Validation of a Monocular Computer Vision System for Basketball Shot Performance Analysis

NOAH VAN MAARE, University of Twente, The Netherlands

Analyzing basketball shot kinematics is crucial for improving player techniques and understanding motor skill adaptation, yet current high-fidelity motion capture systems are often inaccessible. This study presents the development and validation of a novel computer vision system designed to extract detailed shot kinematics of a basketball from a single, readily available camera. The system employs a multi-stage pipeline, including object detection, pose estimation, a hybrid approach combining heuristics and a machine learning model for release event identification, and physics-based 3D trajectory reconstruction in a calibrated World Coordinate System. The system's accuracy in determining release parameters (position, time, initial velocity, and angles) and reconstructing free-flight trajectories is evaluated against an OptiTrack motion capture system. The findings quantify the system's performance, highlighting its potential as an accessible tool for data-driven coaching, sports science research, and potentially for future investigations into complex motor control phenomena like functional solution manifolds in basketball shooting.

Additional Key Words and Phrases: Basketball, Computer Vision, Sports Science, Machine Learning

1 INTRODUCTION

Basketball is one of the most popular sports globally, where the ability to accurately shoot the ball under diverse conditions is critical for success.

A player's shooting technique and the specific shot context culminate in a set of release parameters: speed, angle and height. These parameters in conjunction with the laws of physics almost entirely determine the subsequent trajectory of the shot. Players typically operate within a preferred range of these parameters to achieve successful shots [2]. The concept of a 'solution manifold' refers to the specific combinations of release parameters that result in a successful shot. More advanced players demonstrate an ability to navigate this manifold efficiently, often finding regions where the shot has a high tolerance. In these high-tolerance regions, small, unavoidable variations in the release parameters will still lead to a successful shot, indicating robust control and effective exploitation of motor redundancy[6][11]. A deep understanding of these kinematics is essential for coaches and athletes seeking to optimize performance and refine technique.

Traditionally, detailed kinematic analysis relies on sophisticated multi-camera motion capture systems like OptiTrack or Vicon. While these systems provide high-fidelity data, their cost, complex setup requirements, limit their widespread adoption in regular training settings and field research. This creates a need for more accessible and practical tools that can provide valuable kinematic insights from readily available recording equipment. Monocular computer vision, utilizing footage from a single camera, presents a promising avenue for developing such analytical tools. Advances in object detection, human pose estimation, and trajectory tracking algorithms allow for the extraction of rich information from standard video recordings. However, inferring accurate 3D kinematics from 2D image sequences, particularly robust depth estimation, remains a significant challenge.

The main contribution of this work is an end-to-end system able to detect the release of a basketball, track the ball and reconstruct its trajectory to extract the release parameter and assess it against an OptiTrack 3D tracking system. This validation focuses on the monocular system's ability to accurately determine the release event (time and position), and the initial free-flight parameters (speed, elevation, and azimuth angles) of a basketball shot. The remainder of this paper is organized as follows: Section 2 outlines the problem statement and research questions. Section 3 covers the theoretical foundations. Section 4 details the system design. Section 5 describes the validation methodology. Section 6 presents the results, followed by a discussion in Section 7 and conclusions in Section 8.

2 PROBLEM STATEMENT

A quantitative understanding of basketball shooting mechanics is fundamental for optimizing player performance and informing coaches of better strategies to help their players. Key kinematic parameters at release such as ball speed, elevation angle, azimuth angle, and release height, dictate the success of a shot [2].

Despite this need, a significant gap exists in the availability of practical and validated tools for detailed kinematic analysis outside of specialized settings. High-end motion capture systems, while accurate, are often prohibitively expensive and complex for routine use by many coaches, teams, or researchers studying athletes in more natural environments. For example, Inaba et al studied the release parameters of collegiate player shots using 20 Vicon cameras [7]. Conversely, simpler video analysis methods may lack the necessary 3D accuracy or robust validation. Chakraborty and Meher [4] also researched shooting angles and velocities from a monocular system; however, their work primarily focused on robust trajectory tracking within the video frame, with their estimated parameters remaining pixel-based thereby limiting their direct real-world interpretation for kinematic analysis or coaching.

Therefore, the central problem addressed by this study is the development and rigorous validation of a monocular computer vision system capable of providing accurate 3D kinematic analysis of basketball shots. Establishing the accuracy and reliability of such a system against an industry-standard 3D tracking system is an important first step before it can be confidently deployed for research or practical applications.

TScIT 43, July 4, 2025, Enschede, The Netherlands

2.1 Research Question

This study investigates how effectively a monocular computer vision system, integrating object tracking with a release detection model and physics-based analysis, can determine the kinematics (release event, trajectory, and initial parameters) of a basketball when compared to an OptiTrack system. To address this comprehensively, the study will investigate the following sub-questions:

- How accurately can a monocular computer vision system, using a hybrid heuristic and machine learning model, identify the 3D position and time of a basketball's release compared to an OptiTrack system?
- What is the accuracy of the system's reconstructed 3D free-flight trajectory, as measured by Root Mean Squared Error (RMSE) against ground truth data?
- How accurate are the initial release parameters (speed, elevation, and azimuth angle) derived by the monocular system when evaluated against the ground truth data?

3 THEORETICAL FOUNDATIONS

The analysis of basketball shot kinematics from monocular video requires a robust understanding of several key computer vision, machine learning, and physics principles. This section outlines the theoretical foundations including camera geometry, coordinate system transformations, motion modeling, and the machine learning techniques used for object tracking and event identification, which have been used to extract release parameters from video footage.

3.1 Camera Geometry and Calibration

To extract precise real-world measurements from a 2D video, it is crucial to understand how a camera projects a 3D scene onto a 2D image. The pinhole camera model is the fundamental geometric description used for this. It simplifies the camera to a single point (the optical center) where light rays converge, forming an inverted image on an image plane [15].

A 3D point in the camera's own coordinate system, denoted as $P_c = [X_c, Y_c, Z_c]^T$, is projected onto the 2D image plane at pixel coordinates (u, v). This projection is governed by the camera's intrinsic parameters, which include its focal lengths (f_x, f_y) in pixel units and the coordinates of its optical center (c_x, c_y):

$$u = f_x \frac{X_c}{Z_c} + c_x$$

$$v = f_y \frac{Y_c}{Z_c} + c_y$$

These intrinsic parameters are grouped into a matrix K:

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where s is the skew coefficient, which is typically zero [15]. Real-world lenses introduce distortions (e.g., radial, tangential, or fisheye for wide-angle lenses like the GoPro used in this study), which warp the image.

Camera calibration is the process of precisely determining the intrinsic parameters of (K) and distortion coefficients. This allows for image undistortion or direct undistortion of detected 2D points, ensuring adherence to the pinhole model for subsequent 3D calculations.

3.2 Coordinate Systems and Transformations

To analyze motion in a real-world context, the 3D data extracted from the camera's perspective must be transformed into a consistent World Coordinate System (WCS). This is a fixed 3D Cartesian system that is defined to represent the physical space (e.g., with the origin at a corner of the free throw box, and axes aligned with the court).

The transformation between P_c and P_w involves a rotation matrix R (camera orientation relative to WCS) and a translation vector t (camera position relative to WCS origin):

$$P_c = R \cdot P_w + t$$

Alternatively, to transform a point from the camera coordinate system to WCS: $P_w = R^T (P_c - t)$.

These extrinsic parameters (R, t) are determined by matching $N \geq 3$ known 3D WCS points to their corresponding 2D image projections using techniques like iterative optimization algorithms such as those implemented in OpenCV's solvePnP function [3]. Establishing this WCS allows us to interpret motion in metric units and apply that to physics models.

3.3 Trajectory Modeling and Parameter Estimation

Once a set of 3D positions of the basketball over time over time has been obtained, the initial release parameters (velocity and angle) can be estimated by fitting a physics-based motion model to the trajectory points.

3.3.1 Projectile Motion with Air Drag. While a simplified model only relying on gravity provides a useful approximation for many scenarios, particularly over short flight durations or at lower speeds, a more realistic analysis must incorporate the effects of air drag. The air drag (F_d) exerted on the ball is typically modeled as being proportional to the square of the ball's speed (v) and acts in the direction opposite to its movement. Its magnitude can be expressed as:

$$F_d = \frac{1}{2} C_d \rho A v^2$$

where C_d is a dimensionless drag coefficient (which depends on the object's shape and surface characteristics), ρ is the air density, and A is the cross-sectional area of the ball. This force leads to an acceleration due to drag, $a_d = F_d/m$, where m is the mass of the ball. For a basketball, the following typical parameters apply [12]:

- Mass (m): 0.6 kg
- Radius (r): 0.12 m
- Drag coefficient (C_d): 0.54
- Air density at 20 °C and 101.325 kPa (ρ): 1.204 kg/m^3

Resolving this drag acceleration into components and combining it with gravitational acceleration results in a system of coupled second-order ordinary differential equations (ODEs) describing the ball's acceleration in each dimension. For a 3D trajectory $(X(t), Y(t), Z(t))$ with velocity components $(\dot{X}, \dot{Y}, \dot{Z})$ and speed $v = \sqrt{\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2}$, the equations of motion become:

$$\ddot{X}(t) = -\frac{C_d \rho A}{2m} \dot{X} v + g_x = -k_b \dot{X} v + g_x$$
$$\ddot{Y}(t) = -\frac{C_d \rho A}{2m} \dot{Y} v + g_y = -k_b \dot{Y} v + g_y$$
$$\ddot{Z}(t) = -\frac{C_d \rho A}{2m} \dot{Z} v + g_z = -k_b \dot{Z} v + g_z$$

where $k_b = (C_d \rho A)/(2m)$ is often referred to as the drag parameter, and (g_x, g_y, g_z) are the components of gravitational acceleration along the respective axes ($g_x = 0, g_y = -9.81, g_z = 0$).

Due to the air resistance term depending on v and the individual velocity components $\dot{X}, \dot{Y}, \dot{Z}$), these differential equations generally lack simple solutions for position over time. Consequently, determining the trajectory for a given set of initial conditions (initial position (X_0, Y_0, Z_0) and initial velocity (V_{x0}, V_{y0}, V_{z0})) requires numerical integration using an ODE solver [16].

4 SYSTEM DESIGN

The software system consists of a multi-stage pipeline that processes video footage of a single GoPro to extract 3D basketball trajectories and estimate release parameters. The pipeline can be broadly categorized into five main stages: (1) Camera Calibration and World Coordinate System Setup, (2) Object Detection and Pose Estimation, (3) Release Event Identification, (4) 3D Trajectory Reconstruction in WCS, and (5) Physics-Based Model Fitting.

4.1 Data Collection Setup

A GoPro HERO7 Black action camera was selected for its ability to record high-resolution video at a high frame rate, specifically 1080p at 119.88 (120) frames per second. This helps to ensure that the object detection model does not fail to recognize the ball due to motion blur. The camera was configured to use its "Linear" Field of View (FOV) setting. This mode applies in-camera correction to mitigate the inherent fisheye distortion caused by the wide-angle lens. While "Linear" mode significantly reduces distortion [1], a subsequent software based calibration was employed using the GyroFlow application [2] to correct for any residual lens distortion and to determine intrinsic camera parameters. For the analysis, the camera is positioned on tripod perpendicularly to the ball's trajectory, providing a side-view perspective. The camera is placed at a distance from the shooting position such that the player and the entire flight of the ball to the basket can be captured. See Figure 1 for an example of a shot in an experiment setup.

4.2 Pipeline Stage 1: Calibration and Coordinate System Setup

Before doing any calculations, the system first establishes a WCS by using measured court markings and their associated pixel coordinates in the frame. The system uses OpenCV's solvePnP function with the EPNP [3] algorithm to estimate the camera's extrinsic parameters: the rotation matrix $R_{wcs\_to\_ccs}$ and translation vector $t_{wcs\_to\_ccs}$. These parameters define the camera's precise pose relative to the court's WCS and are crucial for subsequent 3D transformations.

4.3 Pipeline Stage 2: Object Detection and Pose Estimation

4.3.1 Image Pre-processing. Detection is performed on each video frame. While the GoPro's "Linear" mode corrects for most distortion, a pre-computed calibration map is used to ensure maximum accuracy (derived from the full fisheye calibration detailed in Section 3). The full-frame undistortion via cv2.remap is applied (using the map from Section 3). Detections occur on this undistorted image.

4.3.2 Object Detection and Pose Estimation. An Ultralytics YOLOv8n object detection model [8] was trained on a dataset purely focusing on the detection of basketballs [5]. This specific model was chosen for its balance of high inference speed and low computational cost [3] , which was crucial for processing the extensive video data locally. The model outputs a bounding box for the detected ball. The center of this bounding box is taken as the initial 2D pixel location of the ball. A separate YOLOv8n pose estimation model is applied to detect human figures and locate their key body points according to the format of the COCO dataset [10]. From the detected poses, the system identifies the player most likely involved in the shot (typically the one closest to the ball and exhibiting shooting posture). The 2D pixel coordinates of the shooting hand's wrist (COCO keypoint index 10) are extracted. To reduce pixel-level noise from the detections and missed detection, a Kalman Filter [13] is applied to the 2D undistorted coordinates of the ball center, producing a smoother 2D trajectory over time as well as predicting the ball location if a detection is missed. If detections are missed, the Kalman Filter's prediction is used for up to 5 consecutive frames to maintain a smooth trajectory before the point is considered lost for trajectory reconstruction purposes.

4.4 Pipeline Stage 3: Release Event Identification

Projectile motion describes the ball's flight once it is no longer propelled by the shooter. For this system, ball release is defined as the frame where the shooter's hand and the basketball clearly separate, marking the transition to free flight under external forces.

4.4.1 Heuristic Trigger. The system first identifies a candidate release window by analyzing the spatial relationship between the shooter's wrist position and the ball center. A potential release event is flagged when the wrist, previously inside or near the ball's bounding box, is consistently detected outside a predefined proximity margin for 2 consecutive frames. The first frame of this separation sequence is marked as a candidate release. This consistency check tries to mitigate false positives due to momentary detection jitter or rapid, non-release hand movements. Once release is detected, the system records the candidate release frame number and the ball's 2D pixel center at that frame. This frame serves as an anchor for more precise refinement.

4.4.2 Model-Based Refinement. This candidate heuristic frame triggers a more sophisticated analysis using a convolutional neural network (CNN), termed ReleaseRelationNet. This approach is based

[1] https://community.gopro.com/s/article/What-is-Linear-Field-Of-View-FOV?
[2] https://docs.gyroflow.xyz/app
[3] https://docs.ultralytics.com/compare/yolo11-vs-yolov8/#performance-head-to-head-yolo11-vs-yolov8"

on Two Stream Action Recognition for CNNS [14]. This model refines an initial heuristic guess, outputting a score indicating the likelihood of release between consecutive frames. The core idea behind ReleaseRelationNet is to learn the subtle visual cues that differentiate a ball being controlled by the hand versus a ball that has just been released and is in free flight, by examining relation of the hand and the ball in time and space.

For a window of 10 frames before to 5 frames after the heuristic point, the system extracts localized 96×96 image patches . Specifically, for each frame t and its preceding frame $t-1$ in this window, two types of patches are cropped: one centered on the detected wrist position of the shooting hand and another centered on the detected ball position. These patches of serve as input to the network with the following format: ($hand_t$, $hand_{t-1}$, $ball_t$, $ball_{t-1}$). This set allows the network to capture motion and changes in appearance.

ReleaseRelationNet employs a Siamese-like architecture where separate but potentially shared-weight CNN branches process the hand patches and ball patches respectively to extract feature representations. Each feature extraction branch consists of a sequence of convolutional layers, ReLU activations, max-pooling, and batch normalization layers, and come together in an average pooling layer and a fully connected layer to produce a feature vector of size 128. The features extracted from the $hand_t, hand_{t-1}$ sequence and the $ball_t, ball_{t-1}$ sequence are then concatenated.

These combined hand and ball features are fed into a relation module, which is a series of fully connected layers. This module is tasked with learning the relationship between the hand's state and the ball's state across the two consecutive frames. The network outputs a single "release score" between 0 (strong evidence of hand-on-ball) and 1 (strong evidence of release) for each pair of consecutive frames $(t-1, t)$ analyzed.

The final, refined release frame is determined by finding the frame t corresponding to the largest positive change in this release score (i.e., $\text{score}_t - \text{score}_{t-1}$). This sharp increase signifies the most probable transition from a "preparation" state to a "released" state, attempting to pinpoint the moment the ball leaves the hand. See Figure 6 for an example release frame.

4.5 2D Trajectory Extraction and Conversion to Camera Coordinate System (CCS)

Once the ball's release is detected, the system tracks the ball's undistorted 2D pixel center (u, v) in subsequent video frames as seen in Figure 7, forming its initial 2D trajectory. To convert these 2D pixel points into 3D coordinates within the camera's own coordinate system (CCS), each undistorted point (u, v) is transformed into 3D space, resulting in $P_c = [X_c, Y_c, Z_c]^T$. This unprojection requires the camera's intrinsic parameters (f_x, f_y, c_x, c_y from the intrinsic matrix K) and an estimate of the ball's depth (Z_c) from the camera.

4.5.1 Depth Estimation at Release. The depth of the ball at the moment of release ($Z_{c,\text{release}}$) is estimated using the pinhole camera principle, using the ball's known physical diameter and its apparent size in pixels based on the size of the bounding box:

$$Z_{c,\text{release}} = \frac{\text{d}_{\text{real}} \cdot f_{x,\text{avg}}}{\text{d}_{\text{pixels}}}$$

Here, d_{real} is the known physical diameter of the ball, f_{avg} is the average focal length , and d_{pixels} is the ball's diameter in pixels derived from its bounding box at the moment of release. However, due to potential noise in apparent pixel diameter and effect on the final outputs, a fixed depth value of three meters (measured distance of camera position to a general release position) was used as $Z_{c,\text{release}}$ in this implementation.

4.5.2 Conversion to 3D CCS Coordinates. With the estimated depth, the normalized image coordinates are calculated:

$$x_n = \frac{u - c_x}{f_x}$$
$$y_n = \frac{v - c_y}{f_y}$$

For the ball's trajectory, it is assumed that the ball's depth (Z_c) remains approximately constant and equal to $Z_{c,\text{release}}$. Therefore, the 3D CCS coordinates for each tracked point are:

$$X_c = x_n \cdot Z_{c,\text{release}}$$
$$Y_c = y_n \cdot Z_{c,\text{release}}$$
$$Z_c = Z_{c,\text{release}}$$

The Y_c coordinate is typically inverted (most cameras have a positive Y downwards) to align with conventional physics coordinate systems where 'up' is positive.

4.6 Transformation to World Coordinate System (WCS)

The 3D trajectory points obtained in the Camera Coordinate System ($P_c = [X_c, Y_c, Z_c]^T$) are then transformed into the World Coordinate System (P_w). This transformation uses the inverse of the extrinsic parameters (rotation $R_{wcs\_to\_ccs}$ and translation $t_{wcs\_to\_ccs}$):

$$P_w = R_{wcs\_to\_ccs}^T (P_c - t_{wcs\_to\_ccs})$$

This WCS trajectory forms the input for the final physics-based model fitting.

4.7 Trajectory Model Fitting and Release Parameter Estimation

The sequence of 3D WCS (P_w trajectory points is used to estimate the ball's initial release velocity and angle by fitting a physics-based motion model.

4.7.1 Physics Model (ODEs with Air Drag). The proposed 3D projectile motion model incorporates both gravity and air drag. This model is defined by the following system of ordinary differential equations (ODEs) describing the ball's acceleration in each dimension:

$$\ddot{X}(t) = -k_b \cdot \dot{X} \cdot v + g_{cx}$$
$$\ddot{Y}(t) = -k_b \cdot \dot{Y} \cdot v + g_{cy}$$
$$\ddot{Z}(t) = -k_b \cdot \dot{Z} \cdot v + g_{cz}$$

Here, v is the ball's speed, k_b is the drag parameter (defined as $(C_d \cdot \rho \cdot A)/(2 \cdot m)$, where m is the ball's mass), and (g_{cx}, g_{cy}, g_{cz}) are the components of gravitational acceleration in the Camera Coordinate System.

4.7.2 Numerical Integration and Optimization. Numerical integration [1] is used to simulate the ball's trajectory for a given set of initial conditions (an initial position $P_{0,\text{wcs}}$ and an initial velocity $V_{0,\text{wcs}}$).

An optimization process is used to estimate the actual release parameters. An objective function was defined to quantify the sum of squared errors between the simulated trajectory and the observed 3D trajectory points. An optimization algorithm then finds the initial velocity vector $V_{0,\text{wcs}} = [V_{cx0}, V_{cy0}, V_{cz0}]^T$ that minimizes this sum of squared errors. The initial position $P_{0,\text{wcs}}$ is taken as the first point in the observed trajectory segment.

4.7.3 Calculation of Release Parameters. From the optimized initial velocity vector $V_{0,\text{wcs}}$, the key release parameters are calculated:

- **Release Speed:** The magnitude of the fitted initial velocity vector:

$$v_{\text{release}} = \|V_{0,\text{wcs}}\| = \sqrt{V_{x0}^2 + V_{y0}^2 + V_{z0}^2}$$

- **Release Elevation Angle:** The angle of $V_{0,\text{wcs}}$ with respect to the horizontal XZ-plane:

$$\angle\text{Elevation}_{\text{wcs}} = \text{atan2}(V_{y0}, \sqrt{V_{x0}^2 + V_{z0}^2})$$

- **Release Azimuthal Angle:** The angle of $V_{0,\text{wcs}}$ projected onto the horizontal XZ-plane, relative to the camera's X-axis:

$$\angle\text{Azimuthal}_{\text{wcs}} = \text{atan2}(V_{z0}, V_{x0})$$

5 VALIDATION METHODOLOGY

To validate the accuracy of the monocular computer vision system presented in this paper, an experiment was conducted to compare its performance against ground truth data obtained from an OptiTrack system within a controlled indoor environment.

5.1 Experimental Setup

5.1.1 Ground Truth System. An OptiTrack motion capture system (Model: PrimeX 13, Number of cameras: 12) served as the ground truth. The system was configured to track reflective markers at a frequency of 240 Hz. For ball tracking, a Wilson NBA DRV basketball was equipped with 10 reflective markers arranged unevenly to define a rigid body enabling the tracking of the centroid of the ball.

5.1.2 System Under Validation. The GoPro camera and software pipeline described in Section 4.

5.1.3 Environment and World Coordinate System (WCS). The experiment was conducted in a laboratory space with a relatively neutral background at the University of Twente. A specific shooting area was demarcated with tape, approximately 3.3m long by 2m wide, with a table serving as a makeshift backboard. Figure 1 shows the demarcated area. The corners of the demarcated area and at the top of the table contained OptiTrack markers to accurately retrieve their WCS coordinates.

The experiment used a WCS where the Y-axis points vertically upwards, the X-axis points along the primary direction of the shot towards the backboard, and the Z-axis points forward away from the camera.

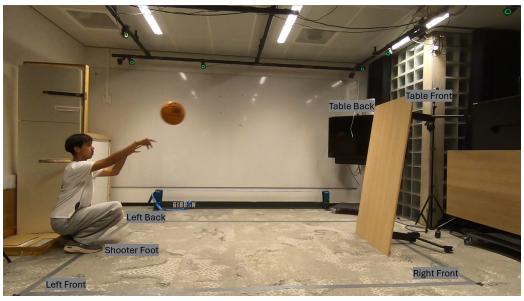


Fig. 1. The testing environment with the demarcated area. Includes labels of used world points.

- **Monocular System WCS**: Established using OpenCV's solvePNP function, as detailed in Section 4. This involved manually identifying the 2D pixel locations in a reference GoPro image of the known 3D points within the delineated shot lane and table backboard. These 3D world coordinates were retrieved by creating rigid bodies in Motive and extracting the respective marker locations. See Appendix C for the known points extracted.
- **OptiTrack System WCS**: The OptiTrack system was calibrated to its own native WCS. To ensure direct comparability with the monocular system's data, which uses a different axis convention, the 3D data exported from OptiTrack's Motive software was transformed to match this convention. This ensured all 3D data from both systems were expressed in an identical coordinate system.

5.2 Data Acquisition Protocol

5.2.1 Synchronization. To align the data streams from the GoPro (119.88 fps) and OptiTrack (240 Hz) systems, a synchronization event needed to be performed for each batch of recordings. For each recording batch, a basketball was dropped, and the first frame of its impact with the ground served as the synchronization event. For the monocular system, the frame depicting the first ground impact was visually identified. For the OptiTrack, the timestamp corresponding to the lowest vertical position Y_{wcs} of the tracked ball markers during the impact event was identified.

5.2.2 Throwing Task. Approximately 10 to 12 basketball shots per batch (totaling around 100 shots across multiple batches) were performed. Shots were taken from a seated position (due to laboratory ceiling height limitations) from a consistent starting area close to (0,0,1) in world coordinates within the calibrated capture volume, aiming towards the table backboard.

5.3 Ground Truth Data Processing: Release Event and Parameters

The continuous 3D WCS trajectory of the basketball from the OptiTrack system was processed to determine the ground truth (GT) release event and initial parameters for each valid shot.

Velocity and acceleration components in the WCS were then computed by taking numerical derivatives of the smoothed position data with respect to time (dt = 1/240 s). Direct calculation of

acceleration from raw position data resulted in excessively noisy profiles. In order to prevent false positives due to noisy data, the system searches for the first frame after an initial upward acceleration (indicative of a launch) where the vertical acceleration ($\ddot{Y}_w$) of the ball stabilizes around the gravitational acceleration ($-9.81\,\mathrm{m/s^2}$) within a defined tolerance ($\pm 1.0\,\mathrm{m/s^2}$) for a minimum duration (0.05 seconds). This signifies the transition to free flight. Similarly, using minimal smoothing like a Savitzky-Golay filter with a shorter window often failed to produce an acceleration signal stable enough for our kinematic heuristic to reliably identify the onset of free-fall, as illustrated for a representative case in Appendix A, Figure 3. Therefore, to obtain a more stable estimate of acceleration, the raw OptiTrack ball position data (X_w, Y_w, Z_w) was smoothed using a Savitzky-Golay filter (window length 25, polynomial order 2). See Appendix A, Figure 2 for the same case with these respective parameters.

The 3D position of the ball at T_{GT} was recorded as P_{GT}. The 3D WCS trajectory segment from P_{GT} until the ball's impact with the table was extracted. The extracted GT trajectory segment was then fitted with the same physics-based projectile motion model (ODEs with air drag in WCS, as described in Section 3) used by the monocular system. This fitting process yielded the ground truth initial release parameters: $P_{0,GT}$ (which should be very close to P_{GT}), speed ($v_{0,GT}$), WCS elevation angle ($\theta_{0,GT}$), and WCS azimuth angle ($\phi_{0,GT}$).

5.4 Data Matching and Comparison

For each shot successfully processed by the monocular system, its detected release frame was converted to an equivalent OptiTrack time using the synchronization offset. The closest valid GT release event (identified as per the previous section) within a tolerance window (±0.5 seconds) was matched to the monocular detection. Shots were excluded if a clear ground truth release instance could not be reliably determined. Despite data smoothing, some OptiTrack recordings exhibited noisy acceleration patterns, making the automated kinematic heuristic for identifying the $-9.81\,\mathrm{m/s^2}$ free-fall transition inconclusive, even after manual review.

5.5 Validation Metrics

For each matched shot, the following error metrics were computed, comparing the monocular system's outputs to the ground truth:

5.5.1 *Release Event Accuracy.*

- **Temporal Error**: The absolute time difference between the monocular system's detected release time (T_{mono}) and the ground truth release time (T_{GT}): $|T_{mono} - T_{GT}|$.
- **Spatial Error (3D Position)**: The 3D Euclidean distance between the monocular system's estimated release position ($P_{0,mono}$ from its WCS fit) and the ground truth release position ($P_{0,GT}$ from the GT WCS fit): $||P_{0,mono} - P_{0,GT}||_2$.

5.5.2 Trajectory Accuracy. To evaluate the system's spatial reconstruction and parameter fitting capabilities independent of the timing error, all subsequent comparisons were performed on trajectory data that was first temporally aligned to a common starting point. The ground truth release time, T_{GT}, was chosen as the definitive start time for both the OptiTrack and the monocular trajectory segments in this analysis.

- **Initial Spatial Offset**: The 3D Euclidean distance between the first point of the aligned monocular trajectory and the first point of the ground truth trajectory ($P_{0,GT}$): $||P_{mono}(t = T_{GT}) - P_{0,GT}||_2$.
- **Root Mean Squared Error (RMSE)**: The RMSE between the 3D WCS coordinates of the monocular system's reconstructed trajectory and the time-interpolated ground truth trajectory over their common flight path duration. Both trajectories were considered from the common start time T_{GT} up to an X_{WCS} coordinate of 2.8m (or their natural termination if earlier), with the RMSE calculated over their overlapping duration.
- **Maximum Deviation**: The largest 3D Euclidean distance between the two trajectories at any commonly observed time point.
- **Percentage of Relevant GT Trajectory Tracked (%)**: The duration of the common flight segment (where both systems have valid data within the aligned analysis window) divided by the total duration of the ground truth trajectory within that same window (from T_{GT} up to $X_{WCS} = 2.8$m)

5.5.3 Release Parameter Accuracy. The absolute errors in release speed, elevation angle and azimuth angle between the monocular system's fitted WCS release parameters and the ground truth WCS release parameters based on the aligned trajectories at T_{GT}.

6 RESULTS

From approximately 100 recorded shots, a final set of 67 matched shots was included in the analysis. The remaining shots were primarily excluded due to inherent limitations in the raw OptiTrack data. Despite standard smoothing procedures (detailed in Section 5.3), these recordings often contained excessive noise or atypical motion patterns that rendered the trajectories unsuitable for robust analysis. Consequently, the kinematic heuristic was frequently unable to reliably identify a stable free-fall transition.

6.1 Tracking Performance

The accuracy of the monocular system's independent release timing detection, prior to temporal alignment, is summarized in Table 1. The mean absolute temporal error between system's detection and T_{GT} was 0.0270 ± 0.0164 seconds. The errors ranged from a minimum of 0 seconds (full agreement ground truth and monocular release) to a maximum of 0.0667 seconds. This corresponds to an average of approximately 3.24 frames at the system's operating frame rate of 119.88 fps. The initial 3D spatial error for the release position ($P_{0,mono}$), was 0.420 ± 0.141 meters. Here the errors ranged from 0.119 to 0.734 meters.

For the subsequent analysis, both trajectories were aligned to start at T_{GT}. At this common start time, the mean initial spatial offset between the monocular and ground truth ball positions was 0.383 ± 0.124 meters. The mean RMSE of the 3D trajectory comparison was 0.239 ± 0.075 meters. The analysis of trajectory completeness showed that the monocular system successfully tracked, on average, 94.7 ± 11.3 percent of the relevant ground truth flight path (up to

Table 1. Summary of Monocular System Performance Metrics against OptiTrack Ground Truth (N=67 Shots).

Performance Metric	Mean	Std. Dev.	Median	IQR
Release Event Detection Accuracy (System's Independent Detection)				
Temporal Error (s)	0.0270	0.0164	0.0250	0.0209
Initial Release Position Error (m)	0.4202	0.1407	0.4052	0.1894
3D WCS Trajectory Reconstruction Accuracy (Aligned at T_{GT})				
Initial Spatial Offset (m)	0.3831	0.1242	0.3725	0.1828
3D RMSE (m)	0.2392	0.0751	0.2361	0.0750
Maximum Deviation (m)	0.3863	0.1498	0.3830	0.1534
Percentage of GT Trajectory Tracked (%)	94.75	11.28	100.00	3.55
Release Parameter Accuracy (Fit on Aligned Trajectory)				
Speed MAE (m/s)	0.5710	0.2233	0.6047	0.2938
Elevation Angle MAE (deg)	13.6223	4.4362	13.8966	5.7872
Azimuth Angle MAE (deg)	8.8472	3.2349	9.0655	4.1196

X_{WCS} = 2.8m), with a median completeness of 100%. However, for some shots, the tracked percentage was lower, with a minimum of 38.5 percent observed. The mean RMSE of the 3D trajectory comparison was 0.239 ± 0.0751 meters with a range of 0.105 to 0.516 meters. The mean maximum deviation observed along any 3D trajectories was 0.386 ± 0.150 meters and ranged from 0.159 to 1.17 meters.

6.2 Release Parameter Estimation

Using the captured trajectory data, the release parameters were calculated with the fitting model which also takes drag into account. On average, the ground truth release speed was 4.12 ± 0.182 m/s, while the monocular system estimated 4.69 ± 0.25 m/s. For elevation, the ground truth was 48.3 ± 9.40 degrees versus the monocular system's fitted estimate of 34.7 ± 8.95 degrees. The ground truth azimuth averaged 1.45 ± 2.70 degrees, compared to the monocular system's average of −7.40 ± 1.85 degrees. This leads to the following absolute error (MAE) of 0.571 ± 0.223 m/s for release speed, 13.62 ± 4.44 degrees for elevation angle, and 8.85 ± 3.23 degrees for azimuth angle.

To assess the agreement and systematic bias between the monocular system and the ground truth for estimated release parameters, Bland-Altman plots were generated for speed, elevation, and azimuth (Figure 5).

The analysis shows a bias of +0.57 m/s for release speed (Figure 5a), indicating a systematic overestimation by the monocular system. The 95% limits of agreement show a range of approximately 0.1 m/s to 1.01 m/s. The plot also shows a clear proportional bias, where the difference between the methods increases as the mean speed of the shot increases.

For release elevation angle (Figure 5b), a large negative bias was observed, with a mean difference of -13.62 degrees. This indicates a consistent underestimation by the monocular system. The 95% limits of agreement were wide, spanning from approximately -22.4 degrees to -4.99 degrees. The points are scattered around the mean difference with no apparent trend.

The release azimuth angle plot the shows a significant negative bias (Figure 5c), with a mean difference of -8.85 degrees. The 95% limits of agreement were calculated to be from -15.0 degrees to -2.55 degrees, indicating a consistent negative offset in the monocular system's measurements. Additionally a slight negative trend can be observed where the difference becomes more negative as the mean azimuthal angle of the two methods increases.

7 DISCUSSION

The results presented provide a quantitative assessment of the system's performance across various metrics when compared to an OptiTrack motion capture system. In a parameter such temporal error, the system got close enough to a point where a reasonable release frame was found that was in accordance to a ground truth release frame. Additionally for a monocular system that cannot directly depth, a fixed depth worked well enough to get sub-meter accuracy. Furthermore, release parameters being off by some margin may work well enough for research on how players can improve their shot kinematics.

7.1 Strengths

The developed monocular system demonstrates several key strength and promising capabilities that could allow it to be used for basketball shot performance analysis. Firstly, the temporal accuracy of release detection seems to be promising. With a mean absolute temporal error of 0.0270 ± 0.0164 seconds, the system's release detection component, can pinpoint the moment of release with a precision of 3.24 ± 1.97 frames at 119.88 fps when compared to the ground truth release frame. It needs to be said however, that the OptiTrack ground truth release frames were based on the heuristic that the ball is considered to be released when the vertical acceleration is ($-9.81\,\mathrm{m/s^2}$) with some defined tolerance and minimum duration. This reliance on signal smoothing, such as with the Savitzky-Golay filter, to identify the stable free-fall acceleration phase means that the exact moment of kinematic release might be pinpointed with a slight inherent delay.

Secondly, the system generally achieved high completeness in tracking the balls relevant portion of the ball's trajectory. On average, 94.77% of the ground truth flight path (up to an X_{WCS} of 2.8m) was successfully tracked, with a median completeness of 100%. This indicates that the object detection in a clean environment without much background objects (like in the experimental setup) that the object detection (YOLOv8) and Kalman filtering components are robust for significant portions of the shot. However, these results may vary in more challenging conditions such as an outdoor basketball court.

Furthermore, the integration of a physics-based model with air drag allows the system to derive release parameters in real world units from the reconstructed 3D WCS trajectory. While the accuracy of some of these parameters is subject to the quality of the actual trajectory (discussed below), the ability to at least estimate them from a single-camera footage provides an advantage of over simpler 2D analysis techniques concerning trajectory shape and pixel-based movement. The end-to-end pipeline, from raw video to estimated kinematic parameters, represents a practical and more accessible approach to detailed shot analysis.

7.2 Limitations and Challenges

The most significant challenge for the monocular system lies in achieving accurate 3D spatial localization within the World Coordinate System. This is most evident in the Initial Spatial Offset at the T_{GT}, which had a mean error of 0.383 meters, and the subsequent mean 3D RMSE for the trajectory of 0.239 meters. A positional error of this magnitude is substantial, potentially representing the difference between a successful shot and a complete miss. The fact that this spatial offset (0.383 m) is only marginally smaller than the spatial error calculated from the system's original fit, based on T_{mono} (0.420 m) may confirm the minimal impact of the small temporal error. Given a mean ground truth speed of 4 m/s, the 0.027 s temporal difference accounts for a positional shift of merely 0.108 meters. This contrast with the 0.383 meters initial spatial offset highlights that the error mostly stems from inherent inaccuracies in the monocular system's 3D reconstruction and World Coordinate System calibration, rather than timing differences.

Visual inspection of the aggregate trajectory as seen in Figure 4 clearly indicates that the discrepancy in the Z_{wcs} component is a major contributor to the errors found. Firstly, this can be attributed to the inherent challenge of monocular depth estimation, as a single light ray provides no direct information about an object's distance, meaning depth is always inferred rather than directly measured. The system's reliance on an initial depth estimate at release ($Z_{c,release}$) based on apparent ball size, followed by an assumption of (near) constant camera-ball depth for subsequent CCS points before WCS transformation, introduces a systematic simplification. While the manual override of $Z_{c,release}$ to 3m which was closer to the actual depth of camera compared to the initially calculated 2.5m for most shots provided stability, it does not reflect true slight depth changes the ball may have throughout its flight. In addition to that, the accuracy of the camera's extrinsic parameters, which define the WCS, is highly dependent of the number, distribution, and quality of the 2D pixel points and 3D real-world points. Even with manual refinement of the translation vector $t_{wcs\_to\_ccs}$) to reduce initial positional offsets, rotational misalignments still persist, contributing to the observed error in the estimated WCS azimuth angle. This also persists throughout the RMSE of the trajectory itself, as over time the distance in the Z_{wcs} component increases leading to a larger error, with the tail sections of flight paths likely accounting for the trajectory maximum deviations.

Furthermore, the challenges in 3D trajectory reconstruction directly propagate to the large, systematic errors in the release parameters, as clearly visualized by the Bland-Altman analysis (Figure 5). As mentioned before, the most significant source is the error in the extrinsic parameters (R, t) derived from the 'solvePnP' calibration. The large negative bias in the azimuth angle (-8.85 degrees) can be attributed to the rotational error around the vertical axis (yaw) in the system's World Coordinate System. Similarly, the large underestimation of the elevation angle (-13.62 degrees) may be a consequence of rotational errors in pitch and translational errors in the camera's perceived height. When the system operates within this flawed reference it calculates launch angles that are correct for that mistaken perspective, resulting in a large, consistent bias against the ground truth.

In turn, the overestimation of speed (mean bias: +0.57 m/s) is a compensatory effect of the angular errors. The physics model, tasked with fitting a curve to a trajectory that is perceived as both flatter and azimuthally displaced, must calculate a higher initial velocity to account for this longer and lower flight path. This demonstrates how the foundational errors from the WCS calibration persist through the entire analysis, corrupting all derived release parameters.

To enhance the system's accuracy and utility, future monocular work should prioritize using a greater number of well-distributed 3D ground truth points to ensure a more successful extrinsic calibration. Future iterations could also explore incorporating stereo vision if a two-camera setup is feasible, or investigate advanced monocular depth estimation techniques, like the Marigold method for monocular depth estimation using deep learning proposed by Ke et al [9]. Improving the robustness of the extrinsic camera calibration (WCS setup) is one the most critical points; this could involve using more calibration points with better 3D spatial distribution and exploring more ways to help the SolvePnP function find a more accurate $R_{wcs\_to\_ccs}$ and $t_{wcs\_to\_ccs}$) to represent the WCS better.

8 CONCLUSION

This research aimed to determine the accuracy of a novel monocular computer vision system for identifying basketball shot release parameters and trajectories. Validation against an OptiTrack system revealed that while the proposed system can identify release timing with high accuracy and track the majority of the relevant ball flight, its estimation of absolute 3D release position , full 3D trajectory, and derived kinematic parameters (speed, elevation, azimuth) is impacted by the inherent challenges of monocular 3D reconstruction, particularly depth estimation and WCS calibration. Despite these current limitations in absolute 3D accuracy, the developed system offers a significant step towards accessible basketball shot analysis, and with clear future goals to further improve its utility for coaches and researchers.

REFERENCES

[1] solve_ivp — SciPy v1.15.3 Manual.

[2] Bartlett, R. *Introduction to Sports Biomechanics: Analysing Human Movement Patterns*, 2 ed. Routledge, London, Oct. 2007.

[3] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[4] Chakraborty, B., and Meher, S. A trajectory-based ball detection and tracking system with applications to shooting angle and velocity estimation in basketball videos. In *2013 Annual IEEE India Conference (INDICON)* (Dec. 2013), pp. 1–6. ISSN: 2325-9418.

[5] Dataset, O. S. basketball-w2xcw_dataset, 2023.

[6] Davids, K., Button, C., and Bennett, S. *Dynamics of skill acquisition: a constraints-led approach*. Human Kinetics, Champaign, Ill., 2008.

[7] Inaba, Y., Hakamada, N., and Murata, M. Influence of Selection of Release Angle and Speed on Success Rates of Jump Shots in Basketball:. In *Proceedings of the 5th International Congress on Sport Sciences Research and Technology Support* (Funchal, Madeira, Portugal, 2017), SCITEPRESS - Science and Technology Publications, pp. 48–55.

[8] Jocher, G., Chaurasia, A., and Qiu, J. Ultralytics YOLOv8, 2023.

[9] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., and Schindler, K. Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation, Apr. 2024. arXiv:2312.02145 [cs].

[10] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft COCO: Common Objects in Context, Feb. 2015. arXiv:1405.0312 [cs].

[11] Müller, H., and Sternad, D. Motor learning: changes in the structure of variability in a redundant task. *Advances in Experimental Medicine and Biology 629* (2009), 439–456.

[12] Okubo, H., and Hubbard, M. Identification of basketball parameters for a simulation model. *Procedia Engineering 2* (June 2010), 3281–3286.

[13] Pei, Y., Biswas, S., Fussell, D. S., and Pingali, K. An Elementary Introduction to Kalman Filtering, June 2019. arXiv:1710.04055 [eess].

[14] Simonyan, K., and Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems* (2014), Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc.

[15] Szeliski, R. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer, London, 2011.

[16] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272.

A SUPPLEMENTARY FIGURES

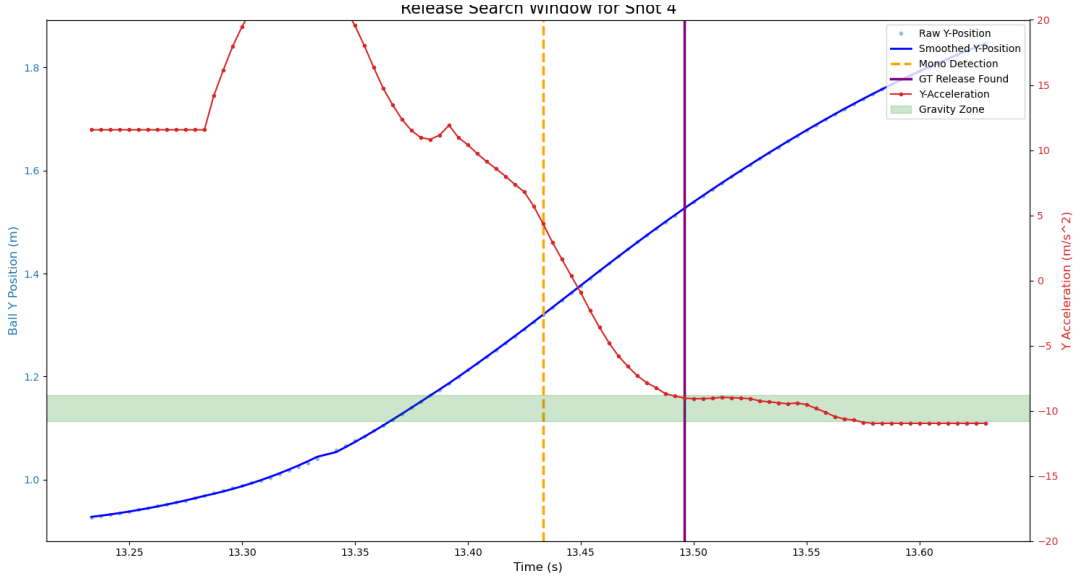


Fig. 2. Release search window for Batch 10 Shot 4, with Y-acceleration (red line) calculated using the system's Savitzky-Golay filter parameters (window length n=25, polynomial order 2) applied directly to position data for the second derivative. The increased smoothing reveals a clear stabilization within the gravity zone (green band), allowing for identification of the GT release point (purple vertical line).

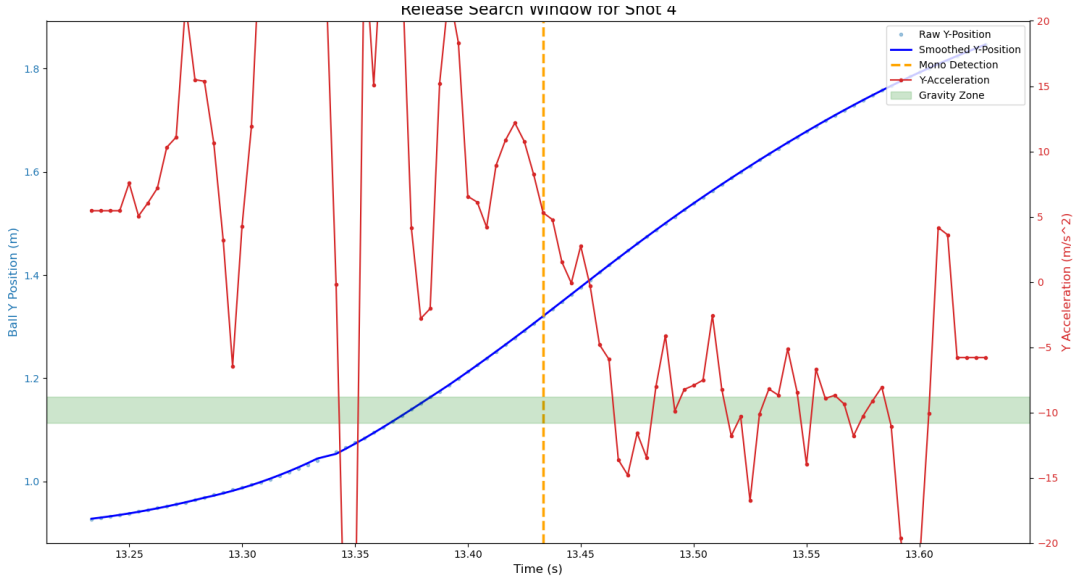


Fig. 3. Release search window for Batch 10 Shot 4, showing Y-acceleration (red line) calculated using a Savitzky-Golay filter with a short window length (n=7, polynomial order 2) applied directly to position data for the second derivative. The residual noise in the acceleration profile prevents the heuristic from identifying a stable free-fall period, hence no GT release point is found. The orange dashed line indicates the monocular system's detected release time for reference.

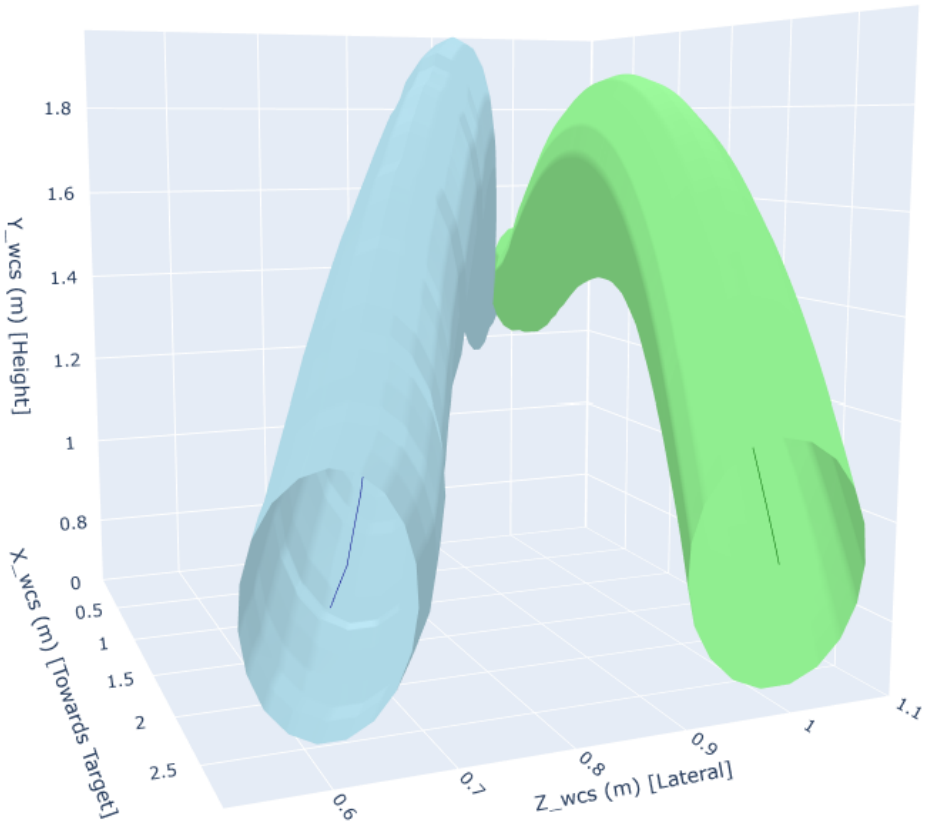
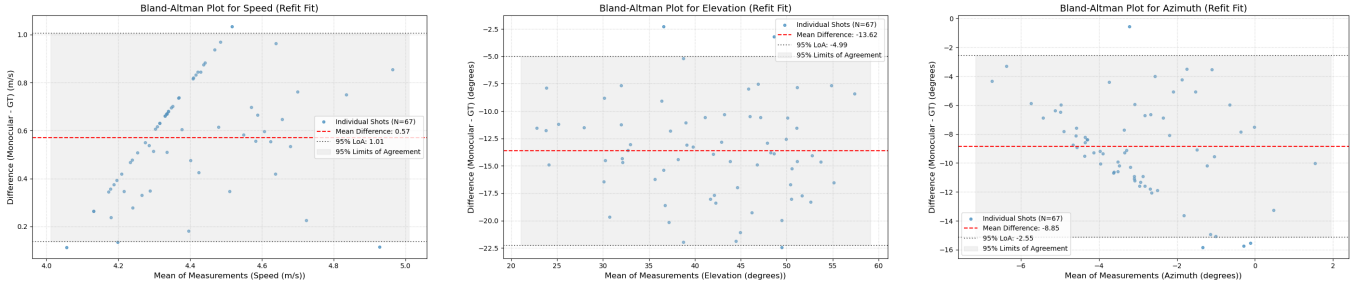


Fig. 4. This plot aggregates all analyzed shot trajectories, showcasing the agreement and variability between the monocular system's estimates (mean trajectory in dark blue, ±1 standard deviation spread in light blue) and the OptiTrack ground truth (mean trajectory in dark green, ±1 standard deviation spread in light green).



(a) Speed (m/s) (b) Elevation Angle (degrees) (c) Azimuth Angle (degrees)

Fig. 5. Bland-Altman plots illustrating the agreement between the monocular system and OptiTrack GT for estimated WCS release parameters: (a) Speed (m/s), (b) Elevation Angle (degrees), (c) Azimuth Angle (degrees). The dashed red line indicates the mean difference (bias), and dotted grey lines show the 95% limits of agreement.

B SYSTEM UI

Fig. 6. Example of a release frame. The red circle highlights the centroid of the ball at release.

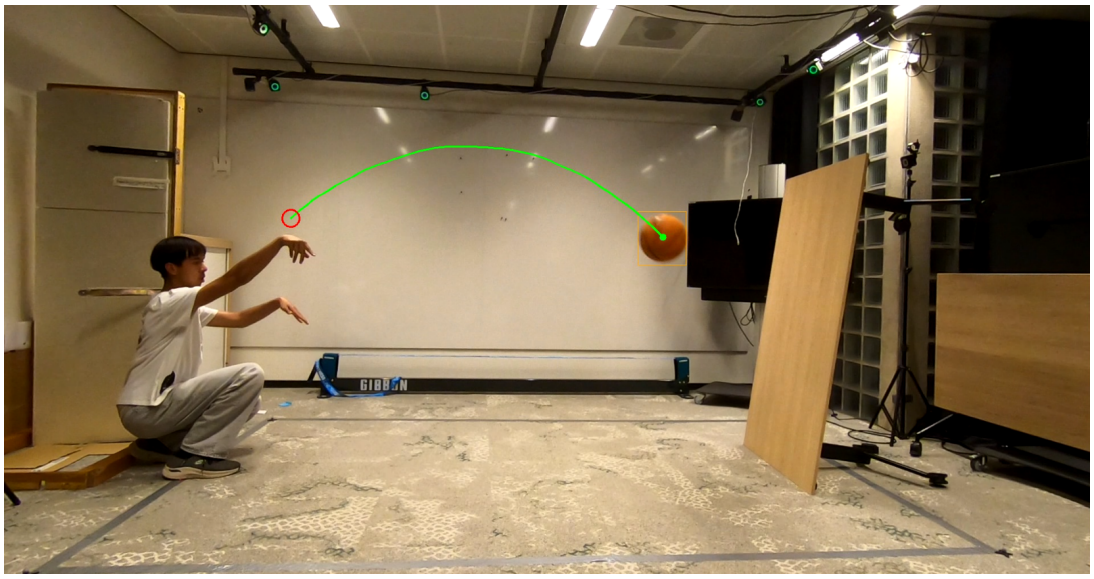


Fig. 7. Example of a tracked shot in the testing environment.

C APPENDIX: KNOWN POINTS

A list of the common world points used for calibration and reference. The last point, labeled "shooter foot" (approx. $[0, 0, 1]$), served as a crucial reference point. It was physically placed on the ground, specifically between the "left front" (P1) and "left back" (P2) markers. This fixed position provided a consistent starting reference when performing shots from the crouched stance.

```
COMMON_WORLD_POINTS_LIST = [
    [0.002955, 0.081709, 0.012093],  # P1: left front
    [0.050017, 0.030884, 1.955751],  # P2: left back
    [3.297116, 0, 0.046505],  # 35: right front
    [3.110943, 1.562576, 0.639417],  # P4: table front
    [3.140699, 1.546863, 1.433986],  # P5: table back
    [0, 0, 1]  # shooter foot (roughly in line with shot)
]
```

Fig. 8. Common world points (in meters) used for calibration and reference in the study.

D APPENDIX: USAGE OF AI

Gemini 2.5 Pro was used as an assistive productivity and analytical tool. It provided support for tasks including brainstorming, clarifying complex theoretical concepts like the basics of camera geometry, and challenging any logical arguments. The AI also served as a proofreader, and offered technical assistance for code debugging and LaTeX formatting. It is important to note that all final analyses, conclusions, and intellectual contributions remain the sole responsibility of the author.