IACV Project

Marcelo Takayama Russo
Politecnico di Milano
marcelo.takayama@mail.polimi.it

Mateus Matzkin Gurza
Politecnico di Milano
mateus.matzkin@mail.polimi.it

Julius Becker
Politecnico di Milano
julius.becker@mail.polimi.it

Abstract

This report details the design and implementation of a system for reconstructing the 3D trajectory of a basketball from a static frontal perspective camera, dealing with the challenges of this particular setup. By combining geometric camera calibration, automated ball tracking, and physics-informed non-linear least squares optimization, we transform 2D image coordinates into a metric 3D space. The results demonstrate the ability to estimate initial velocity and projectile parameters, enabling advanced sports analytics.

1. Problem Formulation

1.1. Introduction

The analysis of basketball ball trajectories plays an important role in understanding shot quality and execution. Characteristics such as the entry angle, release velocity, and trajectory shape provide meaningful information about a player's shooting mechanics and consistency.

In this project, the ball trajectory problem was approached in a controlled setting, with data captured from recordings of our own shots in a chosen environment. This decision simplifies the experimental setup and ensures that the data acquisition process is fully aligned with the objectives of the study, allowing us to focus on the geometric and physical aspects of trajectory reconstruction.

1.2. Objective

The goal is to estimate the 3D position $\mathbf{P}_t = [x_t, y_t, z_t]^T$ of a basketball at each time step t using only a monocular video sequence. This is a classic problem in computer vision, as a single 2D projection $\mathbf{p}_t = [u_t, v_t]^T$ loses depth information. However, it is important to stress one specific particularity of the monocular perspective chosen. The single vanishing point frontal view is distinctly difficult due to depth-height ambiguity, which means movement on Y and Z axis can be confused. In general, the project aimed to overcome both the depth compression and axis ambiguity challenges simultaneously.

1.3. Motivation and Relevance

Traditional 3D tracking systems (e.g., Hawk-Eye) require expensive multi-camera setups. A monocular solution is highly relevant because:

- It enables analysis of archival footage or amateur recordings.
- It reduces hardware costs and setup complexity.

Additionally, the frontal view is a less explored case in the literature, and presents a challenge which can yield results significant to other fields and sports, no only basketball.

2. State of the Art

The analysis and reconstruction of basketball trajectories from video data have been addressed through a broad range of approaches, from classical geometry-based methods to modern learning-based techniques. These approaches differ in terms of computational complexity, data requirements, interpretability, and robustness to challenging visual conditions.

2.1. Classical Geometry-Based Approaches

Classical methods for ball trajectory analysis rely on explicit geometric modeling and physical constraints. In these approaches, the ball is typically detected in the image plane using shape-based techniques, such as the Hough Transform, and subsequently reconstructed in 3D through camera projection models and optimization or triangulation procedures.

When camera calibration parameters are available, the Perspective-n-Point (PnP) formulation enables the mapping

between image coordinates and world coordinates. Combined with projectile motion models, these pipelines yield physically interpretable trajectories and allow for quantitative analysis of shot characteristics such as release angle and velocity. Their main advantages include low computational cost, strong interpretability, and independence from large annotated datasets. However, their performance may degrade under low contrast, motion blur, or partial occlusions, and they often rely on manually defined geometric and physical constraints.

2.2. Learning-Based Detection and Tracking Methods

Recent work has increasingly adopted learning-based approaches for ball detection and tracking, leveraging deep neural networks such as YOLO [5] and TrackNet [3]. These methods have demonstrated high robustness in complex and unconstrained scenarios, including broadcast footage with dynamic camera motion and cluttered backgrounds.

Despite their effectiveness, learning-based methods typically require large labeled datasets and significant computational resources. Moreover, their end-to-end formulation often reduces interpretability and makes the explicit incorporation of physical or geometric constraints less straightforward. Nonetheless, modern learning-based models can facilitate the estimation or enforcement of such constraints implicitly, reducing the need for manual specification and improving robustness in challenging conditions.

2.3. Discussion and Positioning

While learning-based methods represent the state of the art in unconstrained and large-scale scenarios, they may be unnecessarily complex for controlled experimental settings. In such environments, geometry-based approaches remain competitive and provide greater transparency and physical interpretability, albeit at the cost of requiring explicit constraint definitions.

In this context, the present work adopts a hybrid but predominantly geometry-based pipeline. Classical circle detection, camera calibration, and physically grounded trajectory modeling form the core of the approach, while learning-based components are leveraged to enhance robustness and assist in the definition and enforcement of geometric and physical constraints. This design balances simplicity, reliability, and interpretability, while retaining flexibility to handle practical imperfections in the data.

3. Solution Approach

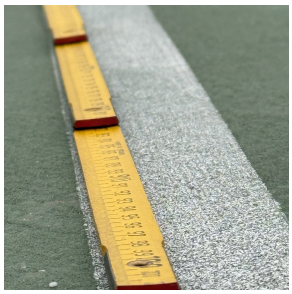
The approach developed solves the problem of reconstructing a 3D trajectory from a single 2D camera view by imposing strict physical and geometric constraints. The solution is divided into four stages: Definition of the World Coordinate System (Including definition of Court Structure), Camera Calibration, 2D Visual Tracking, and 3D Reconstruction.

3.1. Definition of the World Coordinate System

This step represents the foundation of the proposed pipeline, as it establishes the reference used to map image plane coordinates to 3D world coordinates. To enable this transformation, several key dimensions of the basketball court were measured during the video acquisition phase.

The measurements were obtained using a long measuring rule (1.6 meters maximum), which inevitably introduces a non-negligible level of uncertainty. However, the objective was not to achieve high-precision measurements, but rather to verify whether the court geometry was consistent with the official FIBA regulations. This approach ensures a coherent and physically meaningful world coordinate system while remaining aligned with the practical constraints of the experimental setup.

(a) Video Recording



(b) Measurement Example

Figure 1: Experimental setup and reference system

To verify whether the official FIBA dimensions could be reliably adopted, several well-known distances on the basketball court were measured during data acquisition.

Court Element	FIBA Official Value (m)	Measured Value (m)	Error (m)
Basket rim height	3.05	3.10	+0.05
Lane (paint) width	4.90	4.80	-0.10
Lane (paint) length	5.80	5.85	+0.05
Distance from endline to three-point line	5.00	5.07	+0.07

Table 1: Comparison between official FIBA court dimensions and experimentally measured values

Based on this comparison, we concluded that the court geometry was sufficiently consistent with the FIBA specifications to serve as the world coordinate reference. Consequently, the world coordinate system was defined by placing the origin (0,0,0) at the projection of the center of the rim onto the court floor.

It is important to note that all coordinates are expressed in meters. Under this convention, the camera position can be approximately defined as *(0, 14, 1.57)*, where the height

Figure 2: Definition of the world coordinate system in the image plane and 3D space

value of 1.57m was obtained through the same manual measurement with the measuring rule and therefore carries a certain degree of uncertainty.

Those estimated values will later be used for validation and consistency checks of the recovered camera matrix.

3.2. Camera Calibration

Before we can start tracking, the 2D pixel space must be mathematically linked to the 3D metric world. We model the camera using the standard Pinhole Camera Model. In the Pinhole Projection, a 3D point $X_w = [X, Y, Z]^T$ in the world coordinate system is projected onto the 2D image plane $x_{pix} = [u, v]^T$ via the projection matrix P:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot [R|t] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where

- K is the Intrinsic Matrix, containing the focal length (f_x, f_y) and optical center (c_x, c_y).
- R and t are the Extrinsic Parameters (Rotation and Translation) that define the camera's position relative to the court origin.

In order to use this projection, we need to first find the camera's pose (R, t). By identifying specific landmarks on the court (e.g., the corners of the paint) with known 3D coordinates X_w and their corresponding 2D pixel locations x_{pix}, we solve for the optimal rigid body transformation that aligns these points. The chosen points are explicit in red on the right-side image of Figure 2. To solve for the optmal transformation, we use the Perspective-n-Point algortihm (PnP) [4] on a initial estimation of the Intrinsic Matrix, where we consider the focal lenght to be approximately 1.2 times the width of the image. A problem with this algorithm is that initial PnP solutions often suffer from inaccuracies in relation to that focal length estimate. In order to get a better result, we implement a Non-Linear Least Squares Optimization to jointly refine the focal length f and the pose (R, t) after the PnP algorithm. The objective function minimizes the L_2 norm of the reprojection residuals using the Levenberg-Marquardt (LM) method:

$$\min_{f,R,t} \sum_i \|\text{project}(X_{w,i}, f, R, t) - x_{pix,i}\|^2 \tag{2}$$

This step ensures that the virtual camera model perfectly overlays the real-world court lines, and returns the estimated camera center in the real world coordinates system, giving a direct metric of how precise the calibration was by comparing it to the real measured camera position during the recording.

3.3. 2D Ball Trajectory Tracking

This stage aims to extract the 2D trajectory of the basketball across video frames. For this purpose, the Hough Gradient Method for circle detection was employed. Unlike data-driven approaches such as deep learning detectors (e.g., YOLO), this method relies solely on geometric and intensity-based image features and provides a lightweight and sufficiently robust solution under controlled recording conditions.

Image Preprocessing Each frame is first converted to grayscale and smoothed using a Gaussian blur. This preprocessing step reduces noise and suppresses small texture details that could otherwise be erroneously detected as circular structures, which is particularly important given the edge-based nature of the Hough Transform [2].

Circle Detection and Temporal Consistency The Hough Gradient Method is then applied to detect circular shapes parameterized by their center coordinates (u, v) and radius r, within a predefined radius range consistent with the expected apparent size of the basketball. All detected circles are treated as candidates for the current frame.

When multiple candidates are present, temporal consistency is enforced by selecting the circle whose center is closest to the ball position in the previous frame. Let (u_{t-1}, v_{t-1}) denote the detected center at frame $t-1$, and $(u_t^{(i)}, v_t^{(i)})$ the center of the i-th candidate at frame t. The selected detection is obtained as:

$$i^* = \arg\min_i \left\| \begin{bmatrix} u_t^{(i)} \\ v_t^{(i)} \end{bmatrix} - \begin{bmatrix} u_{t-1} \\ v_{t-1} \end{bmatrix} \right\|_2 . \tag{3}$$

This proximity-based criterion enforces temporal coherence and reduces false detections caused by unrelated circular objects.

Trajectory Construction and Resolution Consistency Once selected, the ball center (u_t, v_t) and radius r_t are

stored for each frame, producing a time-ordered sequence of 2D image plane observations that represents the estimated ball trajectory.

A key practical issue addressed in this stage is the resolution mismatch between calibration data and tracking data. While camera calibration is typically performed on high-resolution images (e.g., 4K), tracking often operates on downsampled or compressed video streams (e.g., 1080p). To ensure geometric consistency, an affine scaling transformation is applied to map tracked coordinates $(u_{\text{vid}}, v_{\text{vid}})$ into the calibration coordinate system $(u_{\text{cal}}, v_{\text{cal}})$:

$$u_{\text{cal}} = u_{\text{vid}} \cdot \frac{W_{\text{cal}}}{W_{\text{vid}}}, \quad v_{\text{cal}} = v_{\text{vid}} \cdot \frac{H_{\text{cal}}}{H_{\text{vid}}}. \tag{4}$$

This scaling preserves the geometric structure of the trajectory and ensures consistency with the intrinsic camera matrix K, enabling reliable use of the tracked points in subsequent 3D reconstruction and ballistic modeling stages.

Figure 3: Frame example of ball tracking

It is important to note that, although the Hough-based approach is well suited to the proposed setup, it also presents inherent limitations. In particular, the estimated ball radius is not always accurate, as the method relies on edge strength and local contrast rather than an explicit physical model of the object.

Additionally, detection performance degrades when the ball moves into darker or low-contrast regions of the image. In such conditions, the circular edges become less distinguishable from the background, which may lead to missed detections. Despite these limitations, the method proved sufficiently reliable for extracting the ball trajectory in the controlled recording conditions considered in this work.

3.4. 3D Reconstruction

Recovering the 3D trajectory of a basketball from a single monocular view is an ill-posed problem due to depth ambiguity and in the frontal view case, axial ambiguity between height and depth. To resolve this, we model the problem as a non-linear least squares optimization constrained by physical laws and semantic scene understanding [1]. The idea behind such a formulation is to use the physics constraints of a projectile trajectory in unison with the semantic understanding of limitations, ending and starting points, and general behavior of the shot trajectory. With that, it is expected for one to be able to add constraints to the projection of a parabolic shape in 3D space so that, when projected into 2D space it would fit the sequential movement of the ball from the perspective of the camera. It is precisely the task of adding and minimizing residuals, caused by the constraints and difference in position between the projection and the true 2D tracked trajectory that defines the least squares optimization used for the 3D reconstruction. Because of this, it is important to not only have a precise ball tracking process and camera 3D-to-2D projection but to also define well constructed constraints. The constraints defined for the problem are as follows:

- **Constraint 1**: Projectile Motion: The basketball, when in flight, is affected by gravity. Its path must correspond to the parabolic equations of motion:

$$P(t) = P_0 + V_0 t + \frac{1}{2}\mathbf{g}t^2 \tag{5}$$

  where $\mathbf{g} = [0, 0, -9.81]^T$. This reduces the search space from "arbitrary points" to "points that lie on a valid gravity-influenced curve".

- **Constraint 2**: Semantic Scene Anchors: The trajectory was automatically segmented into "Shot" and "Bounce" phases by analyzing the vertical velocity in image space. If a bounce is detected near the projected location of the rim, a Hard Constraint is enforced to indicate that the ball's height at that moment must be $Z \approx 3.05m$ (Rim Height). Otherwise, we enforce $Z = 0$ (Floor Level). These "anchor points" effectively lock the depth of the trajectory at specific keyframes.

- **Constraint 3**: Shooter Positioning: Using a lightweight YOLO pose detection model, the starting feet positioning of the shooter is estimated and used to indicate a starting value for the ball on the X and Y coordinates. This not only helps a quicker convergence towards the correct parabola, but constraints the positioning of the ball for the "Shot" phase.

Using these constraints, we can construct a multi-objective cost function that iteratively tries to minimize the calculated residuals for each point in the trajectory. The function is composed of five main terms, to each term a "weight" π_n is defined, it multiplies it and represents how much of a priority the term is for the fit:

- Reprojection Error: For every frame i, the physics engine predicts a 3D point P_i. The camera model projects P_i onto the image plane to get pixel coordinates (u_{proj}, v_{proj}). We compare this to the observed tracking pixel (u_{obs}, v_{obs}).

$$\pi_1(P(t|\theta) - x_{obs,t}) \tag{6}$$

where π_1 is a base weight.
This ensures the 3D curve looks exactly like the 2D observation when projected back onto the camera.

- Depth-from-Radius Consistency: The system estimates a rough depth Z_{est} based on the radius of the detected ball in pixels (r_{px}) using the pinhole formula:

$$Z \approx (f \cdot R_{real})/r_{px} \tag{7}$$

It compares this to the depth calculated by the physics model (Z_{model}). This term penalizes the scenario, where we track a tiny ball close to the camera or a giant ball far away.

- Geometric Constraints: Here we penalize deviations from the physical constraint established before. For example the bounce at the rim must be at 3.05m.

$$\pi_2(Z(t_{bounce}) - Z_{anchor}) \tag{8}$$

These constraints are treated as anchors and multiplied by a higher π_2 weight. This is because they are highly important for the restriction of the parabolic trajectory in 3D space when there is Y and Z ambiguity.

In the case of the bounce at the rim, it constraints both the height of and depth of the end and start point, respectively, of a "shot" and "bounce" phase, effectively limiting the possible true parabolas to 1 per phase).

- Physical Validity: The ball cannot go underground. If the predicted $Z(t)$ at any time is negative, a massive error is added.

$$\pi_3(\max(0, -Z(t_i))) \tag{9}$$

The same is applied for values of Y that fall far behind the backboard or behind the camera.

- Semantic Velocity Regularization: This term uses common knowledge as a basis to prevent the solver from choosing extreme velocities to fit noisy data. It adds a small residual penalty proportional to the velocity components (v_x, v_y, v_z) to avoid disproportionally elongated parabolas.

These error terms are concatenated to form the total residual vector R which is passed to the the optimization function, solved using the Trust Region Reflective algorithm (via `scipy.optimize.least_squares`), which allows us to set hard bounds on the parameters (e.g., ensuring the ball starts within the court boundaries).

The initial parameters fed into the optimization function are carefully designed to make it converge more precisely and efficiently. Firstly, the estimated optimal trajectory parameters $\theta_0 = [x_0, y_0, z_0, v_x, v_y, v_z]$ are passed through a linear system solver and chosen through the positioning constraints. This returns an educated guess for 6 parameters that define the parabolic trajectory at time $t = 0$:

- Position: $[x_0, y_0, z_0]$ (Where the ball started).
- Velocity: $[v_x, v_y, v_z]$ (How fast and in what direction it was launched).

After defined, the initial parameters are passed to the Trust Region Reflective Algorithm optimization function that will then, iteratively calculate the residuals and cost by generating the 3D positions θ_t at all future times t, derived deterministically from the initial parameters using gravity and applying constraints and regularization from all the five terms explained earlier. At each iteration, it will adjust the initial parameters to minimize the residuals, effectively solving for θ_0.

3.5. Implementation

The implementation of the system is modularized into four core Python scripts, leveraging libraries such as OpenCV for computer vision, NumPy for matrix operations, and SciPy for non-linear optimization.

Court Module: This module implements `CanonicalCourt2D` and `CanonicalCourt3D`, defining precise 2D and 3D coordinates based on FIBA regulations. It establishes the global world coordinate system and exposes a `canonical_points` dictionary, providing hard-coded 3D landmarks that serve as ground truth for the PnP calibration process. Additionally, a `plot_court()` method allows us to visualize the basketball court in the corresponding dimensions.

Calibration Module: The core `CameraModel` class encapsulates the mathematical conversion between 2D image space and 3D world space using Intrinsic (K), Rotation (R), and Translation (t) matrices. The calibration pipeline, driven by `calibrate_camera_pnp` function, utilizes `cv2.solvePnP` to generate an initial pose estimate. This estimate is subsequently refined via `scipy.optimize.least_squares`, which minimizes the reprojection error by optimizing the Focal Length and Extrinsics simultaneously. Helper classes (`CalibrationDisplay`) manage the user interface for manual point correspondence necessary between chosen court points and the `canonical_points` dictionary.

Tracking Module: The `BallTracker` class handles object detection and coordinate extraction. The current implementation utilizes a Hough Circle Transform (`cv2.HoughCircles`) combined with inter-frame heuristics (positional continuity) to filter false positives. A critical implementation detail is the resolution scaling logic: the module automatically computes scaling factors (s_x, s_y) to map coordinates from the video resolution to the resolution calibration image domain, considering that both may

have different pixel height and widths and ensuring spatial consistency before trajectory fitting. This module also implements the shooter starting point positioning function `get_ball_uv_start` which utilizes a auxiliary module called `PlayerModule`.

Trajectory Module: The `TrajectoryEstimator` acts as the physics engine. It employs a non-linear least squares solver to fit a gravity-constrained projectile model to the noisy 2D tracks. The optimization solves for the initial state vector $\theta_0 = [x_0, y_0, z_0, v_x, v_y, v_z]$ by minimizing a multi-objective cost function that includes reprojection error and physical validity constraints (e.g., $z(t) \geq 0$). The pipeline includes an automated segmentation algorithm (`detect_bounce_split_index`) that identifies bounces based on vertical velocity changes, applying semantic "Hard Constraints" to anchor the trajectory at the rim ($Z \approx 3.05m$) or the floor ($Z = 0m$).

4. Results

4.1. Camera Calibration

The calibration pipeline converged to a physically consistent camera model. The resulting Camera Center (C) indicates a device height of $1.58m$ and a distance of $12.75m$ from the hoop, which aligns perfectly with the handheld recording setup.

$$K = \begin{bmatrix} 2928.6 & 0 & 1434 \\ 0 & 2928.6 & 804 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0.12 \\ 12.75 \\ 1.58 \end{bmatrix} \tag{10}$$

Validation was performed by back-projecting known ground landmarks ($Z = 0$). The model demonstrated high fidelity, yielding a maximum spatial error of approximately $3cm$ on the Y-axis and $1cm$ on the X-axis for points within the court boundaries.

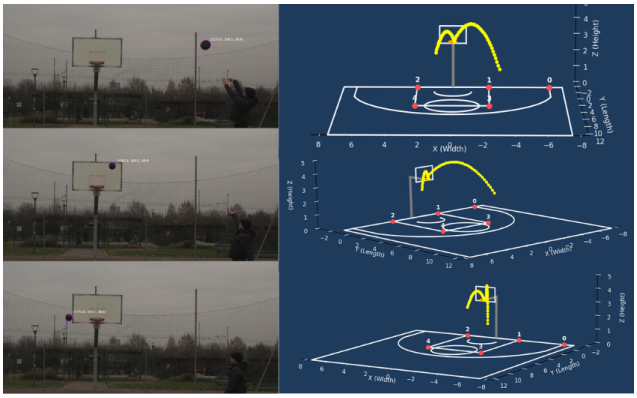


Figure 4: Shot frames and 3D Reconstruction

4.2. Trajectory Reconstruction

By combining the calibrated model with semantic anchors (YOLO-detected feet and rim location), the system successfully resolved monocular depth ambiguity. The optimization solver produced physically valid parabolic trajectories ($g \approx -9.81 m/s^2$), with accurate segmentation between flight and bounce phases. The constraints effectively locked the trajectory endpoints, resulting in a 3D reconstruction that aligns geometrically with the canonical court (Fig. 4).

5. Conclusion

Therefore, based on the results presented in the Results section, the proposed pipeline was able to successfully reconstruct the ball trajectory while respecting all the geometric and physical constraints considered throughout the project. The adopted approach primarily relies on classical geometric methods rather than heavily learning-based solutions, allowing for a deeper understanding of image formation and camera geometry instead of resorting to black-box models with limited interpretability.

Moreover, the developed framework can be seen as a foundational building block for a wide range of basketball-related computer vision applications. Ball detection, trajectory reconstruction, and camera calibration are core components in many sports analytics pipelines, and the methodology presented here provides a solid starting point for further extensions. Several potential directions for future work are discussed in the following section.

5.1. Future Work

5.1.1 Shot Outcome Prediction

A natural extension of the proposed pipeline is the prediction of shot outcome, i.e., whether a given shot results in a successful basket. By leveraging the reconstructed 3D trajectory and estimated ballistic parameters, such as release angle, initial velocity, and entry angle at the rim, it becomes possible to classify shots as made or missed. This task could be addressed using either threshold-based physical criteria or supervised learning models trained on trajectory-level features.

5.1.2 Spatial Analysis of Shot Locations

Another promising direction involves analyzing the spatial distribution of shot attempts across the basketball court. By associating reconstructed trajectories with their corresponding shooting locations in the world coordinate system, it would be possible to study accuracy patterns as a function of court position. Such an analysis could provide quantitative insights into shooting efficiency from different areas of the court and support location-aware performance evaluation.

References

[1] 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*, 20(3):204–216, 2009. 4

[2] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972. 3

[3] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng. Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications, 2019. 2

[4] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81, 02 2009. 3

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016. 2