POVa (Computer Vision): Ball Tracking in 3D from Multiple Cameras

Dave Galea Nikolas Papadakis Lucas Labhini

Brno University of Technology Faculty of Information Technology

Introduction and Task

The goal of this project was to track a single-colored ball in three-dimensional space using multiple cameras. The main focus was on the 3D aspects of the task, including camera calibration and 3D localization of the ball.

Software Flow

The software pipeline consists of several stages, including camera calibration, rectification, ball detection, triangulation, and visualization. Figure 1 illustrates the flow, where red indicates provided code, blue represents scripts, and green shows intermediate files.

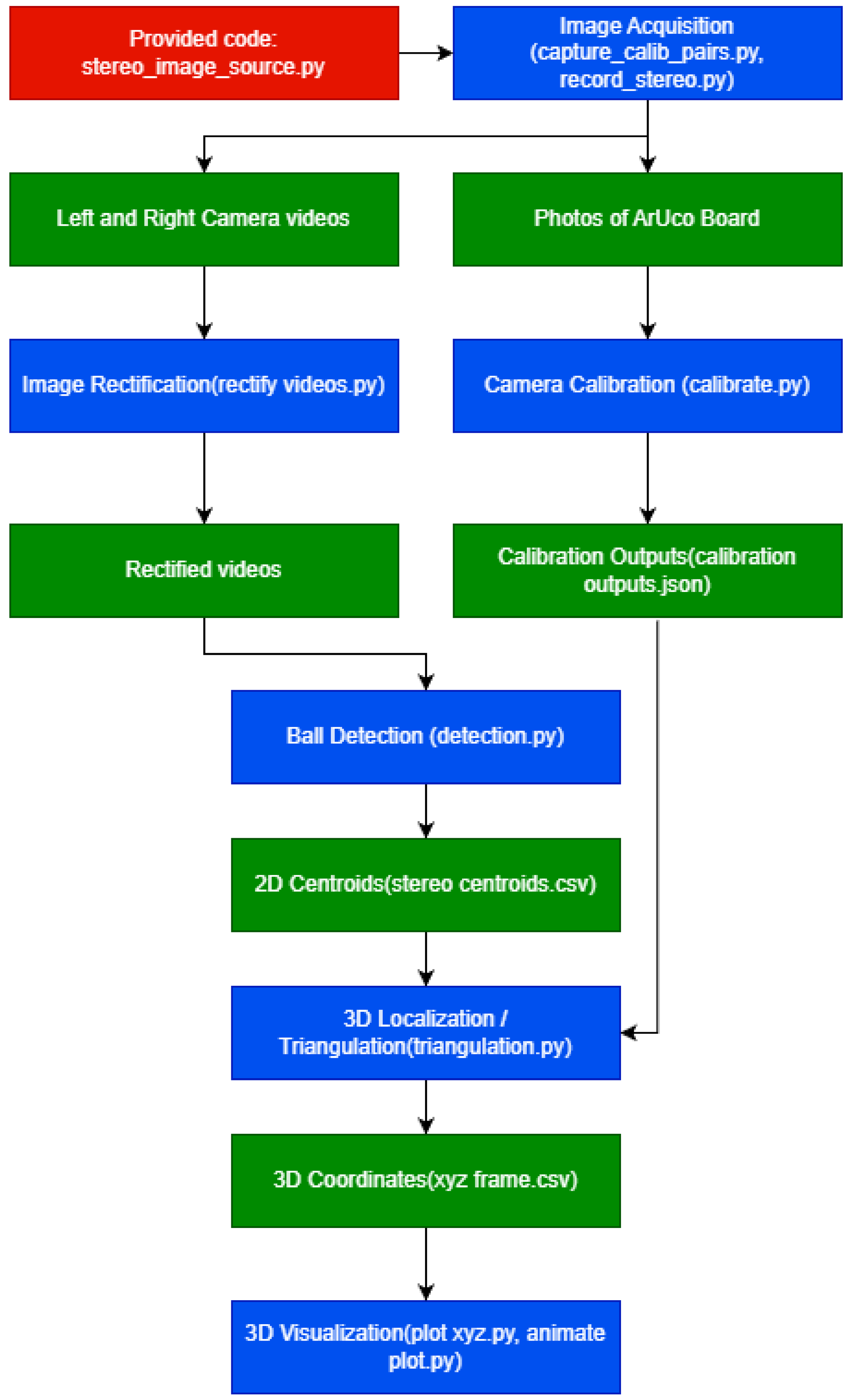

Figure 1. Software pipeline overview.

Camera Calibration

To achieve accurate 3D tracking, both intrinsic and extrinsic parameters were estimated. The intrinsic parameters include focal length, principal point, and lens distortion, while the extrinsic parameters describe the rotation and translation between cameras. The calibration was performed using OpenCV's ArUco GridBoard with a 5x7 marker configuration, where each marker was 0.02 meters in size with a separation of 0.007 meters. The stereo calibration resulted in a root mean square error of 3.81 pixels and a baseline distance of 0.0676 meters between cameras.

Camera Setup

Figure 2. Setup of the Cameras.

Rectification

Rectification was performed to align epipolar lines across the two camera views, simplifying the correspondence problem. This step uses OpenCV's `cv2.stereoRectify` function with the calibration matrices. The horizontal alignment was verified visually by overlaying guide lines on the rectified frames, ensuring proper alignment of corresponding points.

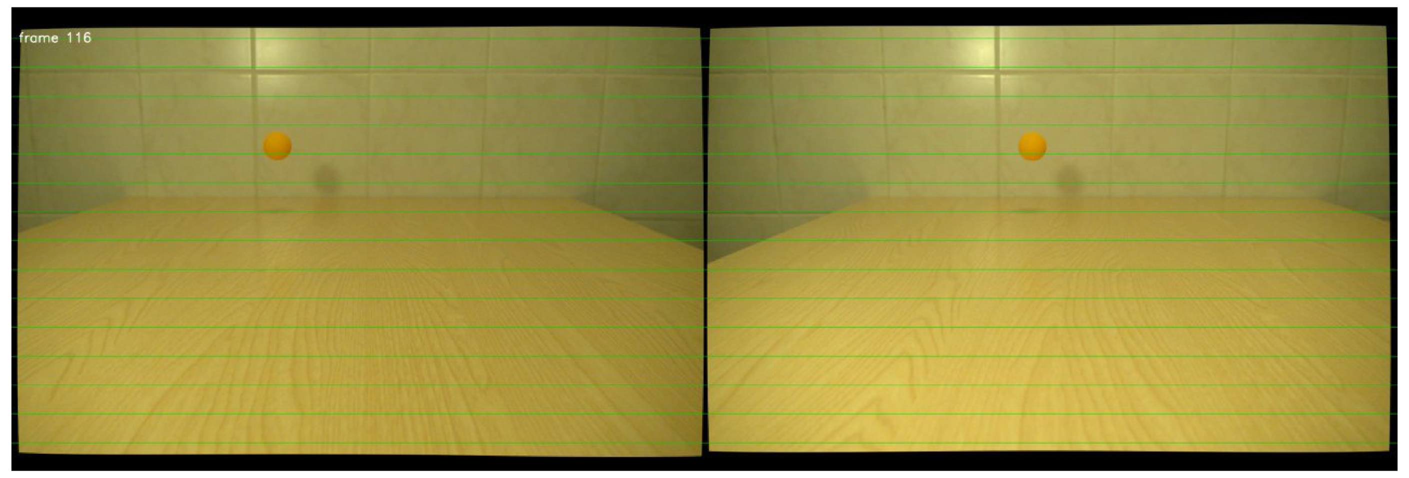

Figure 3. Rectified frames with horizontal guides.

Ball Detection

Ball detection is achieved by converting the frames from BGR to HSV color space and thresholding to isolate the orange color of the ball. Noise is removed using morphological operations, and reflections on the ball are addressed by considering only the top half of the detected contours. The centroid of the detected contour is then used as the 2D ball position in each camera frame for each respective camera.

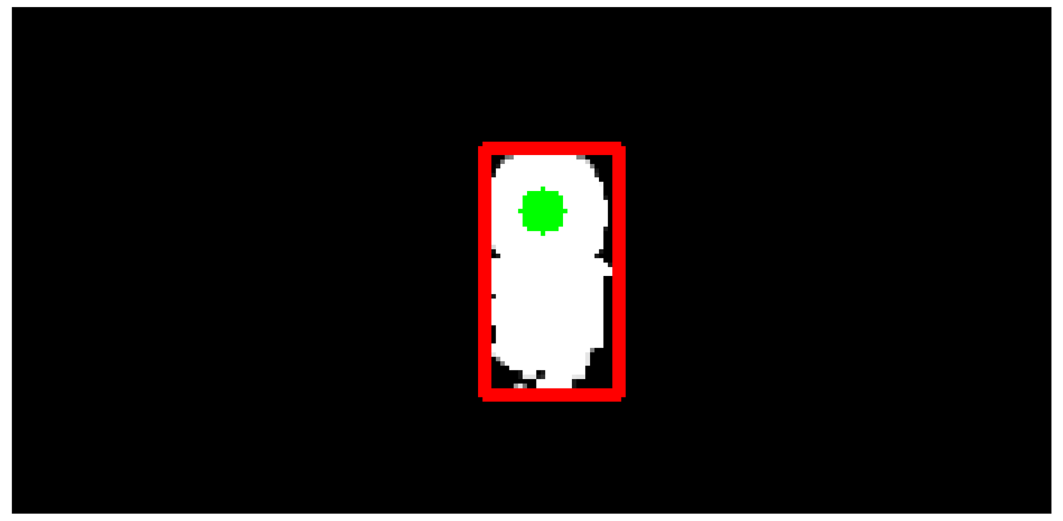

Figure 4. Ball detection with centroid ignoring reflection.

Triangulation

Once the ball is detected in both camera views and the respective centroids are calculated, triangulation is performed to compute its 3D coordinates. The 2D centroids from each camera are combined using the corresponding projection matrices to generate the 3D position.

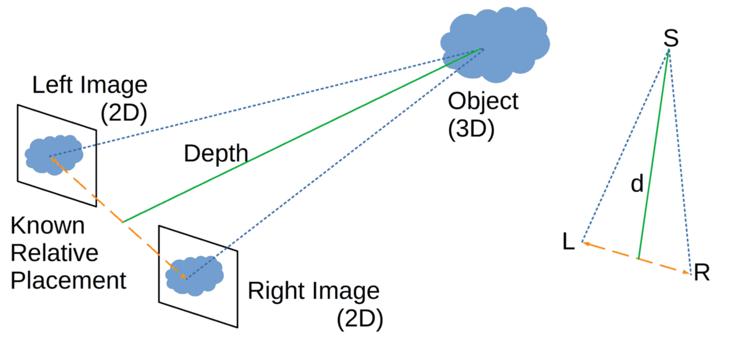

Figure 5. Triangulation concept.

3D Visualization and Evaluation

The computed 3D positions of the ball were visualized by a graph to observe the trajectory over time. When compared with the original video, the bouncing motion of the ball was captured accurately, although minor errors were observed due to calibration error.

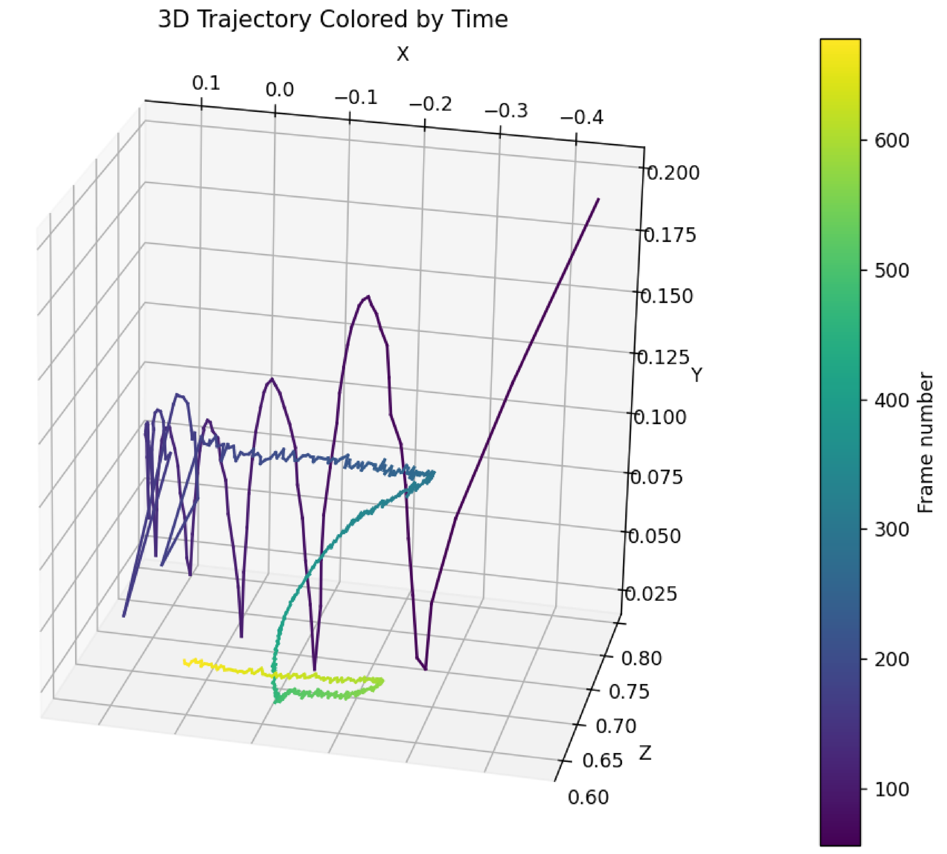

Figure 6. 3D trajectory of the ball over time.

Conclusion

This project successfully demonstrates 3D tracking of a single-colored ball using multiple cameras. The core functionality, including detection and triangulation, was implemented effectively. Although minor smoothness issues occurred due to calibration limitations, the system provides a practical example of multi-camera stereo vision applications.

References

B. Preim and C. Botha, *Visual Computing for Medicine*, 2014.
T. Batpurev, *Python Stereo Camera Calibration*, GitHub, 2020.
Z. Chen et al., *A new image rectification algorithm*, 2003.
X. Ma, *Uncalibrated Stereo Rectification*, GitHub, 2019.
A. Rosebrock, *Ball tracking with OpenCV*, 2015.
A. Valsaraj et al., *Stereo vision system on FPGA*, 2016.
S. Kim, *3D Vision Soccer Ball Tracking*, GitHub, 2018.