Computer Vision - Final Project

Amitai Ovadia

Abstract

In this project, I aimed to perform **3D tracking of a bouncing ball** using one calibrated camera.
I took the following steps:

1. I calibrated the camera and extracted the **intrinsic camera parameters** (slide 5).
2. For each frame in the video, I did the following:
    - **Undistorted** the frame (using a matlab function)
    - **Segmented the Ball** (slide 3).
    - Extracted the **center point and radius in 2D** (slide 4).
    - Extracted the **3D location of the ball's center** (slide 6-7).

1. I Found the **3D trajectory** (slide 8).
2. I Calculated the **speed** of the ball (slide 9)

Segmenting the Ball in the image

The Ball was segmented in every frame using:

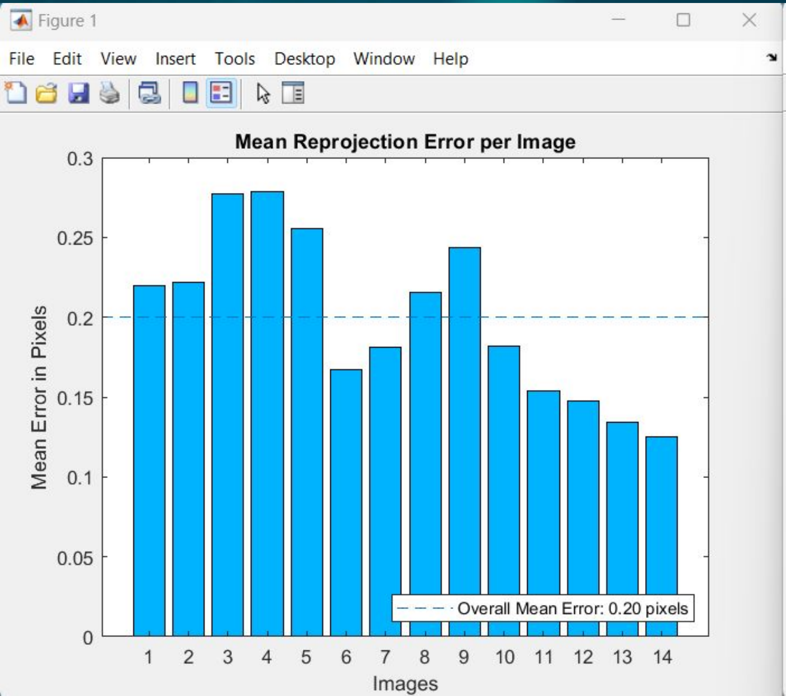
- background subtraction
- thresholding
- binary operations

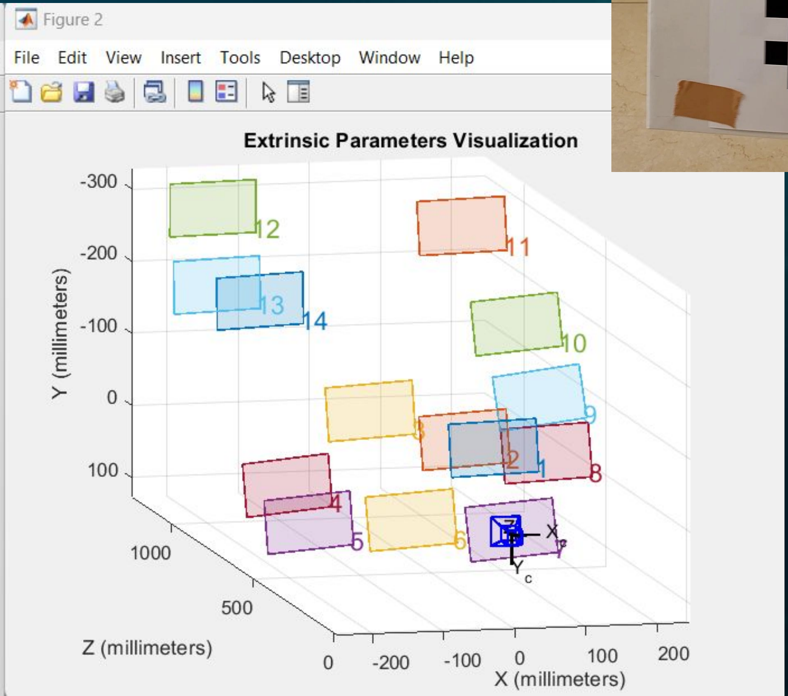
Finding Radius and center 2D

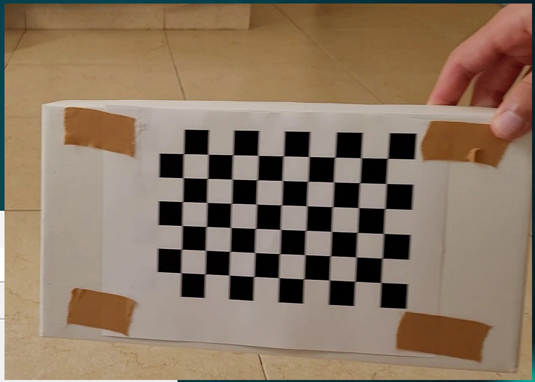
A robust algorithm was then applied to extract the center and radius of the ball in 2D:

- Found the **ball mask** using thresholding and binary operations
- The **edges** of the ball mask were extracted.
- For each point, it's **corresponding most further edge point** was found (the other side of the ball),
- **distance** (the estimated **diameter**) calculated.
- **Outlier** pairs were identified and removed.
- 2D **Center** was calculated as the mean.
- **Radius** was calculated as half of the average pair distance.

Camera Calibration using matlab toolbox and chessboard







Finding 3D rays using 2D image points

- **K is found in calibration**
- I work in the **camera coordinates system** (need only to find angles) hence R=I and t=0
- **'u' is a 3D ray** in homogeneous coordinates
- **'P' is a 2D point** in homogeneous coordinates
- **M is the 3x4 Camera matrix** (3D -> 2D)
- **M+ is the 4x3 inverse projection matrix** (2D -> 3D)

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$M = K\left[R|t\right]$$

$$M = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M^{+} = pseodo\_inverse\left(M\right)$$

$$u = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$u = M^{+}p$$

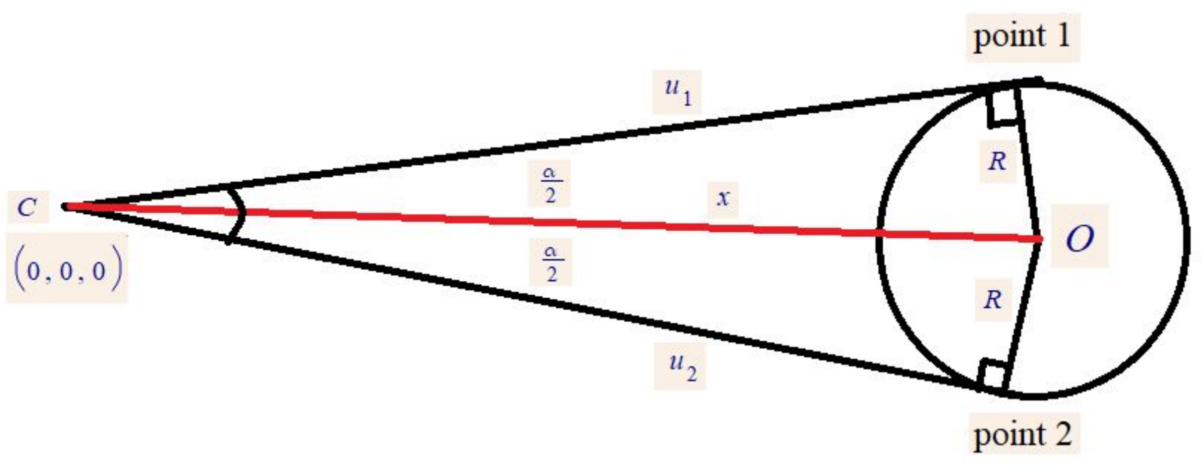
Calculating Ball 3D center

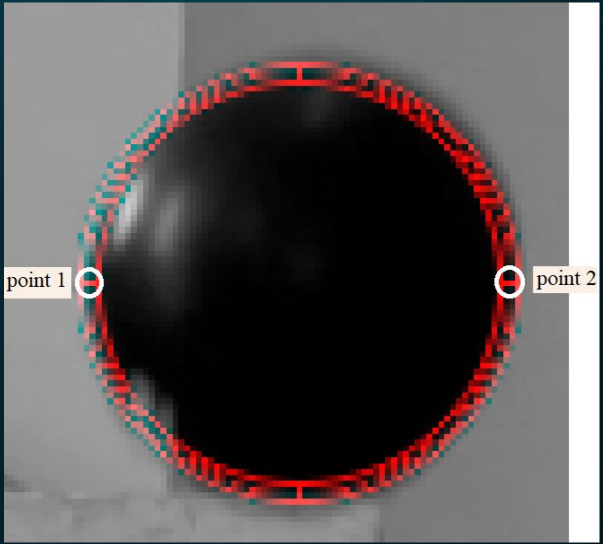
In every frame:

- Found O (ball center) in 2D
- First I found p1 and p2 (2D)
- Found rays u1 and u2 (3D)
- Found angle alpha between u1, u2
- Calculated x
- Found O (ball center) in 3D

$$\hat{O}_{3D} = norm\left(M^{+} O_{2d}\right)$$
$$O_{3D} = x \cdot \hat{O}_{3D}$$

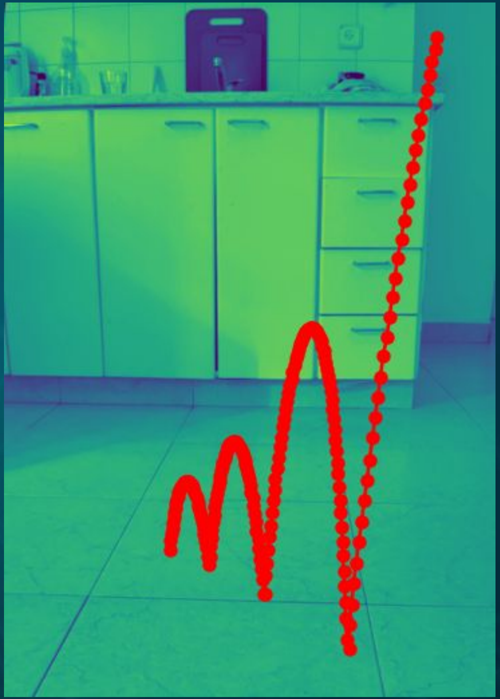
$$sin(\tfrac{\alpha}{2}) = \frac{R}{x}$$
$$x = \frac{R}{sin(\frac{\alpha}{2})}$$



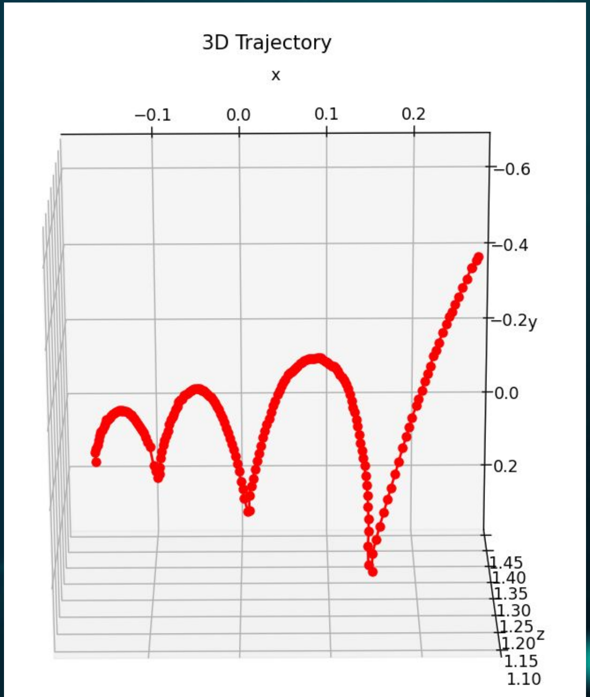


Results

2D trajectory

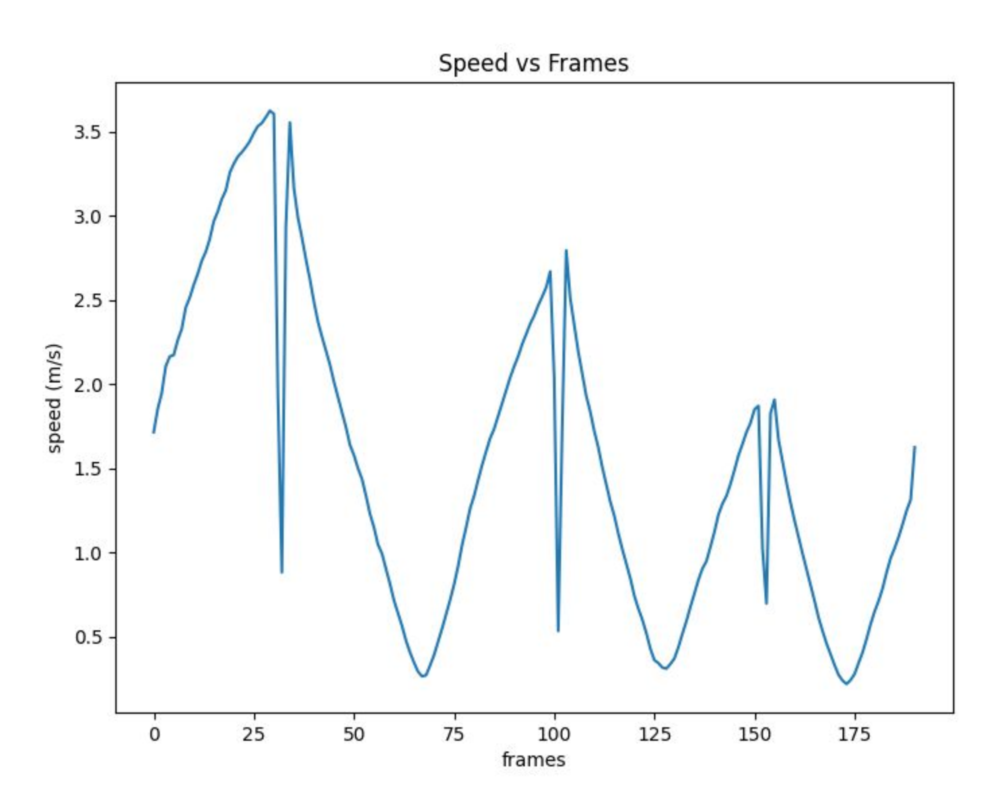


3D trajectory



Speed

- Calculated using:
  W = window size (3)
  (X[i] - X[i-w])/(fps/w)



RESOURCES

The project was implemented in Python and matlab and the videos were taken by a Galaxy s20.

- Numpy
- Matplotlib

THANKS!