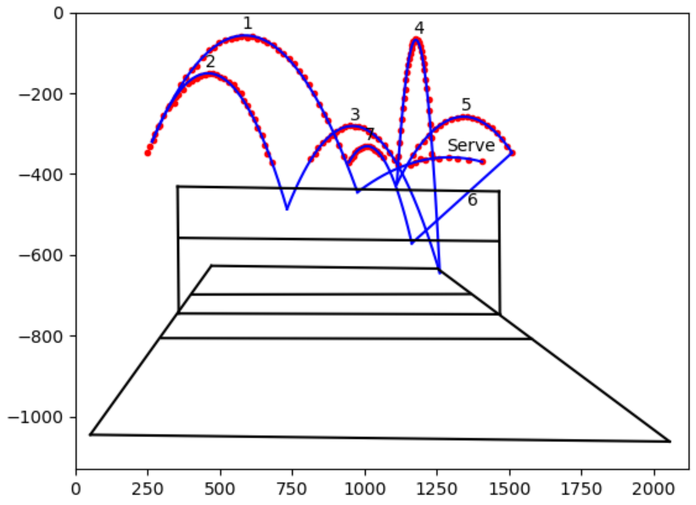
Introduction

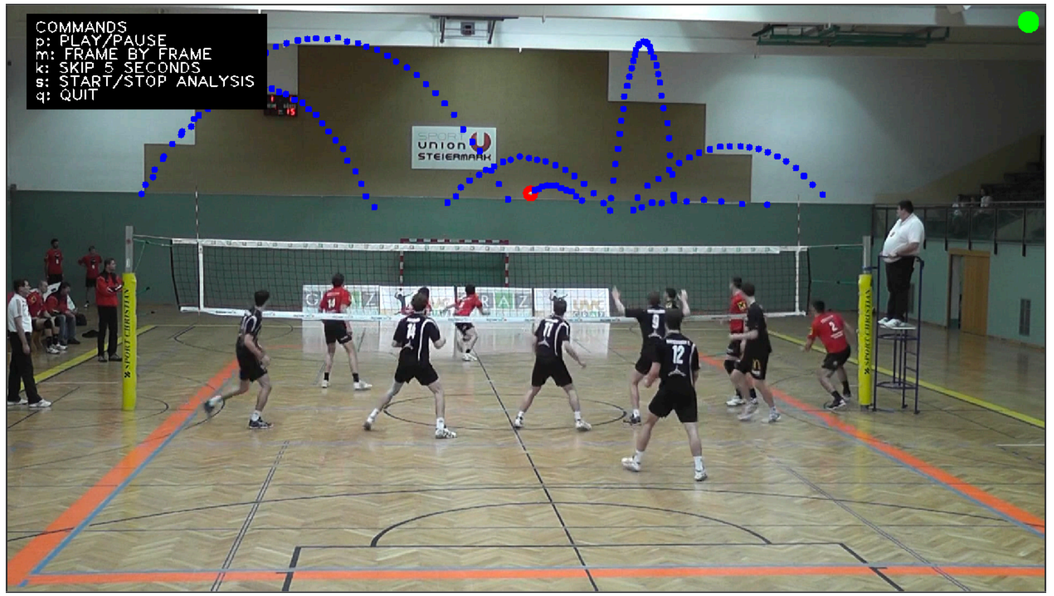
Nowadays, the use of automatized methods for analyzing sport events, both for technical and tactical purpose, is fundamental and constantly growing.

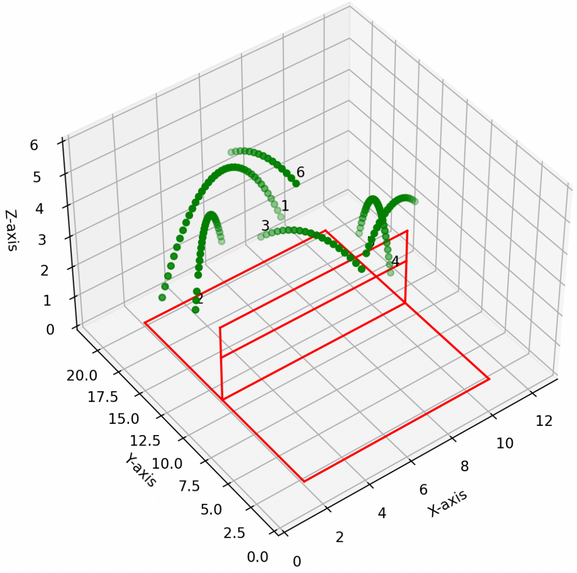
Coaches and players exploit many programs to improve their team and skills or to study the weaknesses of the opponent. A computer-based approach, as in almost every case, allows for a more rapid and complete analysis.

Introduction

In this work, we want to provide an algorithm for ball-tracking and 3D trajectory estimation of a single-camera volleyball action. The results can be deepened for some tactical/technical analysis, like set classification, serve and attack direction, dig quality and so on.







State of the art

- “*DeepBall: Deep Neural-Network Ball Detector*” by Komorowki et al. [2019]
- “*You Only Look Once: Unified, Real-Time Object Detection*” by Redmon et al [2015]
- “*Ball tracking and 3D trajectory approximation with application to tactics analysis from single-camera volleyball sequences*” by Chen, Tsai, Lee, Yu [2011]

Solution approach

In this project, we do not make use of Deep Learning techniques, instead we only exploit Computer Vision methods.

We work on the frame trying to extract the most “likely” ball, generating some trajectories and combining them to obtain the whole ball movement during the action.

The work is performed on a single fixed camera video, capturing the volleyball match from behind one of the 2 baselines.

Solution approach

The solution is divided in 3 phases:

1. ball detection in the frame
2. 2D trajectory extraction
3. 3D trajectory (approximation) elaboration

which will be presented in the following sections.

1. Ball detection

Given each frame of the action video, we need to find the ball.

As the ball follows a physical trajectory, it is possible to limit our search of the ball candidates only in the cleanest area of the frame, which is the zone above the net (ROI).

Although we don’t always have the ball position in the video, we can estimate where the ball might be using physics and geometry.

1. Ball detection

First of all, in the ROI, the candidates are extracted by subtracting from each frame the static objects, creating a mask of the moving ones.

Once we have the candidates, there are 2 situations:

a) the ball was never detected (new action)

b) the ball has already been detected

1. Ball detection

a) In the first case, we need to select the most likely ball, by exploiting its circular shape and dimension.

- each contour is assigned a score, which is the difference between the area of its minimum enclosing circle and its surface
- a sorted list of the contours is returned ranked by the score
- the “best” contour (the one most similar to a ball) is returned.

b) In the second case, it is possible to use the ball history to track the ball: so, among the candidates, the ball is chosen among the possible as the one nearest to the estimated position.

There are 2 cases in which the ball is lost:

- when it is out-of-frame
- when it is below the net

2. 2D trajectory extraction

Once we have a history of the ball position, we can perform the extraction of the trajectory in 2 dimensions.

In this phase, the key part is understanding when the new position is related to another parabola. This can be done by following the trajectory of the ball and detecting a change in the direction which is not the one we expected.

If the ball is simply far from the last position and it is going up, then it is quite easy to detect a new parabola.

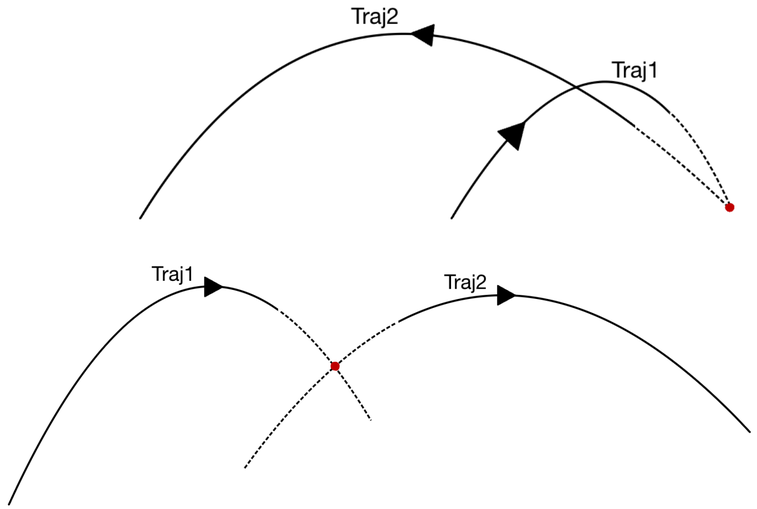
When the ball is near the last position (for example, for a set), then we need to work on the change of the direction of the ball, from downward to upward.

2. 2D trajectory extraction

Solved this phase, we have a set of points, divided in different trajectories.

For each set, a parabola is extracted interpolating the points. These parabolas have no start or end, so the next and final step is to find the starting and final points of each parabola, by intersecting them.

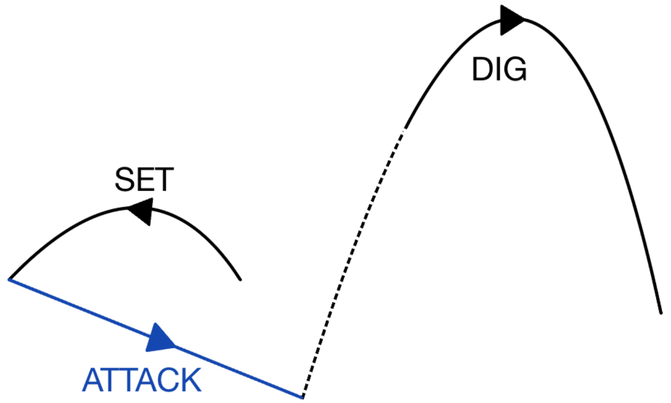
Finally, having for each parabola the equation, the starting and final point, it is possible to draw the complete trajectory of the ball.



2.1 Attack handler

For what concerns the attacks, we need to separate 2 kinds: the attack defended by the opposite team which leads to another action and the final attack.

For the first type, we decided to exploit once again geometry: since we have the two parabolae of the set and the dig, we can connect the 2 trajectories taking into account their directions.



2.1 Attack handler

For a final attack, instead, we do not have any other trajectory to take advantage of and moreover if we try to find the ball during its fast movement towards the ground, we'd encounter a lot of trouble due to the noise in the area below the net of the players.

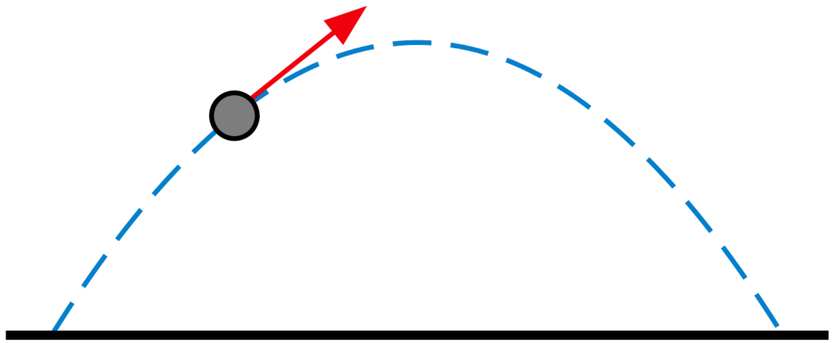
This could be treated by using Deep Learning techniques, like training a model to distinguish the ball from the players' bodies and then draw the attack trajectory.

Therefore, we decided to leave this for future implementation and not totally approximate the final attack.

3. 3D trajectory

Introduction

In a volleyball action the ball has a trajectory that can be divided into multiple sub-trajectories of nearly parabolic curves.



Using the corresponding physical equations and the camera projection matrix, we aim to build a system to find the 3D starting point of a sub-trajectory and the ball's initial velocity.

3.1 Projection matrix

A camera is a mapping from the 3D real world to the 2D image space.

A real world point in 3D space is represented as a homogeneous 4-vector $W = (X, Y, Z, 1)^T$, an image point as a 3-vector $u = (x, y, 1)^T$ and P represents the 3×4 homogeneous camera projection matrix.

The mapping from the 3D real world to the 2D image is written as

$$\vec{u} \times P\overrightarrow{W} = \vec{0} \quad \Rightarrow \quad \vec{u} \times \begin{bmatrix} P_1^T W \\ P_2^T W \\ P_3^T W \end{bmatrix} = \vec{0} \quad \Rightarrow \quad \begin{bmatrix} 0 & -W^T & yW^T \\ W^T & 0 & -xW^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0$$

3.1 Projection matrix

The matrix P has 11 degrees of freedom, so it is necessary to have at least 11 equations to solve for P.

The system obtained is

$$A \cdot \underline{P} = \vec{0} \Rightarrow \begin{bmatrix} A^1 \\ A^2 \\ \vdots \\ A^n \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = \vec{0}$$

Where $A^{2n \times 12}$ is the matrix containing the points correspondences and $\underline{P}^{12 \times 1}$ the projection matrix reshaped as a vector.

3.1 Projection matrix

We chose 14 non co-planar point correspondences to create the system. Since point n.10 is out of frame, we obtain it using projective geometry.

3.1 Projection matrix

With the 14 non-coplanar point correspondences obtained above, we can solve for $\underline{P}$ using the direct linear transform (DLT) algorithm.

The solution is obtained from $argmin_{\underline{P}}\|A\underline{P}\|_2$

$$\begin{cases} argmin_{\underline{P}}\|A\underline{P}\|_2 \\ [U, D, V^T] = SVD(A) \end{cases} = argmin_{\underline{P}}\|UDV^T\underline{P}\|_2 = argmin_{\underline{P}}\|DV^T\underline{P}\|_2$$

Since U is orthogonal it doesn't affect the solution.

To avoid the null solution we constrain its norm to 1.

$$\begin{cases} argmin_{\underline{P}}\|DV^T\underline{P}\|_2 \\ \underline{y} = V^T\underline{P} \\ \|\underline{y}\|_2 = 1 \end{cases} \Rightarrow \begin{cases} argmin_{\underline{y}}\|D\underline{y}\|_2 \\ \|\underline{y}\|_2 = 1 \end{cases} \Rightarrow \underline{y} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow \underline{P} = V_{12}$$

P is obtained by reshaping P , which is equal to the last column of $V^{12\times12}$.

3.2 Trajectory estimation

As stated previously, the ball trajectory can be divided into multiple sub-trajectories. In this project we perform an approximation by considering those sub-trajectories as parabolic curves, described by the following physical equations.

$$
\begin{aligned}
X_t &= X_0 + V_X \cdot t \\
Y_t &= Y_0 + V_Y \cdot t \\
Z_t &= Z_0 + V_Z \cdot t + g \cdot t^2/2
\end{aligned}
$$

$P_0 = (X_0, Y_0, Z_0)$ is the starting point of the sub-trajectory, $V = (V_X, V_Y, V_Z)$ the initial velocity of the ball in P_0 and (X_t, Y_t, Z_t) the ball coordinate at time t.

Using these equations we build a system with P_0 and V as unknowns for each sub-trajectory.

3.2 Trajectory estimation

Using

$u_t = (x_t, y_t, 1)^T$ and

$$W_t = (X_t, Y_t, Z_t, 1)^T = (X_0 + V_X \cdot t, Y_t = Y_0 + V_Y \cdot t, Z_t = Z_0 + V_Z \cdot t + g \cdot t^2/2)^T$$

We can write

$$\begin{bmatrix} 0 & -W_t^T & yW_t^T \\ W_t^T & 0 & -xW_t^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0$$

Obtaining two equations for each point belonging to a sub-trajectory.

3.2 Trajectory estimation

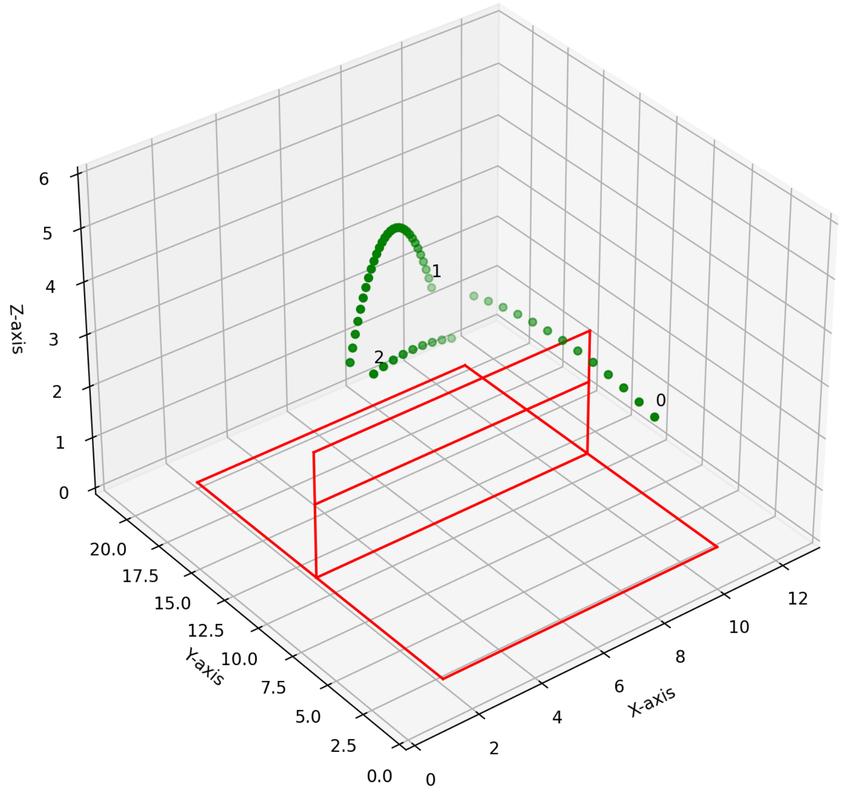
The system to solve for each sub-trajectory is

$$\begin{bmatrix} p_{11} - x_1 p_{31} & p_{11}t_1 - x_1 p_{31}t_1 & p_{12} - x_1 p_{32} & p_{12}t_1 - x_1 p_{32}t_1 & p_{13} - x_1 p_{33} & p_{13}t_1 - x_1 p_{33}t_1 \\ p_{21} - y_1 p_{31} & p_{21}t_1 - y_1 p_{31}t_1 & p_{22} - y_1 p_{32} & p_{22}t_1 - y_1 p_{32}t_1 & p_{23} - y_1 p_{33} & p_{23}t_1 - y_1 p_{33}t_1 \\ \vdots & & \vdots & & \vdots & \\ p_{11} - x_N p_{31} & p_{11}t_N - x_N p_{31}t_N & p_{12} - x_N p_{32} & p_{12}t_N - x_N p_{32}t_N & p_{13} - x_N p_{33} & p_{13}t_N - x_N p_{33}t_N \\ p_{21} - y_N p_{31} & p_{21}t_N - y_N p_{31}t_N & p_{22} - y_N p_{32} & p_{22}t_N - y_N p_{32}t_N & p_{23} - y_N p_{33} & p_{23}t_N - y_N p_{33}t_N \end{bmatrix} \begin{bmatrix} X_0 \\ V_X \\ Y_0 \\ V_Y \\ Z_0 \\ V_Z \end{bmatrix} = \begin{bmatrix} x_1(p_{33}\ gt_1^2/2 + p_{34}) - (p_{13}\ gt_1^2/2 + p_{14}) \\ y_1(p_{33}\ gt_1^2/2 + p_{34}) - (p_{23}\ gt_1^2/2 + p_{24}) \\ \vdots \\ x_N(p_{33}\ gt_N^2/2 + p_{34}) - (p_{13}\ gt_N^2/2 + p_{14}) \\ y_N(p_{33}\ gt_N^2/2 + p_{34}) - (p_{23}\ gt_N^2/2 + p_{24}) \end{bmatrix}$$

The least squares solution gives the vector $(X_0, V_X, Y_0, V_Y, Z_0, V_Z)^T$

3.2 Trajectory estimation

At this point we have the starting point and initial velocity of each sub-trajectory, by combining them with the parabolic physical equations we can derive a list of 3D points that belong to the trajectory and visualise them in a 3D plot.

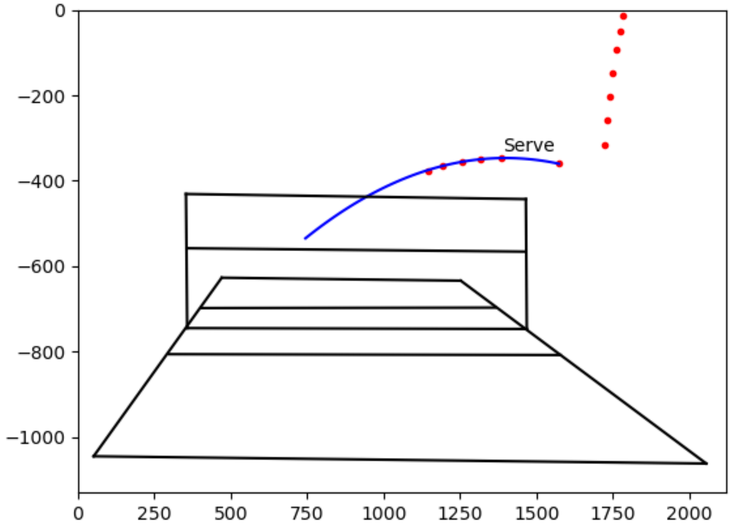


3.3 Limitations

1. We can only estimate points from ball candidates, because we need accurate timestamps. No attacks or interpolated points.

2. Intersection point of sub-trajectories outside of the ROI leads to discontinuity in the overall ball trajectory.

3. Fast moving trajectories provide very few ball candidates if fps are low, leading to underdetermined systems. The serve is almost always affected.

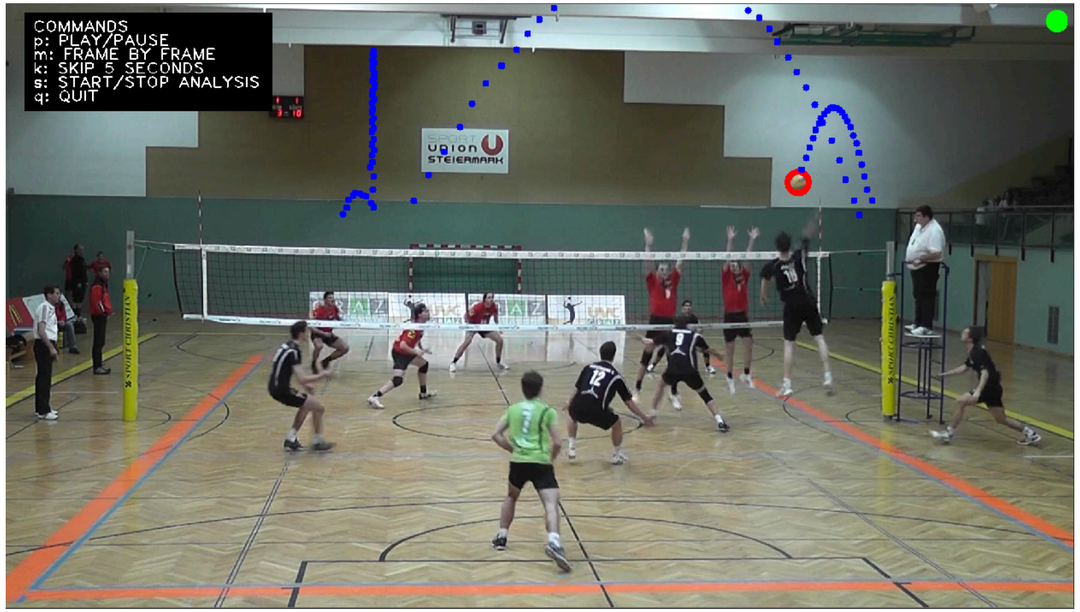


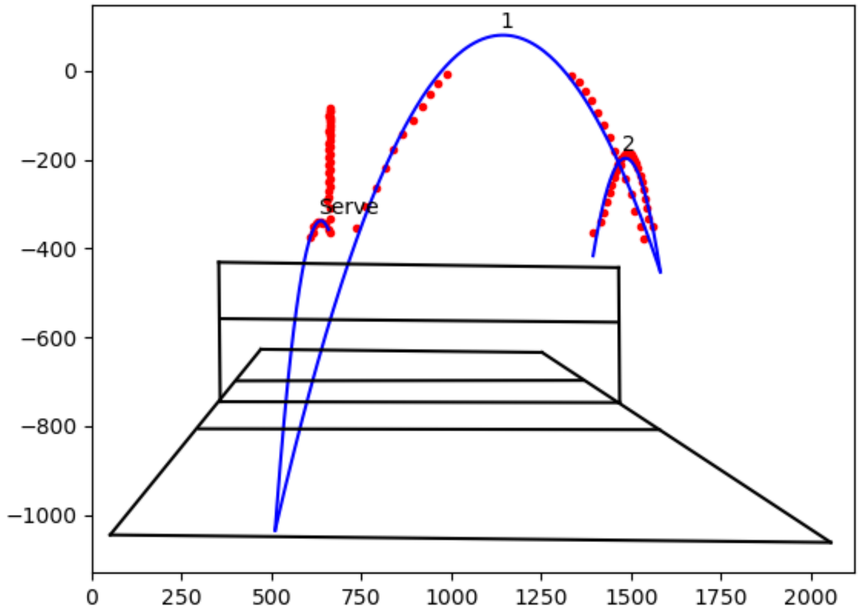
Experiments & results

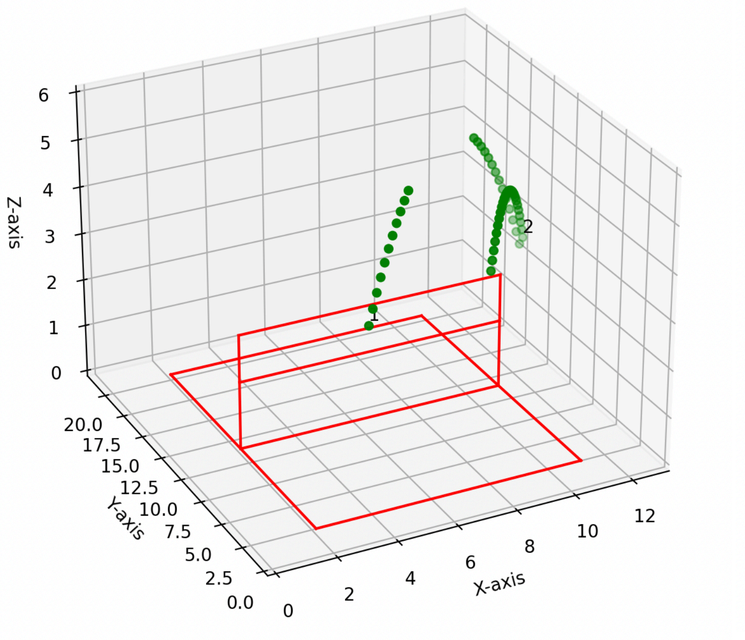
To test the video analysis system, we have used a volleyball match recorded by a camera posed behind one of the 2 baselines and from that we have extracted 7 actions.

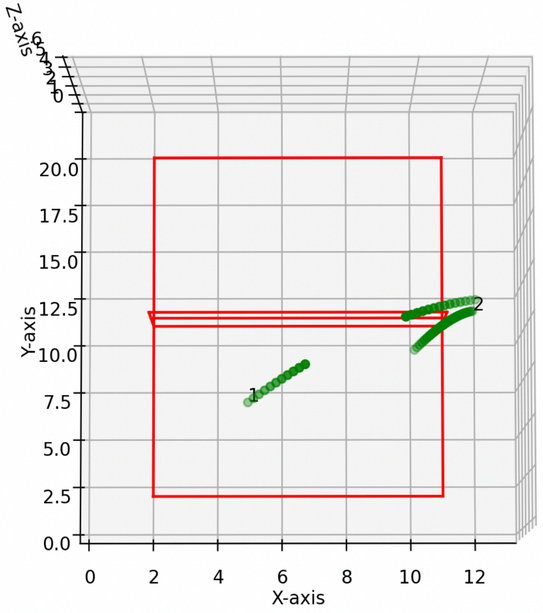
For each action, we will discuss what we obtain and what can be improved.

Action 1

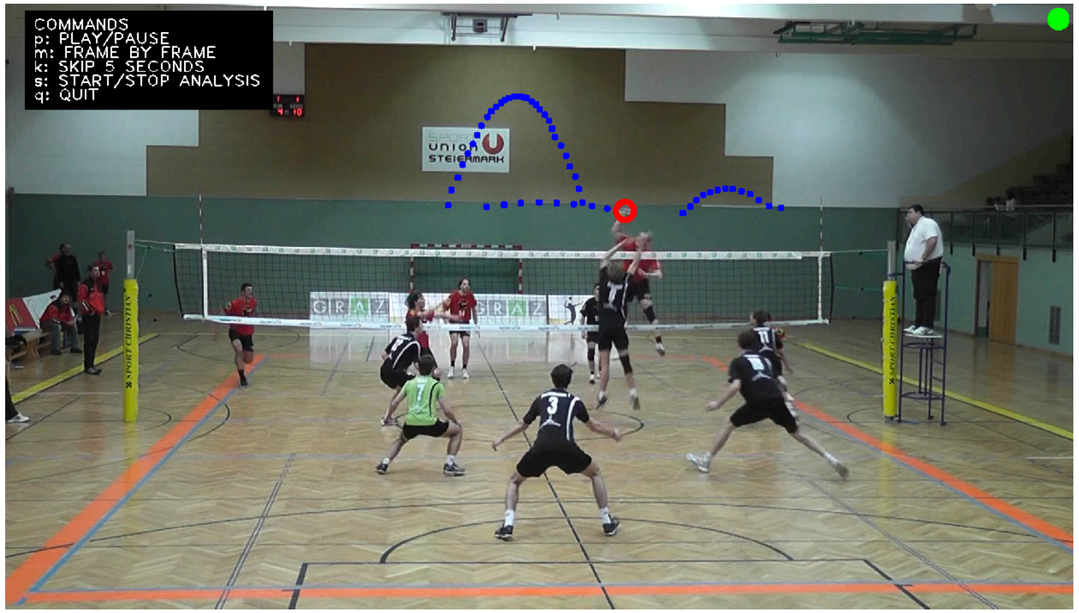

COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT

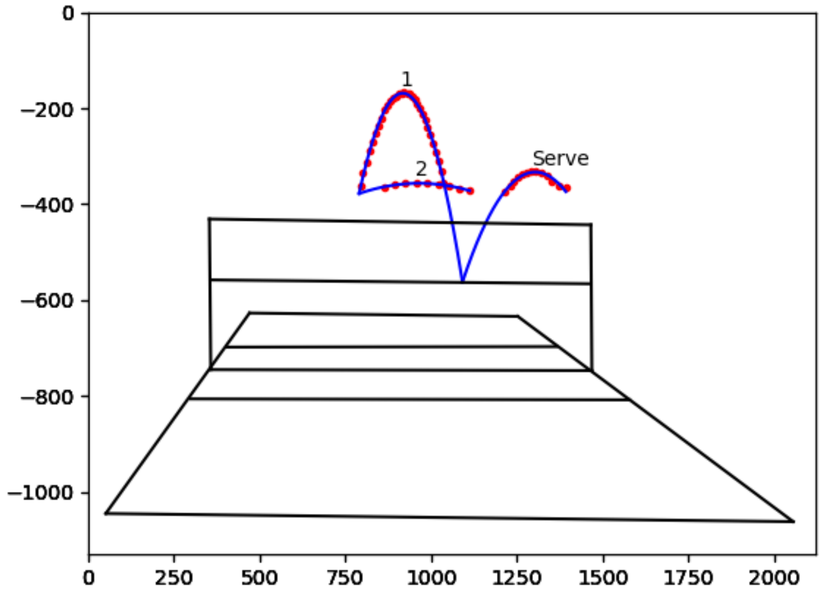

1
2
Serve
0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000

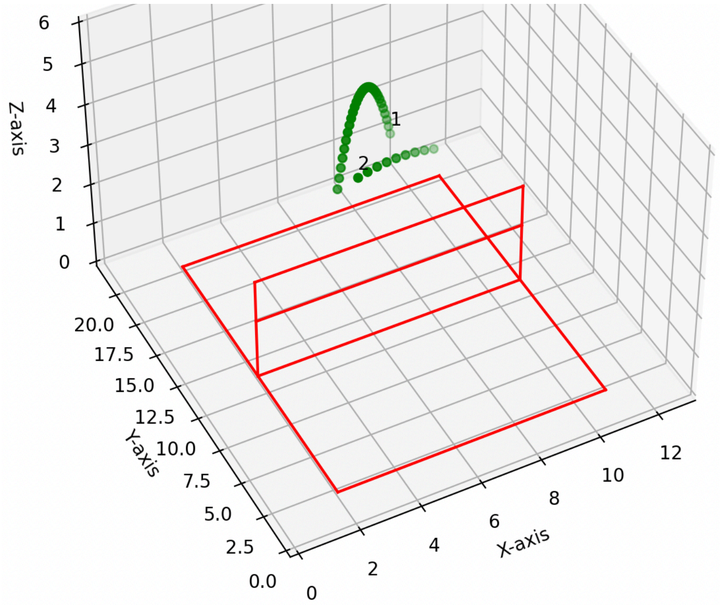

Z-axis
X-axis
Y-axis
1
2


Y-axis
X-axis
1
2

Action 2

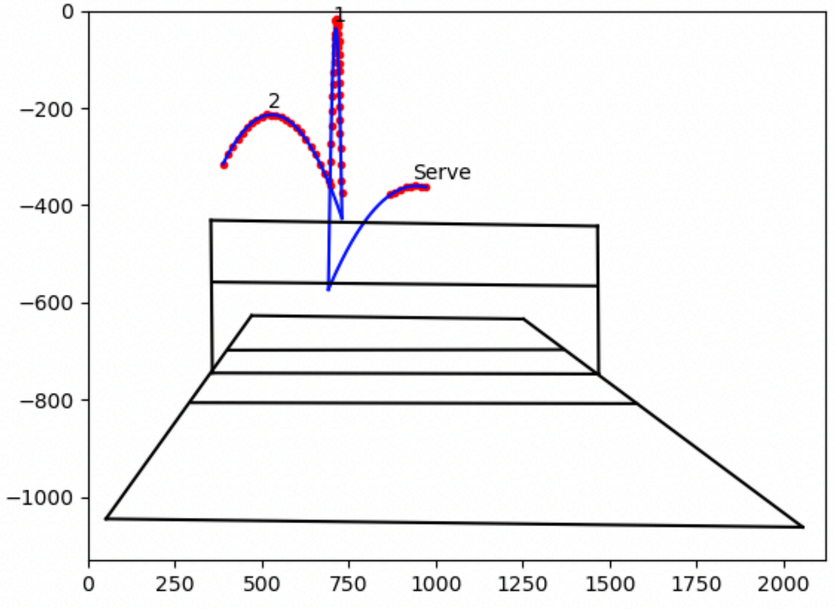

COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT

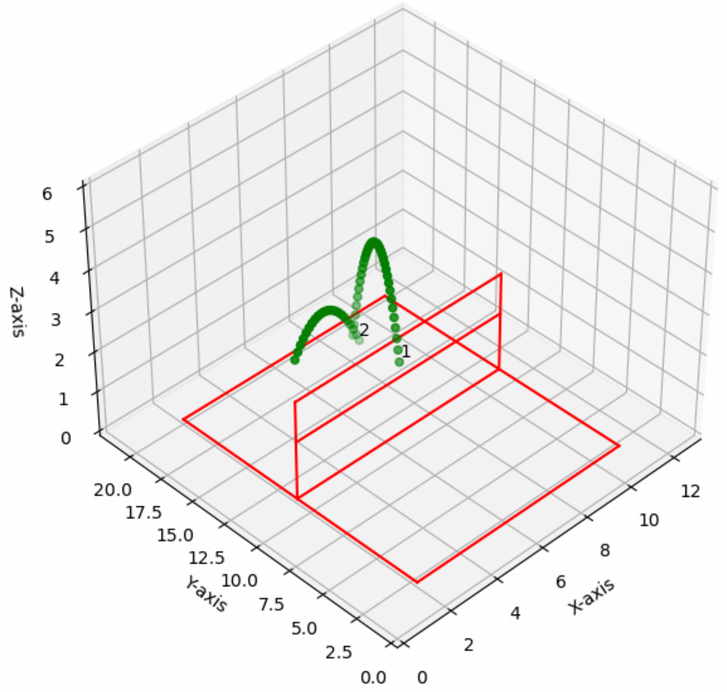

0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000
1
2
Serve


Z-axis
Y-axis
X-axis
1
2

Action 3

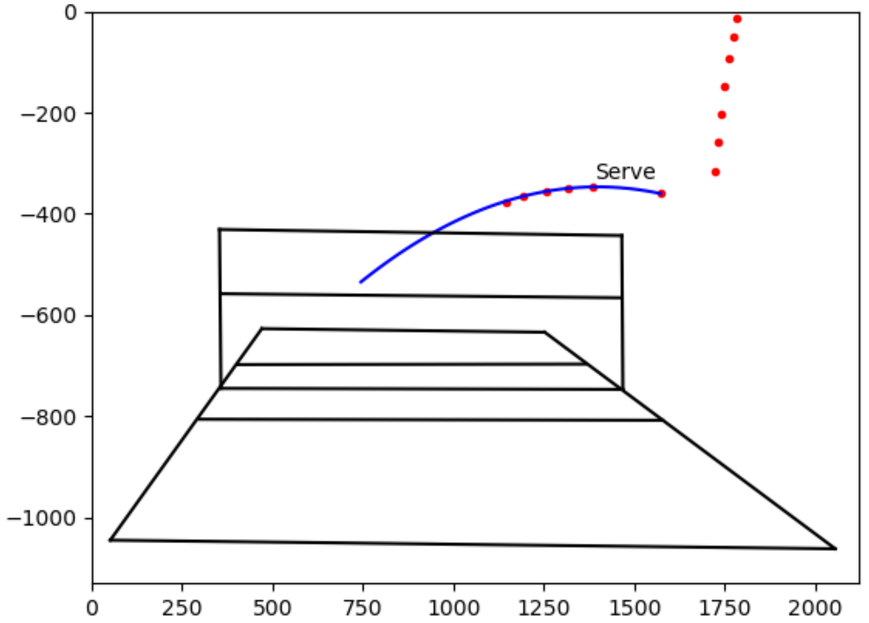

COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT


0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000
1
2
Serve

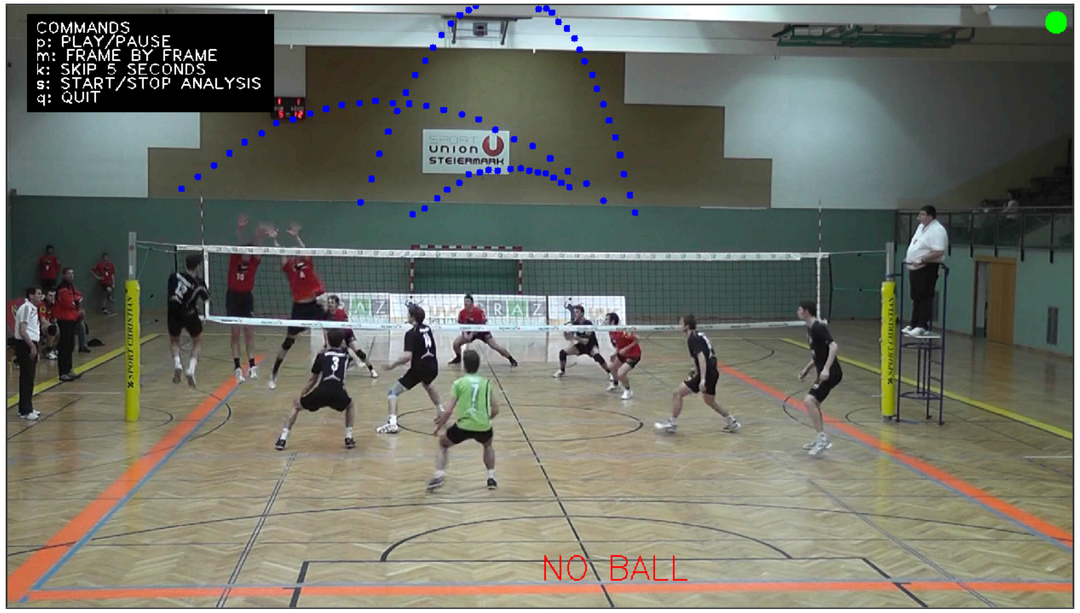

Z-axis
Y-axis
X-axis
1
2

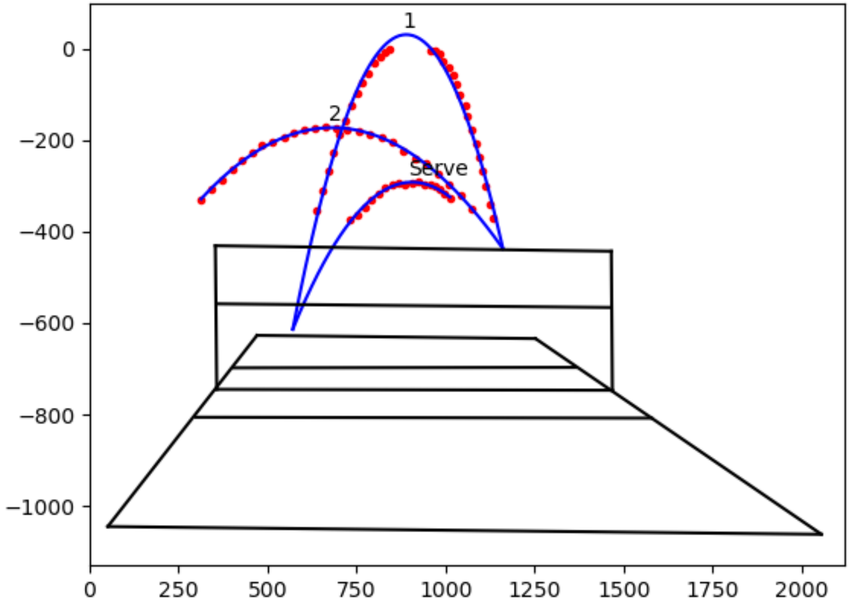
Action 4

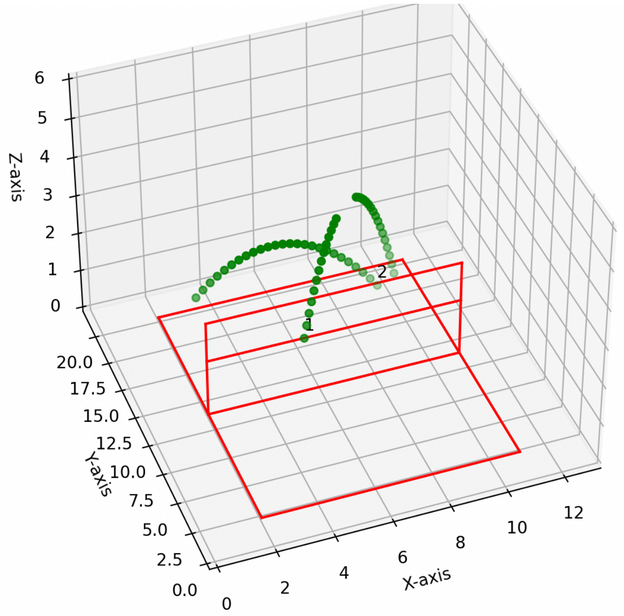

COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT
NO BALL

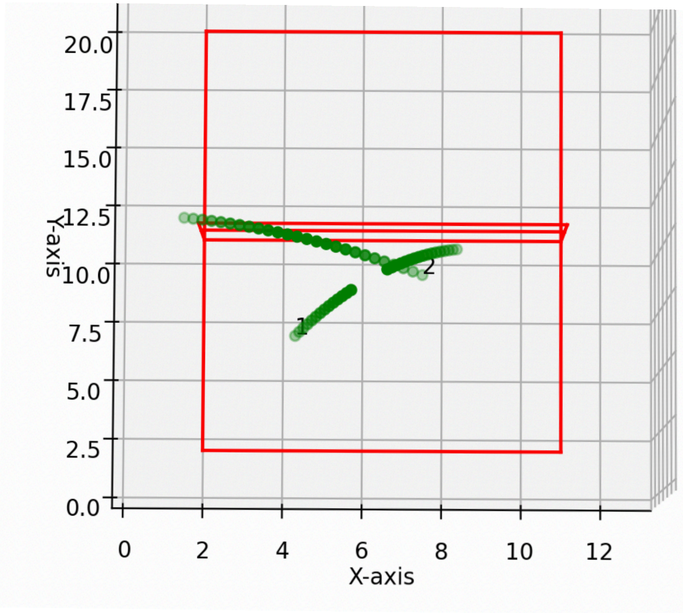

0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000
Serve

Action 5


COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT
NO BALL

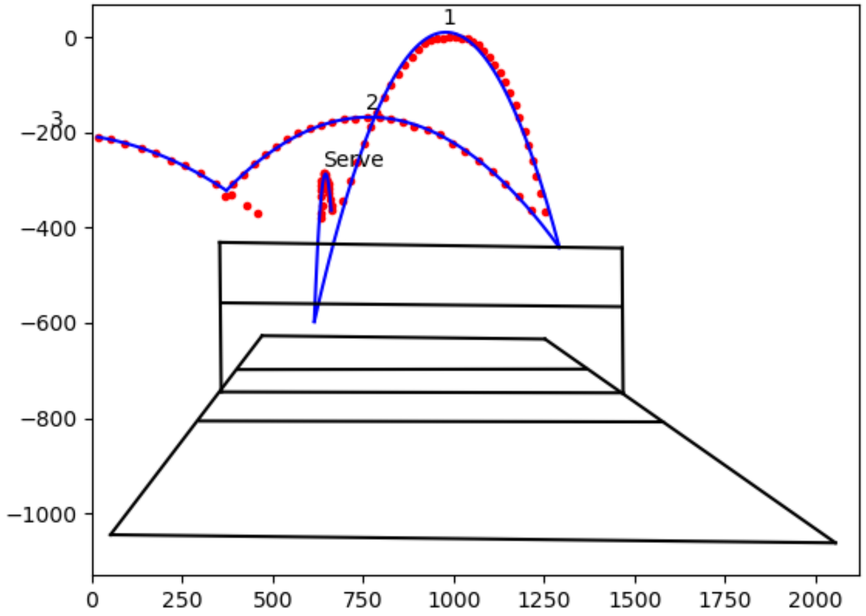

1
2
Serve
0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000

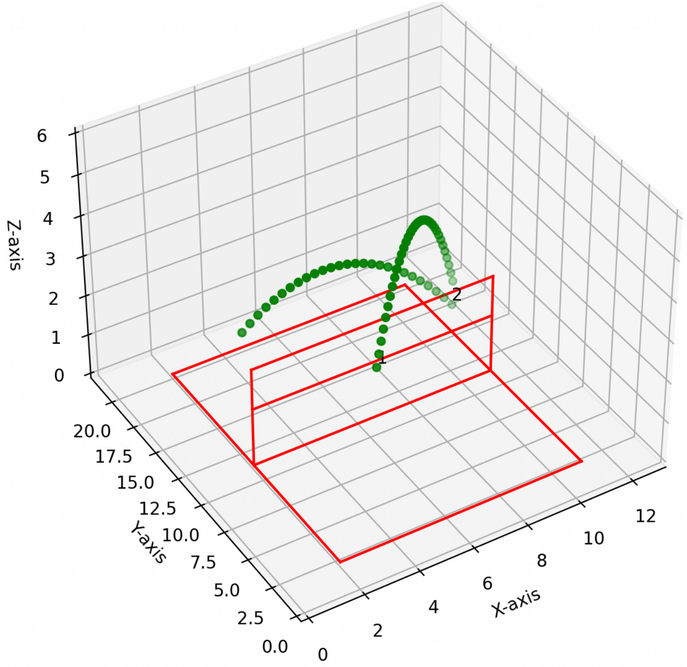

Z-axis
X-axis
Y-axis
1
2

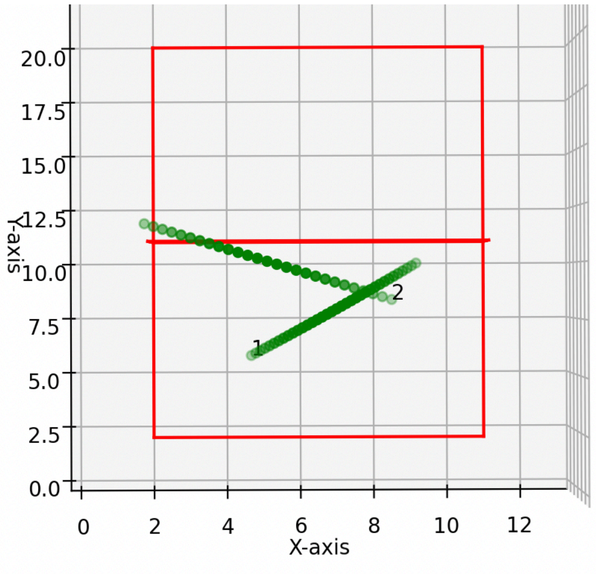

Y-axis
X-axis
1
2

Action 6

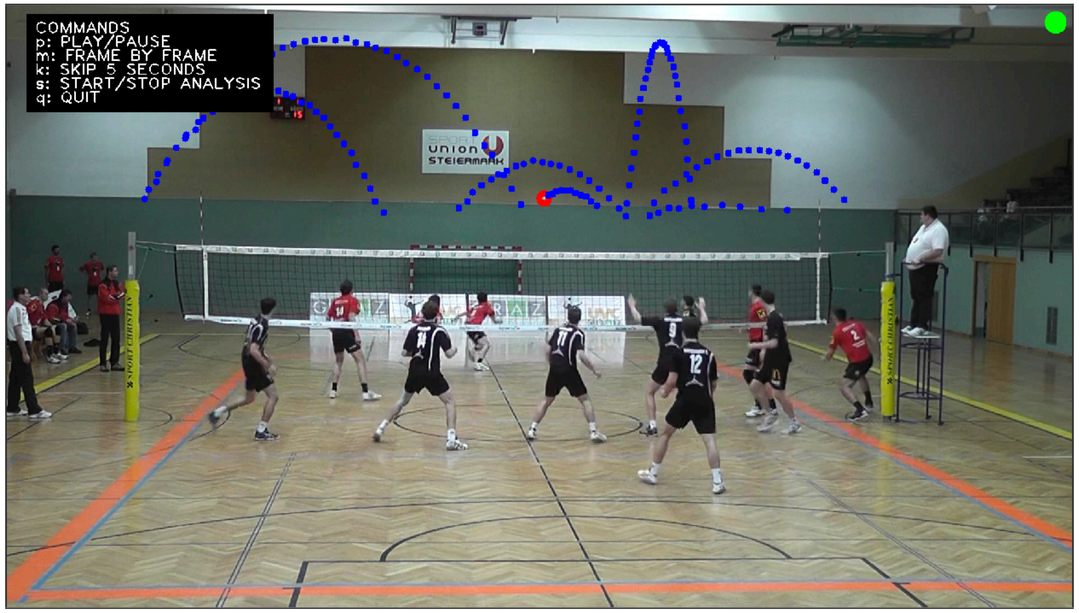

COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT

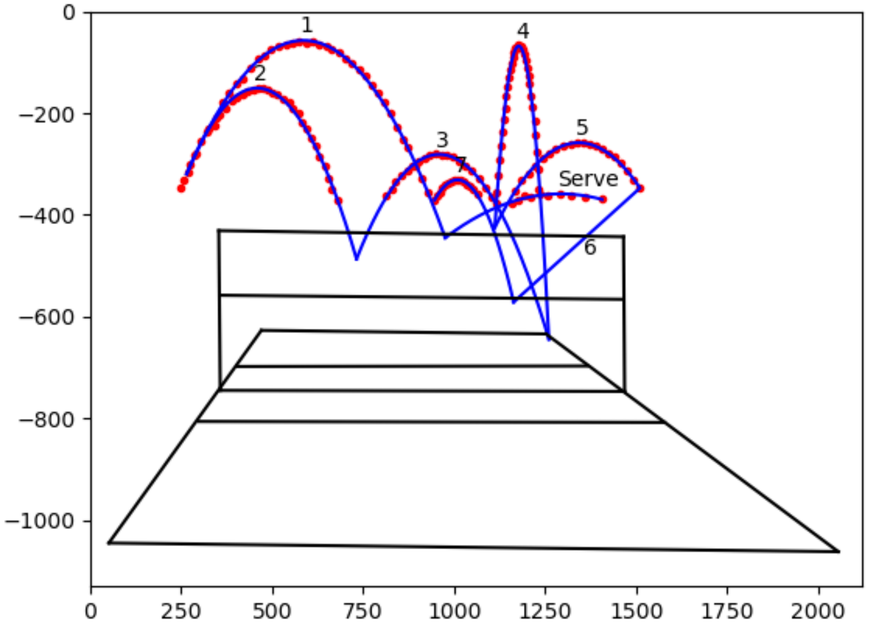

Serve
1
2
3
0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000

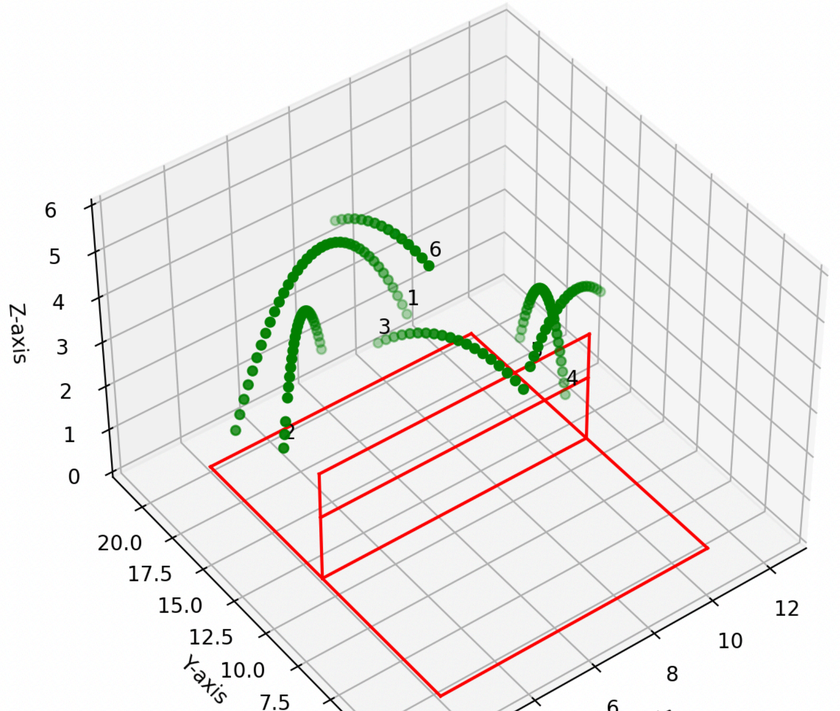

Z-axis
Y-axis
X-axis
1
2


Y-axis
X-axis
1
2

Action 7


COMMANDS
p: PLAY/PAUSE
m: FRAME BY FRAME
k: SKIP 5 SECONDS
s: START/STOP ANALYSIS
q: QUIT


0
−200
−400
−600
−800
−1000
0
250
500
750
1000
1250
1500
1750
2000
1
2
3
4
5
6
Serve


Z-axis
0
1
2
3
4
5
6
X-axis
20.0
17.5
15.0
12.5
10.0
7.5
Y-axis
6
8
10
12
1
2
3
4
6

Conclusion

- The ability to extract the 3D ball position from a video allows for real-time analysis and enhanced training strategies
- We obtained an accurate and reliable ball tracking and 2D reconstruction, allowing for realistic 3D reconstruction
- Improvements are necessary to overcome problems such as occlusion, rapid ball movement and video frame rate