NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

**Software Team Project Report on the Topic:**

# Positioning and Trajectory Prediction of a Table Tennis Ball Using Cameras Array

**Submitted by the Students:**

| | |
|---|---|
| group #БПМИ229, 2rd year of study | Bakin Denis Filippovich |
| group #БПМИ203, 4th year of study | Paleev Daniil Alekseevich |

**Approved by the Project Supervisor:**

Kolesnichenko Elena Yurievna
Candidate of Physical and Mathematical Sciences
Faculty of Computer Science, HSE University

**Co-supervisor:**

Simagin Denis Andreevich
Master of Computer Science
Head of the HSE Robotics Group

Moscow 2024

# Contents

# Annotation

This work, part of the HSE Robotics Group's Handy project [3], aims to enable a robot to play table tennis by researching, implementing, and validating ball positioning and tracking in a high-frequency robotics system. This paper focuses on the process of triangulation and camera calibration, excluding image segmentation and machine learning details.

The robot must quickly locate the ball's precise position, achieved by combining segmentation results from multiple views and triangulating the position. Camera calibration is necessary to obtain intrinsic parameters for calculations. The project outlines the theory, motivation behind selected methods, and compares precision among various triangulation methods.

# Keywords

CV, ML, triangulation, high-frequency systems, pose estimation, projective geometry

# 1 Literature review

There are a lot of articles regarding the topic of this work. However, very few researchers design their proposals as high-frequency systems and rather concentrate on the quality than on more practical applications.

For example, article about reliable tracking of a table tennis ball [2] is very related to this work. However, author does not fully cover the topic regarding the precise frequency of detection and approaches for result validation.

As a reference of the final goal for the whole project the article [6] about a robot for playing ping-pong can be used.

# 2 Introduction

The high-frequency systems that can precisely and almost instantly calculate the position of an object are very important. With the recent advances in technology, they are widely spread in wild nature preservation [5], security industry [1] and [7]. There are multiple methods to do that.

This work is based on using multiple cameras that are synchronized and provide segmentation results every 10ms. Then, based on the provided data, the triangulation is performed. It is the process that estimates the position of an object in 3D world coordinates from 2D coordinates from array of cameras. Then these coordinates can be used in further application that will not be covered.

The development consisted of the following steps:

1 Synchronization of camera frames

2 Calibration of a single camera

3 Stereo calibration

4 Segmentation of a moving object

5 Triangulation of a moving object

# 3 Main structure

Before the detailed description of each development step it is necessary to state a more global structure of the high-frequency system. The pipeline's design was implemented in the

following manner.

First of all, frames from all cameras in an array are captured and checked for successful synchronization. Then the frames are transferred to the segmentation module that is being implemented by Daniil Paleev and will be described separately from this work. As an output to each frame the segmentation module returns a rectangle that fully covers the detected tennis ball. These coordinates are sent to the detection module, that triangulates the 3D position of the ball and returns them as the result of the whole cycle of capture-segmentation-detection. The steps before and after Daniil Paleev's segmentation module were implemented by Denis Bakin.

Moreover, it should be noted that even though the final goal was to present a real-time high-frequency system that provides 100 Hz detections and determines the 3D coordinates of a ball within 10 millisecond, this work will describe how separate modules work and leave real-time functionality aside.

The brief descriptions of completed parts of the project can be found below.

# 4 Synchronization of camera frames

This stage is indeed important in the whole pipeline, because it is crucial for detection to get a pair of synchronized frames, where a tennis ball is shown at the same moment and is not moved between a set of frames. To achieve that, different types of triggers were tested:

First of all, industrial cameras (model: Huanteg Vision HT-SUA230GC-T1V-C) supported three types of triggers.

1 Automatic trigger (was not used because it provides no control over synchronization);

2 Software trigger (camera is given a callback and the trigger function is called from a control program)

3 Hardware trigger (camera is given a hardware impulse to take a picture)

It was decided to use hardware trigger in order to guarantee physical synchronization. To use this approach one of the cameras is considered to be "master" and supposed to get software trigger by USB, while others are considered to be "slaves" and wait for hardware impulse to occur.

Secondly, to calculate the amount of information that should be transmitted in case of an array of two cameras, let us multiply the following parameters:

$$2 \text{ camera} \cdot 1 \text{ channel} \cdot 1920\text{px} \cdot 1200\text{px resolution} \cdot 100 \text{ FPS} \approx 440\text{MB}$$

In order to sustain such channel bitrate it was decided to implement low-level buffer management instead of relying on a conventional frameworks such as ROS2 that was used initially.

Lastly, it is still the subject of experiments, whether initial single-channel Bayer image or a converted RGB version is better to use for a detector. To provide some deeper information about digital camera sensors, let us provide some graphics on the topic.

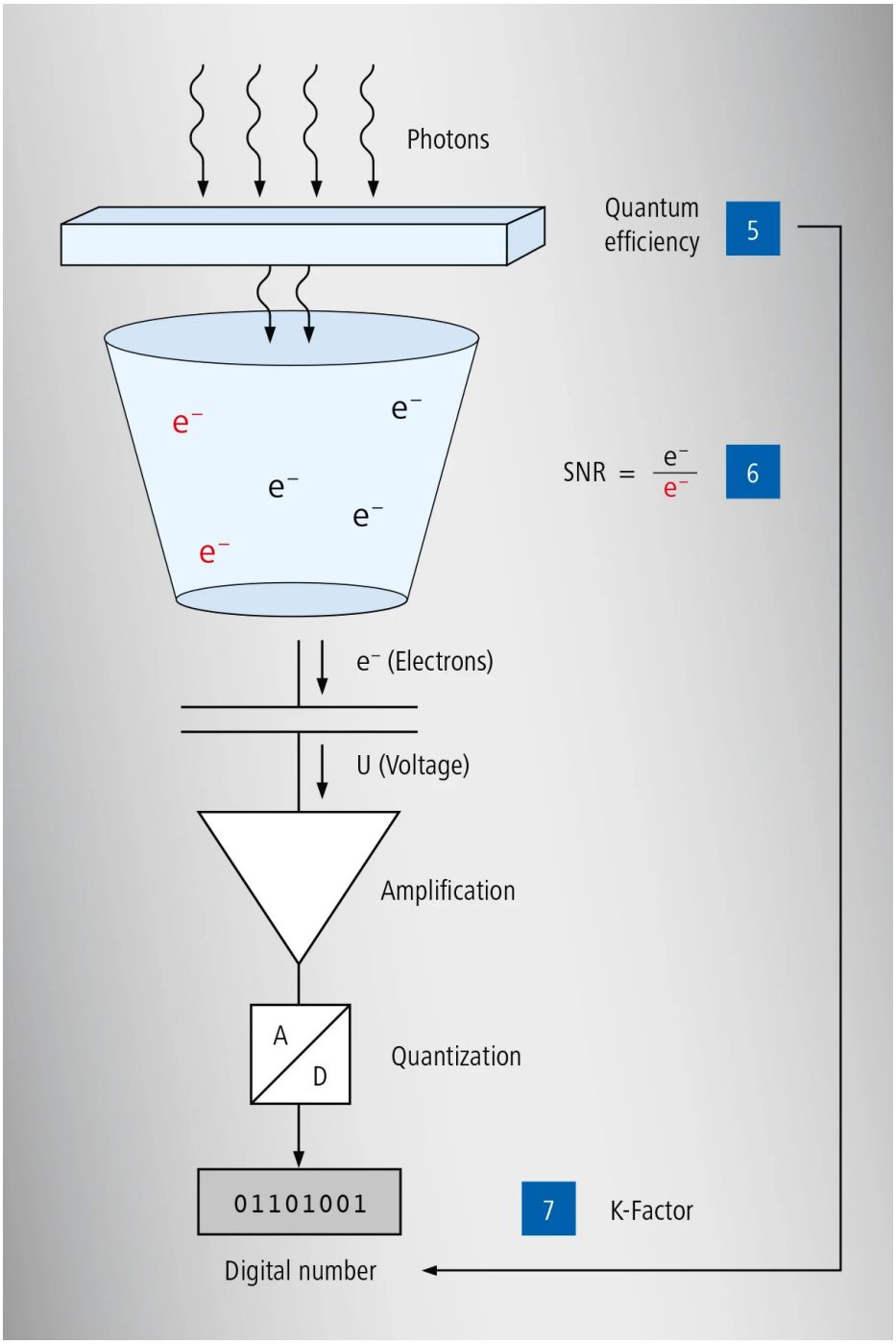


Figure 1. How sensor of a digital camera works.

So, the more photons are captured by the photodiodes, the higher voltage is accumulated on the capacitor. Camera "takes" a frame by measuring voltages on the capacitors and quantizing

them into digital information. However, colors can be still determined from the captured frame, because all photodiodes are covered with color-specific filters. They can be red, green or blue and usually placed according to one of the Bayer filters.

One of the pattern looks like (where R, G and B stand for red, green and blue photodiode's filter respectively):

$$\begin{bmatrix} \dots & B & G & B & G & B & \dots \\ \dots & G & R & G & R & G & \dots \\ \dots & B & G & B & G & B & \dots \\ \dots & B & G & B & G & B & \dots \end{bmatrix}$$

# 5 Calibration

## Pinhole camera model

In this work model of pinhole camera is used. So, there are certain parameters that determine how 3D points in camera coordinate frame are mapped to projection plain. All points and equations will be written in homogenous coordinates.

$$p_{\text{plain}} = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = [K|0]\, p_{\text{cam}} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ x_{\text{cam}} \\ 1 \end{bmatrix}$$

where $p_{\text{plain}}$ is a 2D point on a projection plain, $p_{\text{cam}}$ is a 3D point in the coordinate frame of the camera, $(f_x, f_y)$ - focus of the camera, $(c_x, c_y)$ - principal point of the camera (the point where principal axis of the camera intersects with the image plain), $H$ - camera matrix of a pinhole camera.

## Intrinsic parameters

Another important aspect are the distortion coefficients of the camera. They express the mapping from rectified undistorted points of an ideal projection plain to a distorted real-world frame. Distortion is a non-proportional deviation of a real-life scene after projecting it to the image plain. In most cases it can be described by an high-degree equation with 5 coefficients.

Together with the camera matrix distortion coefficients are called intrinsic parameters and are used in the majority of computer vision algorithms, because undistortion restores the similarity

relations between real-world and projected objects.

Intrinsics can be estimated as the solution of an optimization problem from object-projection correspondences. These correspondences are usually obtained from straight and pre-measured pattern detections, such as chessboard, aruco or charuco boards.

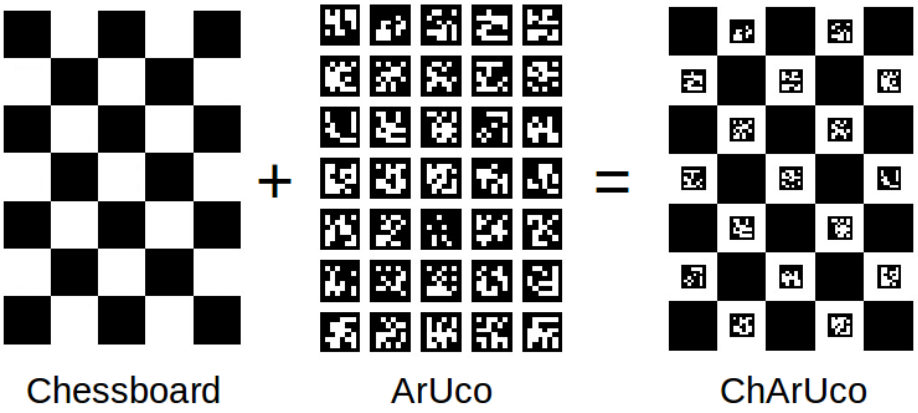


Figure 2. Construction of charuco board as a combination of an aruco grid board and conventional chessboard

The main advantages of charuco board over aruco grid board and chessboard are:

1 higher detection precision especially compared with conventional chessboard;

2 only a part of the board (several aruco markers, to be precise) are required to be detected in order to determine some of the chessboard corner.

In order to automate the process the "calibration pipeline" [4] was developed. It is a visualization tool that guides the user through the stages of gathering, filtering and processing camera frames. It was created mainly for convenience and performance reasons.

It is important to note that too many detected chessboards lead to long calculations and even less accurate results, compared to an attempt with less collected frames. To avoid gathering too many frames the following metric was introduced.

$$\text{IoU}\,(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

where $A$ and $B$ are polygons to measure similarity of.

It allows to measure how large the newly covered area is and then decide whether it is worth collecting for calibration or not. This approach allows us to reduce chances of excessive collection of frames to its minimum.

## Camera to camera pose estimation

Another step to getting the set of cameras calibrate is the estimating of its relation in space to each other. This step is required to determine the transformation (expressed in rotation matrix of linear transformation and translation vector) from first camera's coordinate frame to the coordinate frame of the second one.

It can be done quite similarly to mono calibration, a. e. from the correspondences of the object points to the images points on both frames. It means that the pattern should be placed in the common view of the cameras and detected simultaneously by all of them. This requirements introduces quite a lot of difficulties, because unlike the stereopair (where principal rays of cameras must be colinear and places close to each other) cameras are placed around the table and have pretty small field of common view.

To solve the issue aruco board was finally chosen over charuco board, preserving convenience of working with edges of the frame, but allowing better stability of detections, because aruco marker are significantly larger than markers on the charuco board.

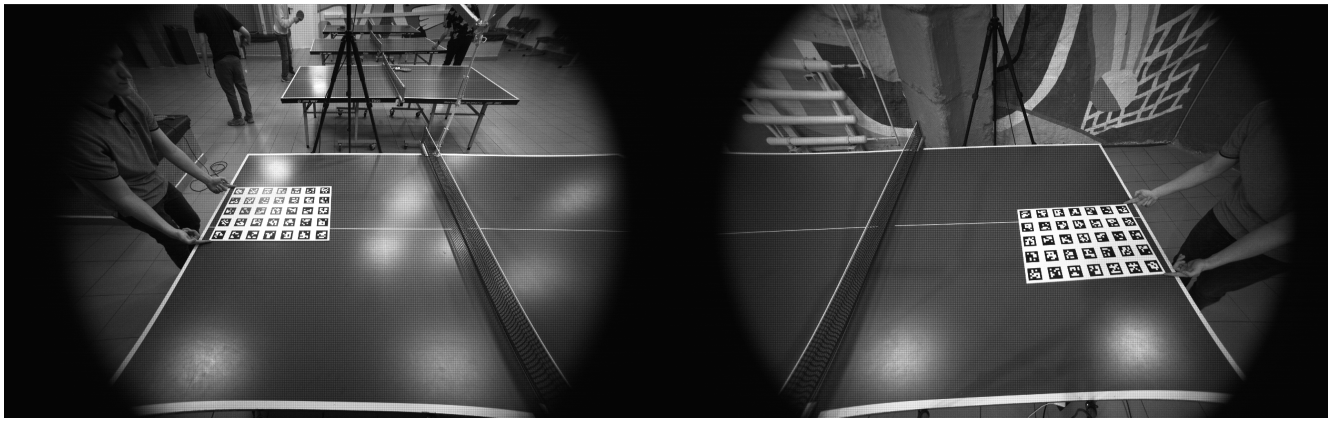

Figure 3. Example of stereo calibrating with aruco board

As a result of a stereo calibration rotation matrix $R \in Mat_{3\times3}$ and translation vector $T \in Mat_{3\times1}$ are obtained. Together they perform a change of basis from the coordinate frame of the first camera to the frame of the second one:

$$X_2 = RX_1 + T$$

where $X_1, X_2$ are points in the coordinate frames of the first and he second camera respectively.

# 6 Triangulation

Triangulation is the process of determining a 3D point from known 2D coordinates of image plains of each of $n$ frames, taken simultaneously. Let us describe several methods of triangulation that were implemented and compared in this work.

## DLT

Direct linear transformation (DLT) is a method to solve a set of equations up to a scalar factor.

Let $X = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ be a 3D point in homogenous coordinates that is observed by $n$ cameras. $S_1, ..., S_n$ are 2D points on image plains of $n$ cameras, where $S_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T$.

So for $i$-th camera projection equation can be denoted as ($K_i$ is the $i$-th camera matrix):

$$S_i = [K_i|0]\, X$$

The cross product of $S_i$ and $[K_i|0]\, X$ equals to zero since these vectors are colinear. Then:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \times \begin{bmatrix} k_{i1} & k_{i1} & k_{i1} \end{bmatrix} X = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \times \begin{bmatrix} k_{i1}X \\ k_{i1}X \\ k_{i1}X \end{bmatrix} = \begin{bmatrix} y_i k_{i3}X - k_{i2}X \\ k_{i2}X - x_i k_{i3}X \\ x_i k_{i3}X - y_i k_{i3}X \end{bmatrix} = \begin{bmatrix} y_i k_{i3} - k_{i2} \\ k_{i2} - x_i k_{i3} \\ x_i k_{i3} - y_i k_{i3} \end{bmatrix} X = \vec{0}$$

where $k_{ij}$ is the j-th row of the i-th camera matrix.

Since the last two equations of a homogenous system (equals to a zero vector) are colinear, the last row can be discarded. So, one point of view does not determine 3D coordinates entirely, that is true. To sum up, one observation of a 3D point gives us the following equation:

$$\begin{bmatrix} y_i k_{i3} - k_{i2} \\ k_{i2} - x_i k_{i3} \end{bmatrix} X = \vec{0}$$

Let us consider all $n$ points of view and possible noise, coming from not ideal calibration parameters and finite machine precision. Then our task is to find the best approximation of a solution of the following system of equations:

$$AX = \begin{bmatrix} y_1k_{1,3} - k_{1,2} \\ k_{1,2} - x_1k_{1,3} \\ y_2k_{2,3} - k_{2,2} \\ k_{2,2} - x_2k_{2,3} \\ \vdots \\ y_nk_{n,3} - k_{n,2} \\ k_{n,2} - x_nk_{n,3} \end{bmatrix} X = \vec{w}$$

where $\vec{w}$ is a vector of noise.

Then the solution can be found from singular value decomposition (SVD) and its properties

$$\|w\|_2 = w^T w = X^T A^T AX = X^T V \Sigma \underbrace{U^T U}_{=I} \Sigma V^T X = X^T V \Sigma^2 V^T X \to \min$$

The norm of $w$ will be minimal in case we take $X = v_r$, because SVD places $r = \operatorname{rk} A$ singular values in a descending order.

$$\|w(X)\|_2 \to \min \Leftrightarrow \begin{cases} X = v_r \\ \|w(X)\|_2 = \sigma_r^2 \end{cases}$$

Also, it should be noted that instead of singular value decomposition the algorithm of least squares (ALS) can be used.

## Midpoint method

Any 2D point on an image plain is mapped to a ray of $O-$ origin point of a pinhole camera and $\begin{bmatrix} x & y & w \end{bmatrix}^T$ – 2D point in homogenous coordinates. Such mapping allows us to consider all 2D coordinates as some rays in 3D space, then the origin object in the scene will be the point of intersection of all the rays (one from each camera).

However, in real-world applications rays almost sertainly does not intersect, that requires us to find minimize the following metric:

$$dist(X) = \sqrt{\sum_{i=1}^{n} \rho(X, l_i)^2}$$

where $l_i$ is the $i$-th ray from $i$-th camera, $X = \alpha \cdot \begin{bmatrix} x & y \end{bmatrix}^T = \begin{bmatrix} x & y & \alpha \end{bmatrix}^T, \forall \alpha \neq 0$ and $\rho(X, l_i)$ is the distance between point $X$ and ray $l_i$.

For example, for an array of two cameras the point will be the middle of the shortest segment between the two rays.

### Via rotation and translation

Rotation and translation parameters or estimated camera to camera pose as was mentioned above can be used for triangulating 3D object position from multiple-view data. Lets us describe the derivation of this method for two cameras.

Let $r_i$ be the $i$-th row of $R = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$ rotation matrix. It was shown above that there is linear transformation between coordinate frames of two cameras:

$$X_2 = RX_1 + T$$

Let $(y_1, y_2)$ and $(y'_1, y'_2)$ are image points from two cameras, $X = (x_1, x_2, x_3)$ is the 3D point in tha coordinate frame of the first camera.

Then:

$$y'_1 = \frac{x'_1}{x'_3} = \frac{r_1 \cdot (X' - T)}{r_3 \cdot (X' - T)} = \frac{r_1 \cdot (y - \frac{T}{x_3})}{r_3 \cdot (y - \frac{T}{x_3})} \Rightarrow x_3 = \frac{(r_1 - y'_1 r_3) \cdot t}{(r_1 - y'_1 r_3) \cdot y}$$

The final coordinates (triangulated 3D point) can be defined as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_3 \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Note that the same derivation can be performed for the second camera, but will not be equal in the presence of noise and finite machine accuracy. In fact, symmetric derivation can be combined with the described one and used to increase the triangulation accuracy and stability of an algorithm.

# 7 Results

As the result of this work successful triangulation was performed with the appropriate precision for MVP. Also, separate trajectory prediction module was tested on the data provided from triangulation.

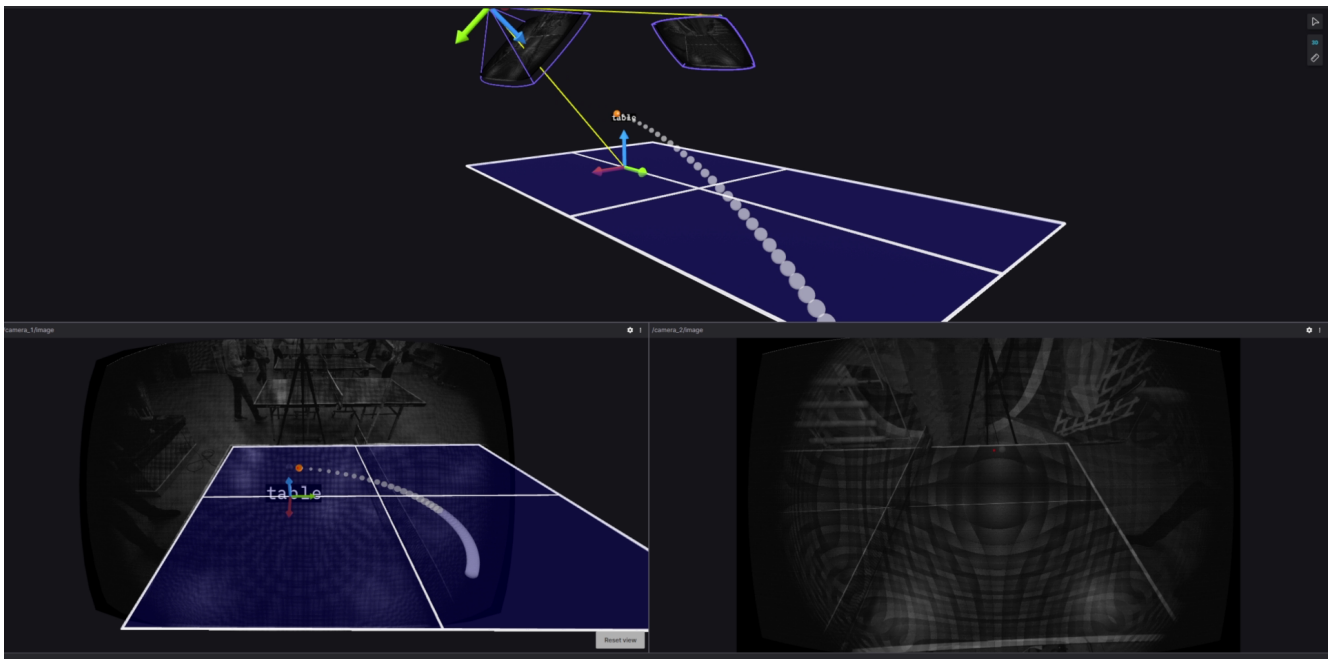


Figure 4. Visualization of the captured scene. White trajectory is shown with respect to confidence of prediction

# References

[1] Robin Bhadoria, Naman Bhoj, Hatim Zaini, Vivek Bisht, Md Nezami, Ahmed Althobaiti, and Sherif Ghoneim. "Artificial Intelligence for Creating Low Latency and Predictive Intrusion Detection with Security Enhancement in Power Systems". In: *Applied Sciences* 11 (Dec. 2021), p. 11988. DOI: 10.3390/app112411988.

[2] Sebastian Gomez-Gonzalez, Yassine Nemmour, Bernhard Schölkopf, and Jan Peters. "Reliable Real-Time Ball Tracking for Robot Table Tennis". In: *Robotics* 8 (Oct. 2019), p. 90. DOI: 10.3390/robotics8040090.

[3] HSE Robotics Group. *Handy*. URL: https://github.com/robotics-laboratory/handy.git.

[4] HSE Robotics Group. *alibration automation*. URL: https://youtu.be/HiV_22iUuEY (visited on Sept. 13, 2023).

[5] Ambreen Hussain, Bidushi Barua, Ahmed Osman, Raouf Abozariba, and Taufiq Asyhari. "Low Latency and Non-Intrusive Accurate Object Detection in Forests". In: Dec. 2021, pp. 1–6. DOI: 10.1109/SSCI50451.2021.9660175.

[6] Asai Kyohei, Nakayama Masamune, and Yase Satoshi. "The Ping Pong Robot to Return a Ball Precisely". In: 2019. URL: https://api.semanticscholar.org/CorpusID:214698536.

[7] Nikos Tsikoudis, Antonis Papadogiannakis, and Evangelos Markatos. "LEoNIDS: a Low-latency and Energy-efficient Network-level Intrusion Detection System". In: *IEEE Transactions on Emerging Topics in Computing* 4 (Mar. 2016), pp. 142–155. DOI: 10.1109/TETC.2014.2369958.