**Understanding Social Behaviour in Fish: Tracking, Position Analysing, Mirror Biting Tests**

Siddharth Sarwate

Ramnarain Ruia Autonomous College, MSc CS P-II

**Abstract**

Tracking and analysing the behaviour of zebrafish is essential for various scientific and practical applications, such as aquatic research, underwater robotics, behaviour research. These abstract outlines a method for tracking zebrafish and studying their behaviour using Channel and Spatial Reliability Tracker (CSRT) algorithm, a robust object tracking algorithm provided by OpenCV library.

**Introduction**

The proposed approach begins with video data acquisition of fish in their natural or control environment. CSRT tracker is initialized by selecting a region of interest (ROI) around the target fish in the first frame of the video. As the video progresses the CSRT tracker robustly tracks the fish across consecutive or subsequent frames, providing accurate position information.

Behaviour analysis is conducted by analysing the trajectories and interactions of the tracked fish. Key behaviour parameters can be generated by seeing fish mirror biting test, fish position in the environment over the time i.e., top to middle to bottom and left to centre to right and vice-versa. Furthermore, the CSRT tracker's adaptability to changes in the scale, lighting conditions, and occlusion enhances its utility for long-term fish tracking in diverse aquatic environments. This adaptability allows for the study of day and night behaviours, as well as tracking fish under varying water turbidity.

Since object is already known there's benefit of object tracking over the object detection method such as, when tracking control or toxin fish it might change the colour or a fish maybe hybrid or there can be different types of fish in the same tank or scientific research wanted to work on different species of fish itself, object detection fails to do the detection hence tracking also it requires more computational power the more objects get detected. On the other hand, object tracking allows tracking without knowing the object with less computational load.

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTER NO.** | | | **TITLE** | **PAGE NO.** |
| **1** |  |  | **Abstract** | **1** |
| **2** |  |  | **Introduction** | **1** |
| **3** |  |  | **Literature Review** | **4** |
| **4** |  |  | **Research Methodology** | **5** |
|  | **4.1** |  | **OpenCV** | **5** |
|  |  | 4.1.1 | Image Processing | **5** |
|  |  | 4.1.2 | Video Analysis | **5** |
|  |  | 4.1.3 | Machine Learning | **5** |
|  |  | 4.1.4 | Computer Vision Algorithms | **5** |
|  | **4.2** |  | **NumPy** | **5** |
|  |  | 4.2.1 | Multi-dimensional Arrays | **5** |
|  |  | 4.2.2 | Mathematical Functions | **5** |
|  | **4.3** |  | **Pandas** | **5** |
|  |  | 4.3.1 | Data Structures | **6** |
|  |  | 4.3.2 | Data Frame | **6** |
|  |  | 4.3.3 | Series | **6** |
|  |  | 4.3.4 | Data Import/Export | **6** |
|  | **4.4** |  | **Matplotlib** | **6** |
|  |  | 4.4.1 | Wide Range of Plot Types | **6** |
|  |  | 4.4.2 | Customization | **6** |
|  |  | 4.4.3 | Integration | **6** |
|  |  | 4.4.4 | Wide Adoption | **6** |
|  | **4.3** |  | **CSRT Tracker** | **6** |
|  |  | 4.3.1 | Feature Extraction | **6** |
|  |  | 4.3.2 | Model Initialization | **6** |
|  |  | 4.3.3 | Online and Offline Learning | **6** |
|  |  | 4.3.4 | Correlation Filtering | **7** |
|  |  | 4.3.5 | Object Localization | **7** |
|  | **4.4** |  | **Working** | **7** |
|  |  | 4.4.1 | Time Interval | **7** |
|  |  | 4.4.2 | Number of Objects | **7** |
|  |  | 4.4.3 | Region of Interest (ROI) Selection | **7** |
|  |  | 4.4.4 | Position Calculation | **8** |
|  |  | 4.4.5 | Counters and Timers | **9** |
|  |  | 4.4.6 | Global Timer | **9** |
|  |  | 4.4.7 | Another Timer | **9** |
|  |  | 4.4.8 | Graph | **10** |
|  |  | 4.4.9 | Data Generation | **10** |
|  | **4.5** |  | **Process Flow Diagram** | **11** |
| **5** |  |  | **Experimental Setup** | **12** |
|  | **5.1** |  | **Software Language** | **12** |
|  |  | 5.1.1 | Python 3.9 | **12** |
|  |  | 5.1.2 | OpenCV | **12** |
|  |  | 5.1.3 | NumPy | **12** |
|  |  | 5.1.4 | Pandas | **12** |
|  |  | 5.1.5 | Matplotlib | **12** |
| **6** |  |  | **Result** | **13** |
|  | **6.1** |  | **Screen Layout** | **13** |
|  | **6.2** |  | **Testing Reports** | **17** |
|  | **6.3** |  | **Selecting Best Tracker** | **17** |
|  | **6.4** |  | **Multiple ROI Selection** | **17** |
|  | **6.5** |  | **Multiple CSRT Tracking** | **17** |
|  | **6.6** |  | **Multiple Parameter Updates** | **17** |
|  | **6.7** |  | **Multiple Position Tracking** | **17** |
|  | **6.8** |  | **3D Projection and Line Graphs** | **17** |
|  | **6.9** |  | **Excel Saves** | **17** |
|  | **6.10** |  | **Functions and Parameters Passing** | **17** |
| **7** |  |  | **Conclusion** | **18** |
|  | **7.1** |  | **Findings of Project Work** | **18** |
|  | **7.2** |  | **Future Enhancement** | **18** |
| **8** |  |  | **Reference** | **19** |
|  | **8.1** |  | **Journal / Article Papers** | **19** |
|  | **8.2** |  | **Websites** | **19** |
| **9** |  |  | **Table of Figures** | **20** |

**Literature Review**

Social interactions are important behaviour factor in any human and especially in non-human species since we can understand whether specific animal feel the same as we human does i.e., fear, threat and anxiety. It also allows to find out if any diseases, stress, mental or physical illness, disorders triggered by social deficits in a species.

[1] Accessing social behaviour phenotypes in adult zebrafish: shoaling, social preference, and mirror biting tests by Allan V. Kalueff and Adam Michael Stewart and team did several researches and articles in 2012 are done by neuroscientists based on the zebrafish behaviour. Shoaling test with 2 groups of fish around 7 mins average, where four average length fish within same shoal considered. In social preference test, around 70% of the time fish remained together. In 6 minutes of biting test, fish started mirror biting around 3-4 minutes in two tests 1) Introducing fish to the mirrored tank and vice-versa.

[2] A versatile setup for measuring multiple behaviour endpoints in zebrafish by Gilbert Audira, Bonifasius P Sampurna, Stevhen Juniardi, Sung-Tzu Liang, Yu-Heng Lai, and Chung-Der Hsiao in 2018 did a research paper based on zebrafish behaviour with five vertical layers, two horizontal columns, novel tank diving test, predator avoidance, also XY coordination on natural and control environment such as ethanol (EtOH) for 96 in a single video. Materials and Methods are used such as Animal and Housing, Ethanol Treatment, Video-Tracking and Data Analysis performed in room temperature between 11:00 and 16:00.

[3] Multiple Object Trackers in OpenCV: A Benchmark OpenCV (Open-Source Computer Vision) allows us to develop computer vision related applications mainly where the image or video related tasks are involved such as video surveillance, autonomous and smart vehicles which makes live time decision while driving via camera and other sensors with the help of AI, medical treatments, and many others. In these activities real-time tracking is involved without training an actual object each and every time i.e., without image detection. So, OpenCV provides several trackers such as BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTRUN, MOSSE, CSRT. Paper is mainly based on SOT (Single Object Tracking) and MOT (Multiple Object Tracking) and object tracking is performed on four videos, fifty objects at each frame on the dataset like MOT2O. MOSSE and MEDIANLOW having potential to become real-time object tracking where number of objects can be exceeded even by 100! Other trackers don’t show this potential if the number of objects exceed 10. CSRT shows best performance for tracking followed by MIL and Boosting, where as these two are faster than CSRT but slow in accuracy and success rate.

**Research Methodology**

**OpenCV –**

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. It allows us to work on images and videos of various types such as mp3, wav, mp4 etc also with live camera vision by providing a wide range of tools, functions and algorithms that allows researchers and developers to work with visual data. It's primarily written in C++ language and does contain bindings for various programming languages including Java, Python.

**Image Processing –** OpenCV offers packages for image processing functions to do tasks like image filtering, edge detection, colour manipulation, image transformation.

**Video Analysis –** OpenCV supports video input/output, live video capture, and video analysis, doing suitable tasks like object tracking, video stabilization, and motion detection.

**Machine Learning –** OpenCV provides machine learning algorithms for tasks such as object detection, face recognition, and feature classification.

**Computer Vision Algorithms –** It includes various computer vision algorithms, including feature detection and matching, image stitching, and camera calibration. One of the most effective tracking algorithms including CSRT is provided by OpenCV.

**NumPy –**

NumPy, which stands for "Numerical Python," is an open-source Python library for numerical computing. It is one of the fundamental libraries in the Python data science ecosystem and is widely used for performing mathematical and numerical operations on large, multi-dimensional arrays and matrices. Some key features and components of NumPy include:

**Multi-dimensional Arrays –** NumPy provides a data structure called numpy.array, which is an efficient, homogeneous, and multi-dimensional array object. These arrays can have any number of dimensions and are used to represent a wide variety of data, including scalars, vectors, matrices, and higher-dimensional data structures.

**Mathematical Functions –** NumPy includes a comprehensive set of mathematical functions for performing operations on arrays. These functions include basic arithmetic operations, mathematical functions (e.g., sine, cosine), linear algebra operations, statistical functions, and more.

**Pandas –**

Pandas is an open-source data manipulation and analysis library for the Python programming language. It provides data structures and functions that simplify working with structured data, such as tabular data, time series, and more. Pandas is widely used in data analysis, data cleaning, data transformation, and data preparation tasks. Some of the key features and components of Pandas include:

**Data Structures –** Pandas provides two primary data structures, the Data Frame and the Series:

**Data Frame –** A two-dimensional table or spreadsheet-like data structure with rows and columns, similar to a SQL table or Excel spreadsheet. Each column can have a different data type.

**Series –** A one-dimensional array-like data structure that represents a single column or row of data.

**Data Import/Export –** Pandas allows you to read data from various file formats, including CSV, Excel, SQL databases, and more. You can also export data to these formats.

**Matplotlib –**

Matplotlib is a popular Python library for creating static, animated, 2D/3D, and interactive visualizations in various formats, such as charts, plots, and graphs. It provides a customizable and flexible framework for creating publication-quality figures and is widely used in scientific computing, data analysis, and data visualization.

**Wide Range of Plot Types –** Matplotlib supports a wide variety of plot types, including, histograms, pie charts, line plots, scatter plots, bar plots, , 2D/3D plots, and more.

**Customization –** Users can extensively customize the appearance of their plots by specifying attributes such as colours, markers, line styles, labels, and fonts. This flexibility allows you to create highly tailored visualizations.

**Integration –** Matplotlib can be used seamlessly with other popular Python libraries and tools, such as NumPy for numerical operations and Pandas for data manipulation. It also has built-in support for LaTeX for mathematical annotations.

**Wide Adoption –** Matplotlib is widely adopted in the scientific and data analysis communities. Many other data visualization libraries and tools are built on top of or integrate with Matplotlib.

**Channel & Spatial Reliability Tracker (CSRT) Tracker –**

**Feature Extraction –** CSRT extracts features from a target object's initial region of interest (ROI) in the first frame of the video by creating bounding box around it and getting related coordination in X, Y, w, h format. The feature captures the information about the object's appearance.

**Model Initialization –** The extracted features are used to initialize a model of the target object.

**Online and Offline Learning –** As the video progresses, in each frame CSRT continuously adapts its model based on the appearance of the object in current frame and consecutive frames. The tracker learning can be online to adapt the changes in the object’s appearance. CSRT in offline manner is suitable for scenarios where the object's appearance remains relatively consistent throughout the video sequence.

**Correlation Filtering –** The CSRT algorithm uses a correlation filter to match the learned object model with the image data in each frame. This helps in locating the object in subsequent frames.

**Object Localization –** The tracker uses the correlation response to estimate the object's position in the current frame. The bounding box continues to change its position and X, Y, w, h coordination along with the frame.

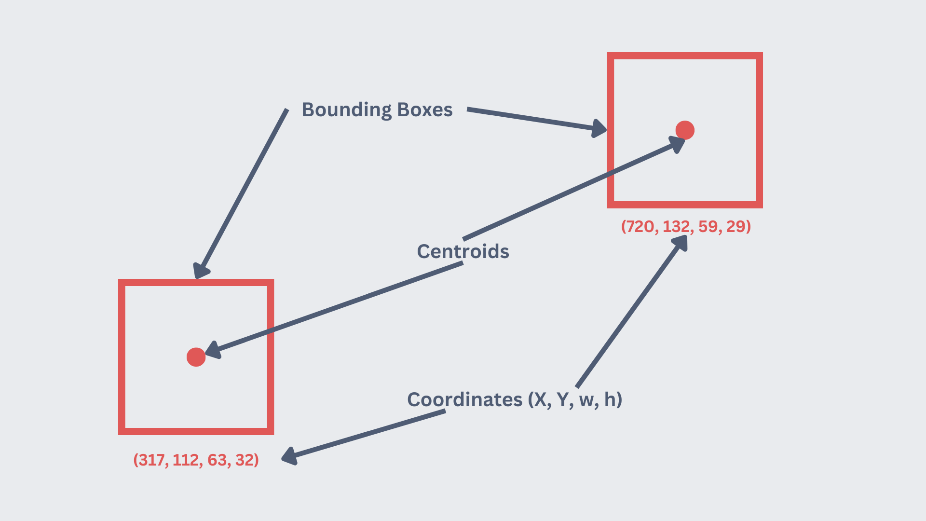


Figure 4.1: Region of Interest (ROI)

**Working –**

**Time Interval –** Select time interval to fetch location of the moving object. This will later on help to store coordinates in excel and further behaviour analysis

**Number of Objects –** Select number of objects to be tracked. This system is able to track at least one object and up to three objects flawlessly.

**Region of Interest (ROI) Selection –** Select ROI in the first frame of the video in X, Y, w, h bounding box format and get centroids C1 and C2.



Figure 4.2: Centroids Calculation

**Positions Calculation –** After tracker started tracking it’ll calculate the position of the object or multiple objects on the basis of left-mid-right and top-centre-bottom of the tank

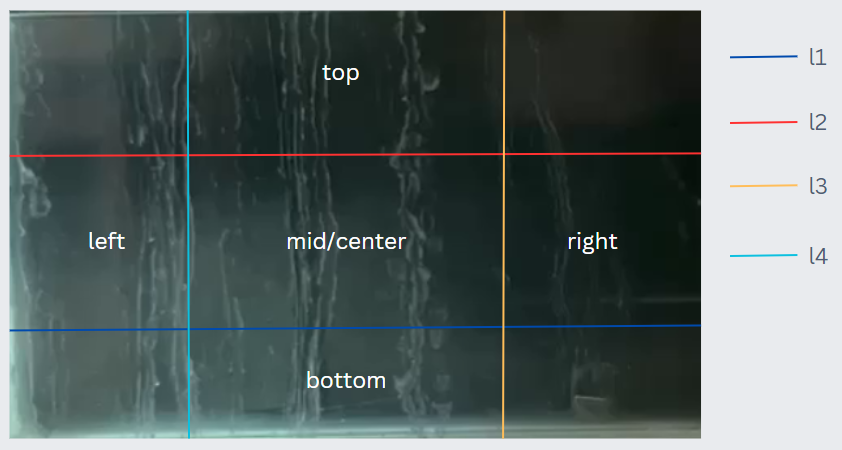


Figure 4.3: Traversing Positions

displaying of counter and timer for these positions with the help of generated C1 and C2 (centroids).

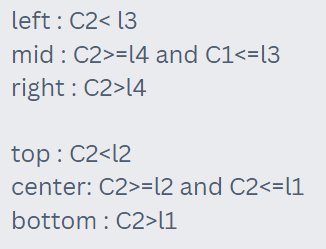


Figure 4.4: Position Calculation

Also, on the same basis it’ll show timer for left-mid-right and top-centre-bottom when object is moving in the particular area with the help of Booleans.

This will be helpful to find out tests such as mirror biting or predator avoidance test.

**Counters and Timers –**

The displayed Counters of entering of an object into specific part (left-centre-right and top-mid-bottom) and the Timer that for how much time it remained into that area will be displayed if the object itself exists.

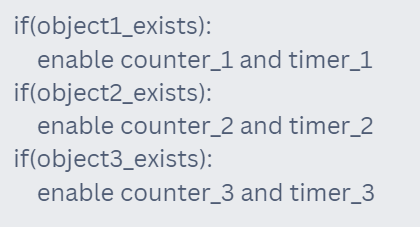


Figure 4.5: Counters and Timers Visibility

**Global Timer –** Global Timer will be started as soon as selection of objects to be tracked (ROI) gets done.To display video playback timing along with left-mid-right and top-centre-bottom time calculation.

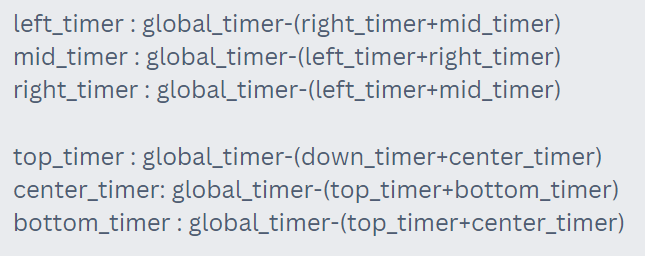


Figure 4.6: Timer Calculation

**Another Timer (XY coordinates position) –** Another Timer which will calculate XY coordinates to be tracked.By selecting interval time after which position of the object gets calculated at that moment.

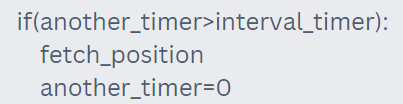


Figure 4.7: Interval Position Fetching Timer

Depending upon the number of object’s selected (at least one and up to three), the calculated positions will be stored for 2D graph view in 3D projection along with position of objects with respect to time in the line graph. Positions of objects will be stored into an excel sheet

**Graph –** First, it’ll check that how much objects exists then append the positions and store it for 2D graph generation into 3D projection and Line graph

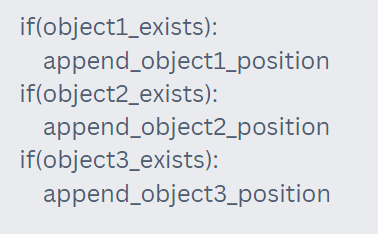


Figure 4.8: Fish Position Fetcher

**Data Generation –** On the basis of stored XY coordinates with respect to time, the graph will get plotted also further research can be done.

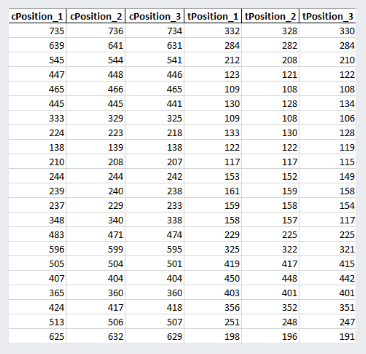


Figure 4.9: Excel (xlsx)

**Process Flow Diagram –**

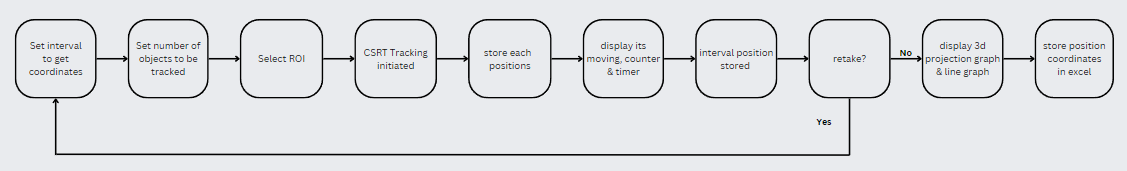


Figure 4.10: Control Flow

**Experimental Setup**

**Software Language –**

**Python 3.9 –** Python is an interpreted, high-level and general-purpose programming language. Python’s design philosophy emphasises code readability with its notable use of significant whitespace.

**OpenCV (Open-Source Computer Vision Library) –** OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel. The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

**NumPy –** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin into Numeric, with extensive modifications, NumPy is open-source software and has many contributors.

**Pandas –** Pandas is a software library developed by Wes McKinney written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

**Matplotlib –**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, Qt. Matplotlib was originally written by John D. Hunter. Since then, it has had an active development community and is distributed under a BSD-style license.

**Result**

**Screen Layout –**

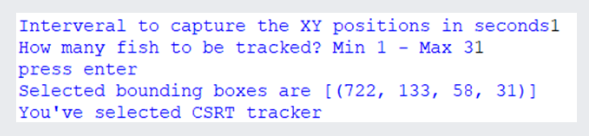


Figure 6.1: Time Interval, Number of Objects and Tracker Type Parameters

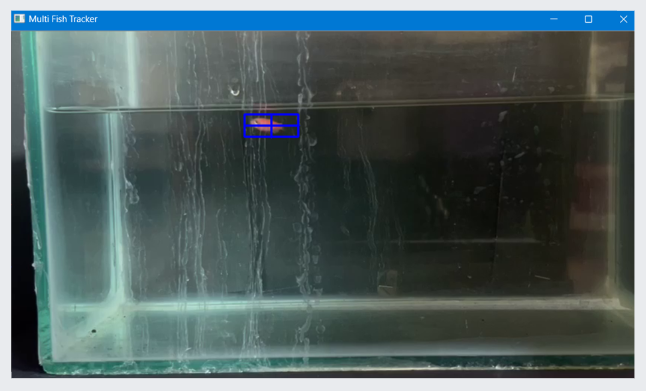


Figure 6.2: Region of Interest (ROI) Selection

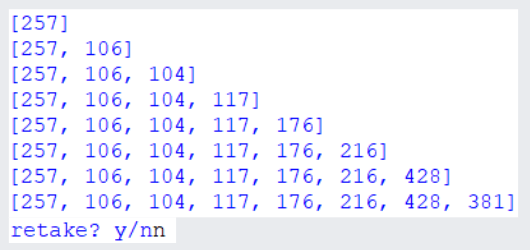


Figure 6.3: Position after intervals and Retake Option

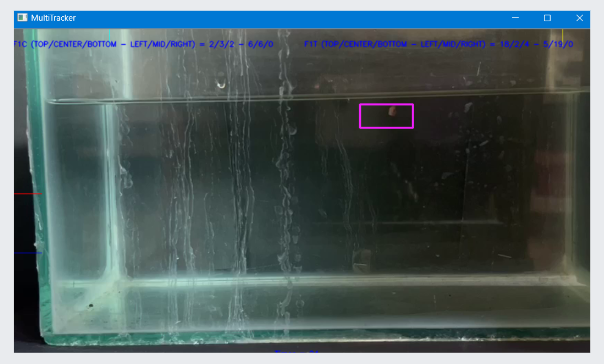


Figure 6.4: Object Tracking (Natural Environment)

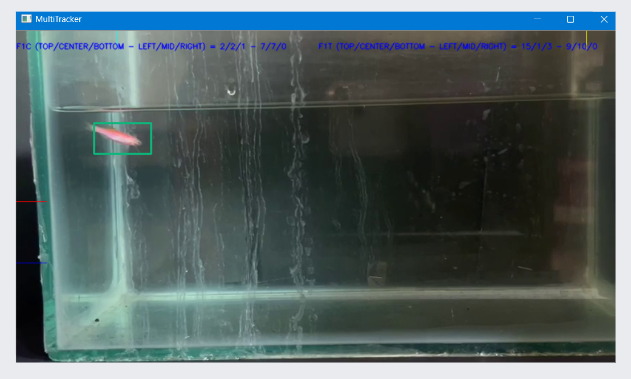


Figure 6.5: Object Tracking (Control Environment)

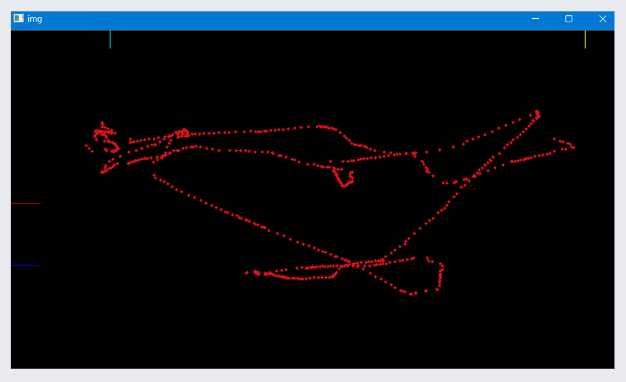


Figure 6.6: Object Path Tracking (Natural Environment)

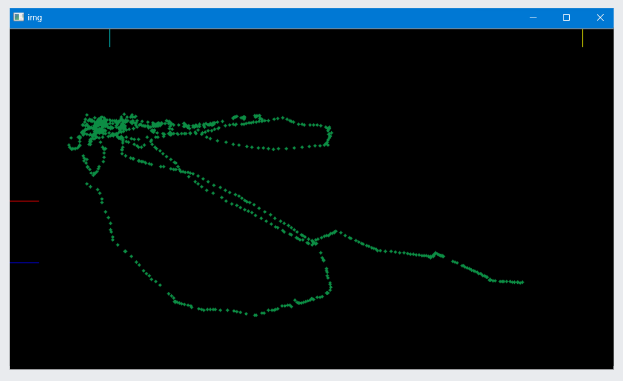


Figure 6.7: Object Path Tracking (Control Environment)

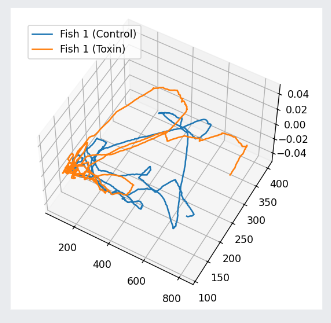


Figure 6.8: 2D Graph in 3D Projection

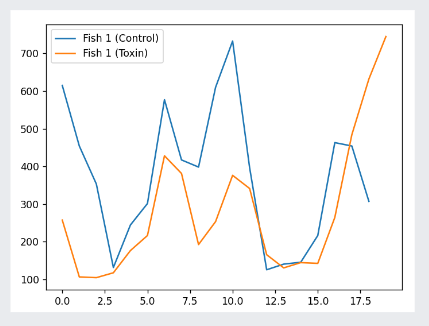


Figure 6.9: Line Graph (Position with respect to Time)

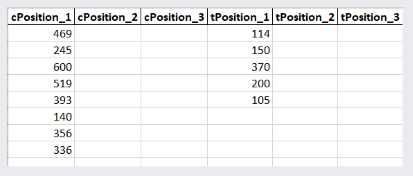


Figure 6.10: Data Saving into Excel Format (xlsx)

**Testing Reports –**

**Selecting Best Tracker –**

**Status:** Success

Tested multiple trackers including BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTRUN, MOSSE, CSRT.

TLD tracker don’t work at all, MOOSE AND MEDIANFLOW found not suitable for current scenarios whereas MIL works better than BOOSTING but tank needs to be clean and image has to be clear. CSRT performs very well among all with around 75%-80% accuracy and success rate.

**Multiple ROI Selection –**

**Status:** Success

At least one object and up to three objects can be selected.

**Multiple CSRT Tracking –**

**Status:** Success

At least one object and up to three objects can be tracked along with tracking path.

**Multiple Parameter Updates –**

**Status:** Success

Normal object arrays and interval-time arrays working fine.

**Multiple Position Tracking –**

**Status:** Success

Top-Mid-Bottom and Left-Centre-Right Counter with Timer working without delay along with global timer working.

**3D Projection and Line Graphs –**

**Status:** Success

3D projection graph working along with line graph with multiple objects after getting coordinates from the tracker working.

**Excel Saves –**

**Status:** Success

Interval-time arrays for multiple objects will be stored into excel for line graph working.

**Functions and Parameters Passing –**

**Status:** Success

Functions and Parameter for natural and control as well as retaking working.

**Conclusion**

**Findings of Project work –**

Despite lot of short comings and the disadvantages of the object tracking over the object detection it is certainly an innovative way to track an object and see its behaviour, unlike in object detection, it has to have prior knowledge about an object which is not the scenario in object tracking. Multiple fish, hybrid or dyed fish can be tracked without training a model. Scientific research can be done based on this by noticing the behaviour and generated data from the tracking such as biting test, predator avoidance, anxiety of a fish.

**Future Enhancement –**

More objects for tracking along with detection of an object.

Lost tracking recovery from a specific point.

Software based approach for UI enhancement.

Various type of graphs.

3D Projection with the help of two cameras placing simultaneously.

Various animals apart from fish.

**Reference**

**Journal / Article Papers –**

A Versatile Setup for Measuring Multiple Behaviour Endpoints in Zebrafish<https://www.researchgate.net/publication/328796417_A_Versatile_Setup_for_Measuring_Multiple_Behavior_Endpoints_in_Zebrafish>

Assessing Social Behaviour Phenotypes in Adult Zebrafish: Shoaling, Social Preference, and Mirror Biting Tests

<https://www.researchgate.net/publication/257931790_Assessing_Social_Behavior_Phenotypes_in_Adult_Zebrafish_Shoaling_Social_Preference_and_Mirror_Biting_Test>

Multiple Object Trackers in OpenCV: A Benchmark

<https://arxiv.org/pdf/2110.05102.pdf>

**Website –**

[**https://opencv.org**](https://opencv.org/)

[**https://learnopencv.com**](https://learnopencv.com/)

[**https://broutonlab.com**](https://broutonlab.com)

[**https://github.com**](https://github.com)

[**https://www.youtube.com**](https://www.youtube.com)

[**https://stackoverflow.com**](https://stackoverflow.com)

[**https://www.geeksforgeeks.org**](https://www.geeksforgeeks.org)

**Table of Figures**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITILE** | **PAGE NO.** |
| 4.1 | Region of Interest (ROI) | 7 |
| 4.2 | Centroids Calculation | 8 |
| 4.3 | Traversing Positions | 8 |
| 4.4 | Position Calculation | 8 |
| 4.5 | Counters and Timers Visibility | 9 |
| 4.6 | Timer Calculation | 9 |
| 4.7 | Interval Position Fetching Timer | 10 |
| 4.8 | Fish Position Fetcher | 10 |
| 4.9 | Excel (xlsx) | 10 |
| 4.10 | Control Flow | 11 |
| 6.1 | Time Interval, Number of Objects and Tracker Type Parameters | 13 |
| 6.2 | Region of Interest (ROI) Selection | 13 |
| 6.3 | Position after Intervals and Retake Option | 13 |
| 6.4 | Object Tracking (Natural Environment) | 14 |
| 6.5 | Object Tracking (Control Environment) | 14 |
| 6.6 | Object Path Tracking (Natural Environment) | 14 |
| 6.7 | Object Path Tracking (Control Environment) | 15 |
| 6.8 | 2D Graph in 3D Projection | 15 |
| 6.9 | Line Graph (Position with respect to Time) | 15 |
| 6.10 | Data Saving into Excel Format (xlsx) | 16 |