Multi-Camera 3D Volleyball Tracking: Calibration, Detection and Trajectory Estimation

Ylenia Graziadei, Pietro Lechthaler

Department of Information Engineering and Computer Science
University of Trento, Italy
ylenia.graziadei@studenti.unitn.it, pietro.lechthaler@studenti.unitn.it

1 Introduction

This report presents a tool for 3D volleyball court reconstruction and ball tracking using a multi-camera system composed of 10 synchronized cameras strategically positioned around the court. The proposed pipeline integrates intrinsic and extrinsic camera calibration. Intrinsic calibration estimates each camera's focal length, principal point and lens distortion to correct image deformations, while extrinsic calibration determines the cameras' positions and orientations within a shared 3D coordinate system [6, 7]. For intrinsic calibration, the official OpenCV Calibration Method [2] is employed with a checkerboard pattern. Extrinsic parameters are optimized using Perspective-n-Point (PnP) algorithms and bundle adjustment by matching keypoints across multiple views.

By combining these calibrations with multi-view geometry and deep learning-based ball detection (YOLOv11 [5]), the system achieves robust 3D localization of the volleyball, even under challenging conditions such as occlusions and rapid motion. To model trajectory dynamics, a particle filter [1] is implemented, incorporating physics-based motion priors to enable accurate predictions during brief detection dropouts. The estimated trajectory is further refined using a low-pass filter.

The code is available in the GitHub repository.

2 Camera Calibration

Camera calibration [6, 7] is a fundamental step in 3D computer vision that estimates the geometric properties of a camera to ensure accurate measurements from 2D images to 3D world coordinates. It involves two main steps: intrinsic calibration, to compute internal parameters of each single camera and extrinsic calibration, which determines parameters of each camera relative to a global coordinate system.

2.1 Intrinsic Parameters Estimation

Intrinsic calibration utilizes a planar chessboard pattern of known dimensions (see Figure 1), following the OpenCV Calibration tutorial [2]. A set of high-quality images is selected based on sharpness to ensure reliable corner detection. The chessboard corners are automatically detected and refined to sub-pixel accuracy using an iterative optimization approach. These detected points, along with their corresponding real-world 3D coordinates, serve in a non-linear least-squares optimization to compute the camera matrix and distortion coefficients.

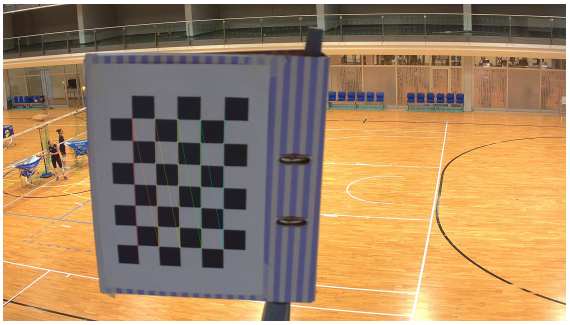

Fig. 1. Chessboard calibration pattern used for intrinsic camera calibration. The intersecting corners provide known reference points to estimate camera parameters: 8×6 squares, 28 mm each.

The resulting calibration provides the intrinsic camera matrix $\mathbf{K}$, which encodes the focal length (f_x, f_y) and optical center ((c_x, c_y)), as well as the radial distortion parameters (k_1, k_2, k_3) and tangential distortion parameters (p_1, p_2).

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$\mathbf{d} = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix}^\top \tag{2}$$

2.2 Extrinsic Parameters Estimation

The extrinsic calibration process [3] establishes the position and orientation of each camera within a unified world coordinate system, enabling accurate multi-view 3D reconstruction. This calibration leverages a set of precisely measured 3D reference points distributed across the volleyball court, with their corresponding 2D projections manually annotated in each camera view using a custom-developed annotation tool (see Figure 2). This tool is designed to streamline the manual marking of corresponding points across camera views, thereby improving both efficiency and measurement precision.

Fig. 2. Custom annotation tool for extrinsic calibration: enables manual marking of 3D reference points to find correspondences across camera feeds.

The calibration requires a minimum of four corresponding points visible in each camera pair. Using the intrinsic parameters, the PnP algorithm computes the rotation vector $\mathbf{r}$ and translation vector $\mathbf{t}$ that align each camera's coordinate system with the global reference frame. This transformation enables the conversion of 2D image observations into consistent 3D world coordinates.

$$s\mathbf{p} = \mathbf{K}\left[\mathbf{R}\ \mathbf{t}\right]\mathbf{P_w} \tag{3}$$

where:

- $\mathbf{P_w}$: 3D world point
- $\mathbf{p}$: 2D image pixel coordinates
- $\mathbf{K}$: camera intrinsic matrix
- $\mathbf{R}$: rotation matrix
- $\mathbf{t}$: translation vector
- s: scaling factor

The accuracy of the calibration is validated by measuring the reprojection error, which quantifies how well the computed model predicts the observed 2D image points: a low reprojection error confirms that the camera model accurately represents the imaging process.

3 Multi-View System Configuration

The custom-developed web interface shown in Figure 3 serves as the operational hub, allowing users to select any point in a virtual representation of the court and instantly visualize its corresponding position across all camera feeds.

The transformation framework operates through a bidirectional processing pipeline that maintains geometric consistency throughout the system. For world-to-camera projection, perspective homography matrices are computed for each individual camera using a RANSAC-based estimation approach [3]. This robust calculation requires a minimum of four corresponding point pairs per camera, ensuring reliable transformation even in the presence of minor annotation inconsistencies. The homography matrices effectively model the perspective distortion unique to each camera's viewpoint, accounting for varying angles and positions around the court.

The reverse transformation, known as camera-to-world back-projection, forms a critical link for 3D position estimation. This process combines data from multiple camera views to reconstruct the field point position in world coordinates. When a point is identified in one camera view, the system leverages the precomputed homographies to predict its appearance in all other cameras. This multi-view consistency check is particularly valuable when addressing temporary occlusions or detection uncertainties in individual cameras.

Fig. 3. Interactive multi-view projection system: selecting any point in the virtual court model instantly displays its corresponding projections across all camera feeds (green markers), enabling real-time spatial verification.

4 Ball Tracking

The objective of this phase of the project is to reconstruct the 3D trajectory of the ball during a match. To achieve this, the process begins by extracting the 2D coordinates of the ball in each frame of the recorded action. These 2D positions are then converted into 3D coordinates using the calibration parameters estimated in the previous sections. Finally, an approximate trajectory is computed and smoothing techniques are applied to refine the final result.

4.1 Ball Detection Algorithm

After conducting a thorough review of the state-of-the-art, the YOLOv11 architecture is selected for the ball detection task [5]. The first step involves creating a custom dataset using LabelImg [4], a graphical image annotation tool that allows users to draw bounding boxes around objects of interest and save the corresponding annotations in a YOLO-compatible format. The dataset consists of 1,015 frames, split into approximately 80% for the training phase and 20% for the validation phase. To ensure the reliability and generalization of the detection algorithm, specific actions tested in the final evaluation are intentionally excluded from these frames.

The training process occurs over 100 epochs, utilizing YOLOv11's built-in data augmentation capabilities, which automatically apply transformations such as rotation, scaling and color variation to increase dataset variability and improve model robustness. Upon completion of the training, the model outputs a list of coordinates for each camera.

4.2 3D Position Estimation

The 3D position of the ball is estimated through triangulation of synchronized 2D detections from calibrated camera pairs. For each pair of cameras, the 3D coordinates are computed by solving the linear system derived from their projection matrices. The projection matrix for each camera is constructed by combining both intrinsic parameters ($\mathbf{K}$) and extrinsic parameters ($\mathbf{R}$ and $\mathbf{t}$) of the camera. The full projection matrix for each camera is constructed as $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, allowing the transformation of 3D world coordinates into 2D image points.

Using this setup, triangulation is performed via OpenCV functions. After estimating the homogeneous 3D coordinates of the ball from matching 2D detections across the camera pair, these coordinates are converted to Cartesian coordinates, resulting in the final 3D positions of the ball. An example of a detection is provided in Figure 4.

Fig. 4. Cropped frame (camera 2) showing detected ball (red bounding box) and detection confidence.

4.3 Trajectory Estimation

For the estimation phase, the Particle Filter [1] is chosen due to its effectiveness in handling uncertainties and occlusions common in sports scenarios, where detections may be intermittent or noisy. This probabilistic approach maintains multiple hypotheses about the ball's state (position and velocity), with each hypothesis updated through a physics-based motion model that accounts for parabolic projectile motion under gravity. At each timestep, the particles are propagated according to the dynamic model and then reweighted based on their consistency with new observations, effectively managing temporary occlusions and sensor noise. The filter's output distributions capture both the estimated trajectory and its associated uncertainty.

Finally, a low-pass Butterworth filter is applied to the position estimates, preserving the natural motion characteristics while removing high-frequency jitter from the raw particle filter outputs. This results in smooth, physically plausible trajectories suitable for analysis, as illustrated in the plot in Figure 5.

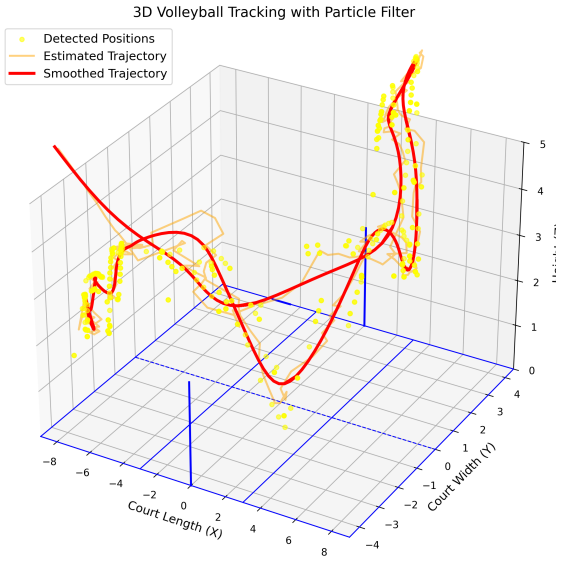

Fig. 5. Ball trajectory reconstruction showing: (1) raw triangulated positions (yellow), (2) Particle Filter estimates (orange) and (3) final smoothed path (red) after low-pass filtering.

5 Conclusion and Future Work

The developed system demonstrates robust 3D volleyball tracking through multi-camera calibration and particle-filter-based trajectory estimation. Although effective for single-ball scenarios, future enhancements could address more complex match conditions by integrating advanced detection models capable of managing occlusions, multiple balls (e.g. during training drills) and player interactions. Further optimizations may include real-time processing capabilities and adaptive filtering to accommodate varying ball dynamics during serves, spikes and blocks. The modular design of the system facilitates seamless integration of these improvements while preserving the core calibration framework.

References

[1] Jung Uk Cho et al. "A real-time object tracking system using a particle filter". In: *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2006, pp. 2822–2827.

[2] OpenCV Team. *Camera Calibration*. Last Access: 2025-03-27. OpenCV.

[3] OpenCV Team. *Camera Calibration and 3D Reconstruction*. Last Access: 2025-03-27. OpenCV.

[4] Tzutalin. *LabelImg: Graphical Image Annotation Tool*. `https://github.com/HumanSignal/labelImg`.

[5] Ultralytics. *YOLOv11 Model Overview*. Last Access: 2025-04-21. Ultralytics.

[6] YM Wang, Y Li, and JB Zheng. "A camera calibration technique based on OpenCV". In: *The 3rd International Conference on Information Sciences and Interaction Sciences*. IEEE. 2010, pp. 403–406.

[7] Zhengyou Zhang. "Camera calibration". In: *Computer vision: a reference guide*. Springer, 2021, pp. 130–131.