POLITECNICO MILANO 1863

Trajectory and Spin Detection in a Bowling Throw

25.05.2025 | Davide Corradina, Michele Fassini

Contents

Problem Formulation

01

Problem Formulation

Objective:

Estimate the **2D trajectory** and **rotational spin** of a bowling ball using only a single, uncalibrated video.

Core challenges:

- **No markers** or multi-camera setups
- **Low-quality input**: low-resolution, noisy, inconsistent frame rates
- **Unstable camera**: freely positioned and possibly rotating
- **Difficult conditions**: minimal surface texture, yet needs accurate rotation estimation

Motivation and State of the art

Application fields

- Coaching and training feedback
- Performance analytics
- Equipment testing

Limitations of current solutions

- Rely on inertial **sensors**, **marker-based** motion capture, or **multi-camera** rigs
- Require **specialized hardware**, **precise calibration**, or **controlled environment**
- Are often **expensive**, **intrusive**, and **difficult to set up** in typical bowling alley conditions

Our contribution

- **End-to-End Pipeline**: A fully classical, integrated computer vision pipeline to jointly estimate 2D trajectory and rotational spin from a single video.
- **Automated Video Analysis Output**: Automatic generation of annotated output videos showing ball path, spin direction, and angular velocity over time.
- **Low-Cost & Accessible**: No need for high-speed cameras, IMUs, or multi-camera rigs—runs on standard consumer video.
- **Non-Intrusive & Markerless**: Requires no special markers, no sensor attachments, and works with unstructured, freely captured footage.
- **Real-World Deployability**: Robust to camera motion, low resolution, variable lighting, and frame rate inconsistencies—tested on actual lane-side recordings.

Classical Computer Vision Pipeline

The proposed pipeline is organized into a series of well-defined **modules**, each responsible for a key component of the analysis: lane detection, ball detection, lane reconstruction, trajectory reconstruction, and spin estimation.
The following sections provide a detailed description of each module and its role in the overall system.

- Lane Detection
- Ball Detection
- Reconstruction
- Ball Trajectory
- Spin Analysis

Solution Approach

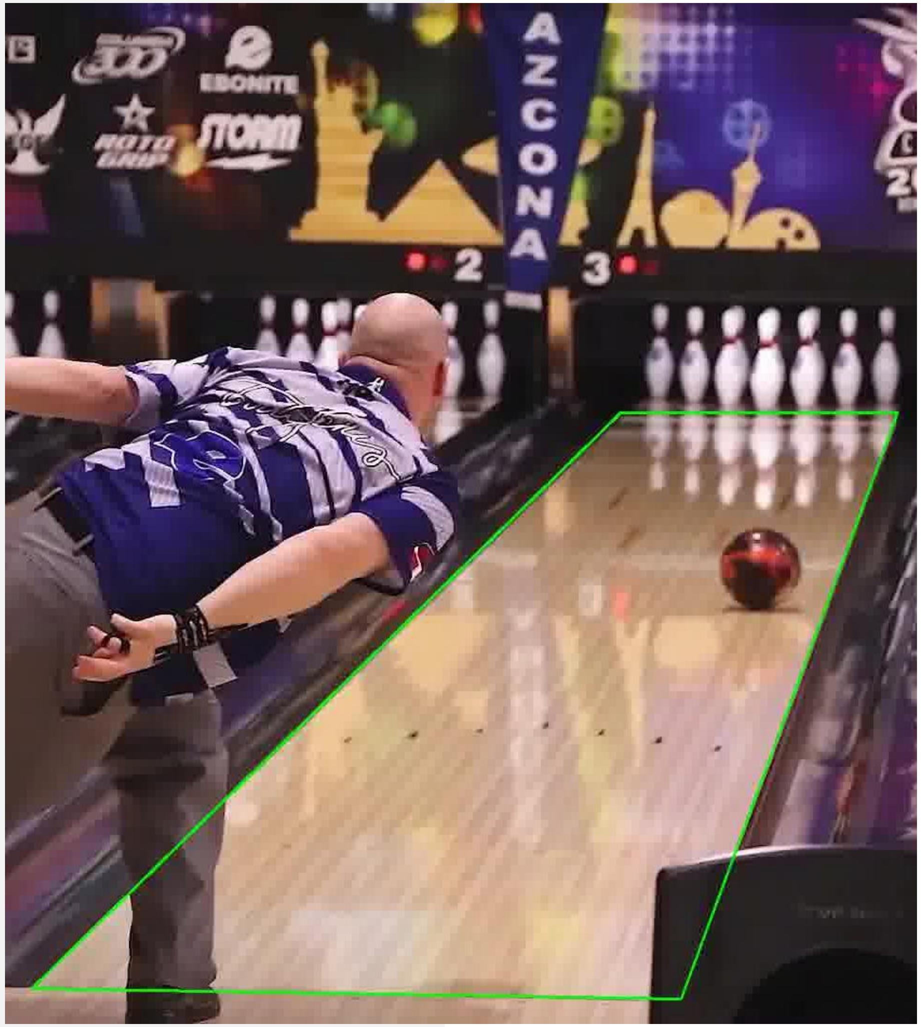
02


Brunswick
AZCONA
OPEN
2020
1
2
3

Lane Detection

02.1

Lane Detection



The goal of this section is to detect and track the lane across video frames

- Lane detection is applied to the original video
- Use Canny edge detection and Probabilistic Hough Transform
- Post-Processing of the detected lines to enhance the performance
- Initial estimate of the camera motion

Background Motion Estimation

Purpose

Estimate how much the camera move to handle two different cases: **still** and **moving cameras**

Procedure

- Feature tracking with **ORB** (Oriented FAST and Rotated BRIEF)
- Estimate homographies between frames
- Filter using **RANSAC** to isolate static background
- Extract translation components
- Make an average of the movements over the video to have an index that estimate the motion

Results

- If avg-motion is less than 1 the video is considered static
- Otherwise the video is considered moving

Bottom Line Detetion

Processing

- Focus Area: Lower quarter of each frame
- Binary masks for light brown and rose hues
- **Canny edge detection** (Otsu thresholding)
- Detect lines via **Probabilistic Hough Transform**
- Keep only horizontal segments
- Choose the **most marked line** per frame

Post-Processing

- Post-processing on the closest point of the line to the origin (identifies uniquely the line)
- **Outlier removal**: distance-based and statistical
- **Smoothing and interpolation** for dynamic scenes
- **Averaging** for static scenes

Lateral Lines Detetion

Processing

- Binary masks for light brown and rose hues
- **Canny edge detection** (Otsu thresholding)
- Detect lines via **Probabilistic Hough Transform**
- Discard lines: nearly horizontal, below the bottom line
- Classify lines as **left/right** relative to center
- For every frame select **closest** valid line **to center** on each side

Post-Processing

- **Slope filtering** to ensure consistency
- Post-processing on the closest point of the line to the origin
- Dynamic scenes: outlier removal (**DBSCAN**), smoothing and then interpolation
- Static scenes: outlier removal (distance based), then averaging

Upper Line Detetion

Initial Estimate

- Geometric setup: Triangle from bottom + lateral line
- Color filtering (brown & rose in HSV)
- Scan upwards for first row **>98%** black pixels
- Estimate **pin dimensions** for template matching using:
    - Real-world lane and pin dimensions
    - Estimated upper line length

Template Matching

- Match resized **pin template** to filtered image
- Get the lower point of the pin
- Final line: horizontal through matched point

Color filtering

Template matching

Final Post-Processing

Static video

- **Averaging** through the first half of the video

Moving video

- Extract vertical positions of top corners
- Remove outliers using sliding window
- Estimate missing corners using proportional displacement from bottom boundaries

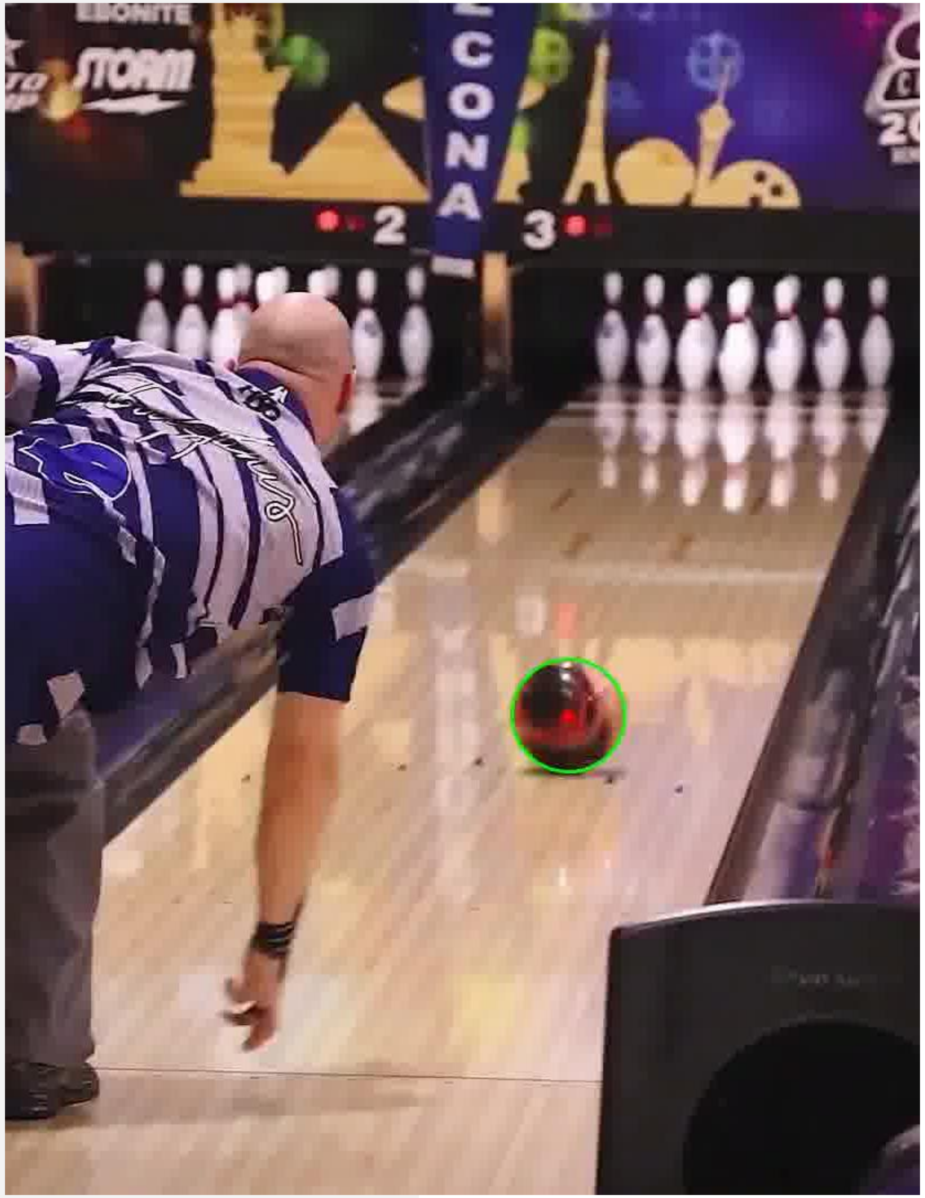
Occluded bottom line

- Maintain upper corners’ vertical positions
- Infer bottom corners from previous relative displacement


Brunswick
AZCONA
OPEN
2020
1
2
3

Ball Detection

02.2



Ball Detection

The goal of this section is to detect and track the center of the bowling ball across video frames

- Ball detection occurs after lane boundaries are identified
- Use **OpenCV HoughCircles** with lenient parameters to maximize recall
- Ensures **accuracy and robustness** using initialization, adaptive tracking, and post-processing

Processing

Initial Detection

- Restrict detection to the lane area
- Focus on lower part of the lane to avoid as much as possible noise
- Apply HoughCircles to detecte the ball in the first frames of the throw

Adaptive Search Window

- If the ball is detected in at least 5 consecutive frames and the centers are very close switch to focused search strategy
- Define square window centered on last center coordinates
- Apply HoughCircles again with tighter bounds

Reinitialization Logic

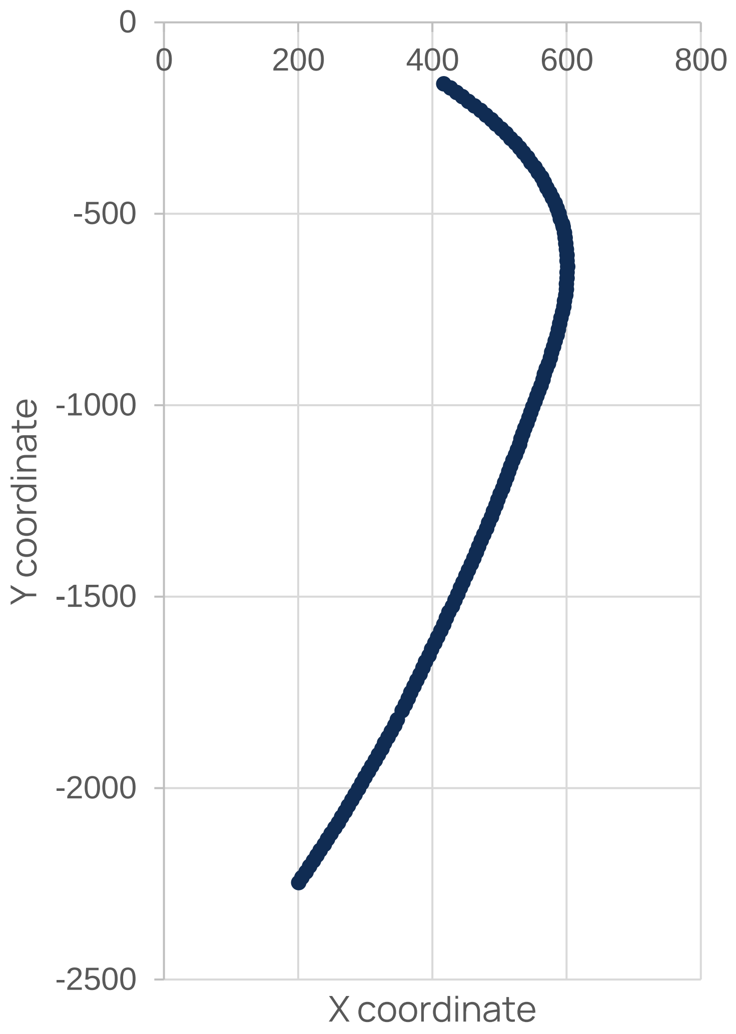
- If there is no ball detection for 2 frames, reset strategy: return to full-lane search
- Once re-detected for 5 frames, resume adaptive tracking

Post-Processing

Radius Outlier Removal
- Fit exponential curve to radius trend
- Remove outliers >10px from expected radius

Lane-Referenced Filtering
- Transform coordinates via lane homography
- Remove points before foul line (≤50 cm) and beyond back line
- End the detections after 5 points within 10 cm of back line

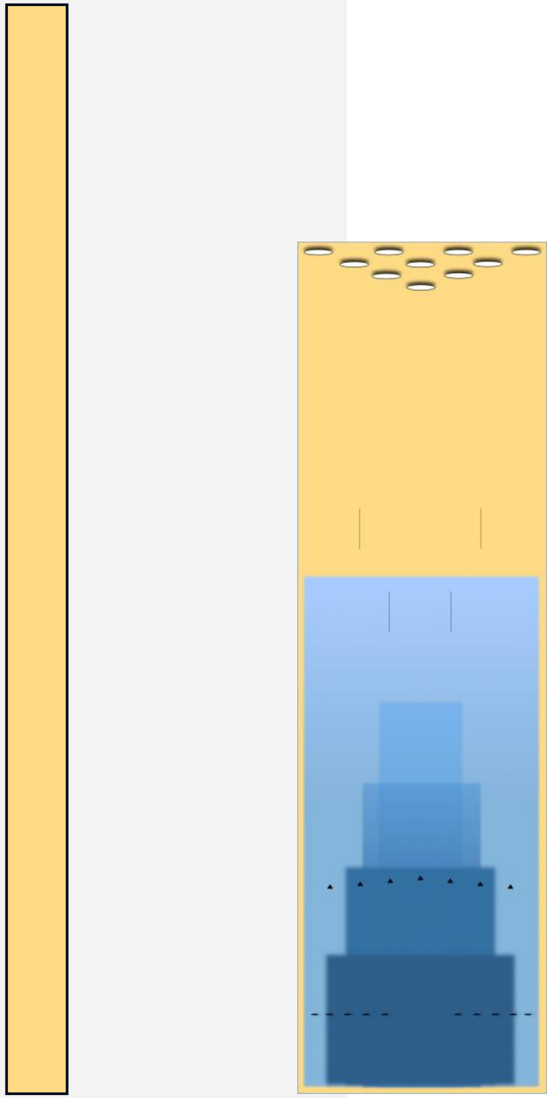
Interpolation & Smoothing
- Moving-median filter + Z-score test (MAD)
- Interpolate missing values
- Apply Savitzky–Golay filter for smooth trajectory




Brunswick
AZCONA
OPEN
2020
1
2
3

Lane Reconstruction

02.3



Lane Reconstruction

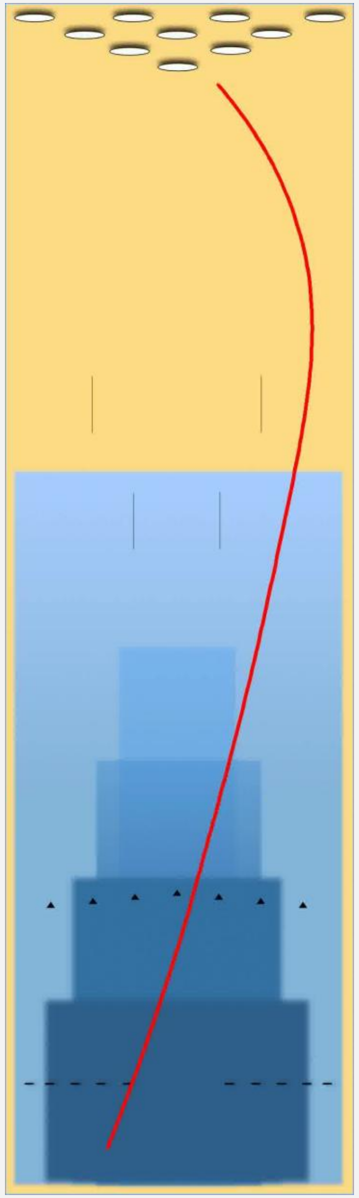
The goal of this section is to map the detected lane to the real lane dimensions

- OpenCV function findHomography() compute the transformation matrix
- All detected features (e.g., ball,) are transformed into a common reference space

Real lane is too long relative to width for easy visualization (aspect ratio ~18:1). To improve interpretability, we apply a non-isometric transformation to a standard stretched lane used in professional bowling tools.

Trajectory

02.4



Trajectory of the ball

In this section we construct the trajectory of the ball for each frame in different environments:

- Trajectory on the **original video**
- Trajectory on the **reconstructed lane**
- Trajectory on the **deformed reconstructed lane**

Trajectory on the original video

Direct plotting of contact points can fail due to camera motion.
Our solution was to apply the current frame’s homography to all past contact points.
The trajectory is entirely refreshed every frame, however in this way a possible rotation of the camera does not modify the position of the trajectory.

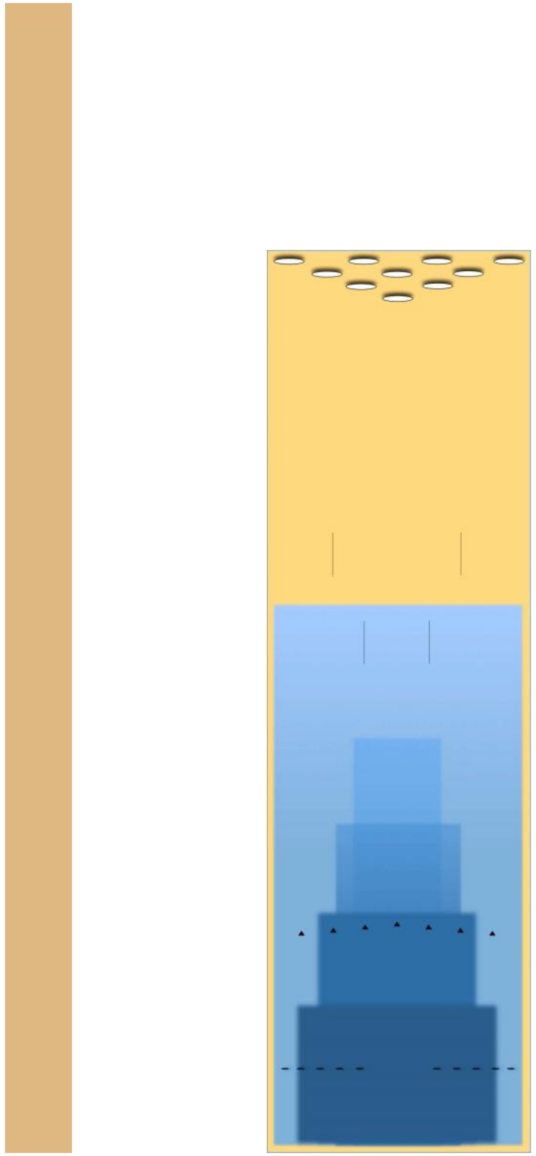
Trajectory on reconstructed lane

Trajectory on the original Lane

- Contact point: estimated as the lowest point of the detected ball circle
- Apply the homography to map this point into the canonical lane frame
- Obtain accurate ball position per frame in standardized coordinates

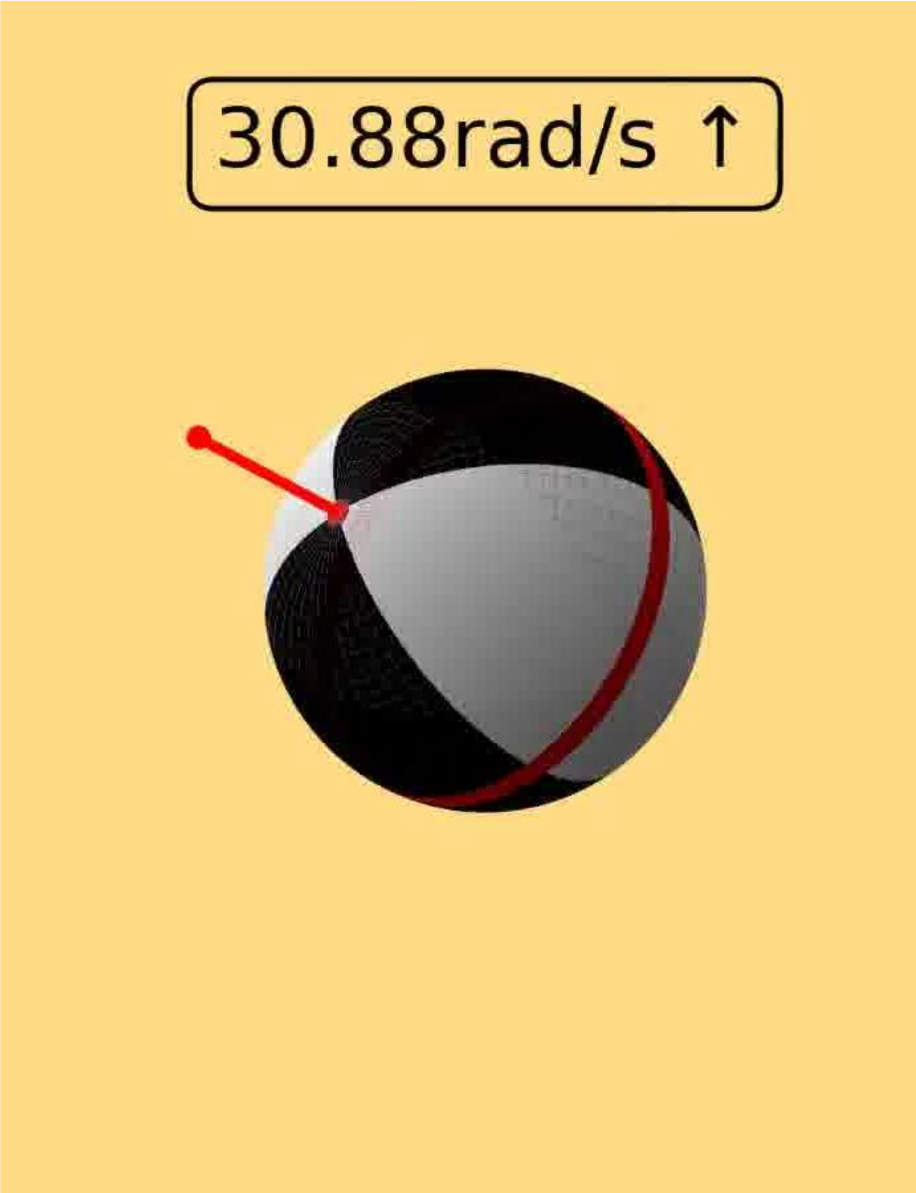
Trajectory on the deformed lane

- Transformed ball positions are rescaled accordingly



Spin Estimate

02.5



Spin analysis

Starting from the knowledge of the position and dimension of the ball in each frame is possible to compute the rotation of the ball. In this section we:

- Calculate the **axis of rotation**
- Calculate the **angular velocity**
- Create a live **3D representation** of the ball

Spin processing

Feature Detection & Tracking

- Crop the analysis only to the region inside the ball's detected circular boundary
- Perform **Shi-Tomasi corner detection** in order to select the points of interest on the surface of the ball
- Perform **Lucas-Kanade optical flow** (pyramidal)
- **Forward-backward validation** in order to keep only reliable tracks

Extract Rotation and Axis

- Get 3D points
- Compute the rotation via **Kabsch Algorithm**
- Extract **rotation axis and rotation angle**

$$\theta = arcos(2Tr(R) - 1)$$

$$a = \sin \theta_1 \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$$

- Compute the **angular velocity**

$$\omega = \theta \cdot FPS$$

Spin post-processing

Axis Post-Processing

- Flip the axis with z > 0
- Remove outliers based on inconsistent x/y signs
- Fit linear model to x/y and discard high-deviation frames
- Apply Gaussian smoothing to reduce residual noise

"ω" Post-Processing

- Detect outliers via Z-score thresholding
- Interpolate and smooth angular velocity over time

Final Results

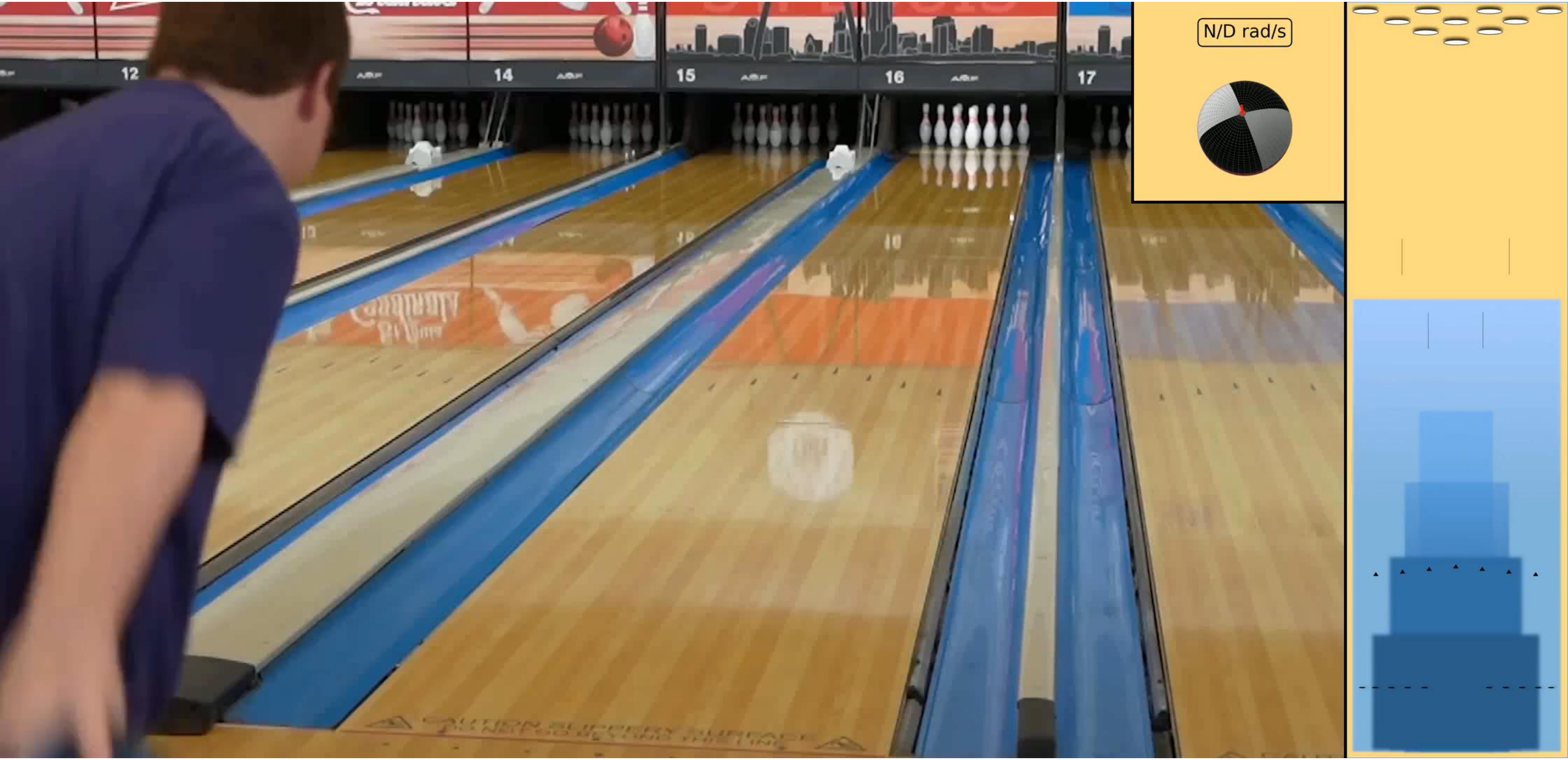
03


N/D rad/s


Brunswick
AZCONA
1
2
3
N/D rad/s


N/D rad/s


1
2 3
SUPER 9 NO-TAP TOURNAMENT
4 5
18
EMPLOYEES ONLY BEYOND THIS POINT
N/D rad/s


N/D rad/s
12
14
15
16
17

Conclusions

04

Conclusions

About what we have done

- Presented a full pipeline for analyzing bowling ball trajectory and spin from monocular video using classical computer vision.
- Robust to camera angle, resolution, and motion — no deep learning required.
- Used line detection, Hough Transform, homography, and optical flow for accurate analysis.
- Homography enabled consistent tracking under dynamic conditions.
- Lays groundwork for practical, accessible sports analytics.

Future work

- Mobile/web tool, deep learning for robustness, multi-view support, and real-time applications.

Grazie per l’attenzione