International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 12 | Dec 2024 www.irjet.net p-ISSN: 2395-0072

OBJECT TRACKING AND VISULIZATION USING OPENCV AND UNITY 3D

DEEP RAJESH UTHALE[1], DEVASHISH MAYUR POTNIS[2], PROF. SUPRIYA BALOTE[3]

[1,2]Student, Dept. Artificial Intelligence & Machine Learning Engineering, PES's Modern College of Engineering, Pune Maharashtra, India
[3]Assistant Professor, Dept. Artificial intelligence & Data Science Engineering, PES's Modern College of Engineering Pune Maharashtra, India

---

Abstract - *This paper presents the development of a real-time 3D ball tracing system utilizing OpenCV for computer vision and Unity for 3D rendering. The system tracks the movement of a ball in real-time by processing video input through OpenCV, which identifies the ball's position using image processing techniques such as color segmentation and contour detection.[1] [2] The extracted data is then transferred to Unity, where it is used to render the ball's movement in a 3D virtual environment. This integration allows for accurate tracking and dynamic visualization, making the system ideal for applications in sports analysis, augmented reality, virtual reality and robotics.[4] The proposed approach demonstrates significant potential for real-time motion analysis and interactive visualization in various domains.*

***Key Words*: OpenCV, Unity, 3D, Ball tracking, Real-time, Virtual environment, Dynamic visualization.**

1. INTRODUCTION

This research focuses on the development of a real-time 3D object tracking system by integrating OpenCV for computer vision and Unity for 3D visualization. The system processes video input to track an object, such as a ball, using techniques like color segmentation and contour detection. The tracked data is transmitted to Unity using UdpClient [5], enabling dynamic visualization of the object's movement in a virtual 3D environment. This project has significant applications in sports analysis, robotics, and augmented reality, providing a seamless connection between physical and virtual spaces for enhanced interaction and analysis.[4]

The implementation utilizes computer vision techniques provided by OpenCV, a widely-used open-source library for image and video processing. Python offers simplicity and extensive support for computing and it serves as a programming language. The project demonstrates the integration of these tools to achieve real-time 3D tracking, highlighting the potential for creating sophisticated virtual environments that interact seamlessly with real-world dynamics.[2]

This research contributes to the field by providing a framework for real-time object tracking in virtual spaces, paving the way for advancements in immersive technologies and automated analysis systems. The methodologies and findings presented can serve as a foundation for further exploration into more complex scenarios involving multiple objects, fast motions and intricate interactions within virtual environments.

1.1 Context

The growing demand for interactive and immersive technologies has driven advancements in the fields of computer vision and 3D visualization. Real-time object tracking and visualization play a critical role in applications ranging from augmented reality and robotics to sports analytics and gaming. Accurately bridging the physical and virtual worlds requires systems capable of detecting, tracking, and representing objects in real-time with precision and efficiency.[2] [4,7]

OpenCV, a widely used open-source computer vision library, provides robust algorithms for image processing and object tracking [1]. Unity, a leading real-time 3D development platform, offers powerful tools for rendering and visualizing virtual environments.[6] The combination of these technologies enables the creation of systems that not only track real-world objects but also seamlessly integrate their movement into virtual environments.

This project addresses the need for such integration by utilizing OpenCV for real-time object tracking and Unity for dynamic 3D visualization. By combining these tools, the system provides an effective solution for applications that require accurate motion tracking and interactive visualization, contributing to the advancement of technologies in various domains such as augmented reality, robotics, and sports performance analysis.[7]

1.2 Problem Definition

Real-time object tracking and visualization face challenges such as low accuracy, latency, and limited integration between physical tracking and virtual representation. These issues impact applications in fields like sports analytics, robotics, and augmented reality, where precise and dynamic interaction is essential. This project addresses these challenges by utilizing OpenCV for accurate tracking and Unity for seamless 3D visualization, providing a reliable and efficient solution for real-time object tracking and its virtual representation.[3]

2. OBJECTIVES

- **Develop a Real-Time Object Tracking System:** Utilize OpenCV to accurately detect and track the position and movement of a 3D object, such as a ball, in video input.[2]
- **Integrate Data Transmission:** Establish efficient communication between OpenCV and Unity using networking protocols like UDP/TCP for seamless data transfer.[5]
- **Create a Dynamic 3D Visualization:** Use Unity to render the tracked object's motion in a virtual 3D environment, ensuring real-time synchronization with physical movement.[6]
- **Enhance System Accuracy and Responsiveness:** Minimize tracking errors and latency to improve the reliability and real-time performance of the system.
- **Enable Practical Applications:** Demonstrate the system's applicability in fields such as sports analytics, robotics, and augmented reality for enhanced interactivity and analysis.[7]

3. METHODOLOGY

3D Ball Tracking in Virtual Environment project involves a series of systematic steps to achieve real-time object tracking and representation. The process begins with video input acquisition, where frames are captured in real-time from a camera or video source. These frames are pre-processed by applying noise reduction techniques, such as Gaussian noise/blur, and converting the color space from BGR to HSV to enable efficient color-based detection. The next stage involves creating a mask to isolate the ball based on its unique color characteristics, followed by contour analysis to identify and localize the ball within the frame. To estimate the ball's position in three-dimensional space, depth information is derived using stereo vision or depth sensors, enabling the computation of accurate 3D coordinates.[8]

Once the ball's position is determined, motion analysis techniques are used to predict its future trajectory. This predictive capability enhances the system's responsiveness and realism, particularly in dynamic scenarios. Finally, the ball's tracked position and predicted trajectory are integrated into a virtual environment, ensuring that the simulation accurately reflects real-world dynamics. This methodology emphasizes modularity and precision, making it adaptable for various applications, including virtual reality, sports analytics, and interactive gaming.[7]

4. ARCHITECTURE

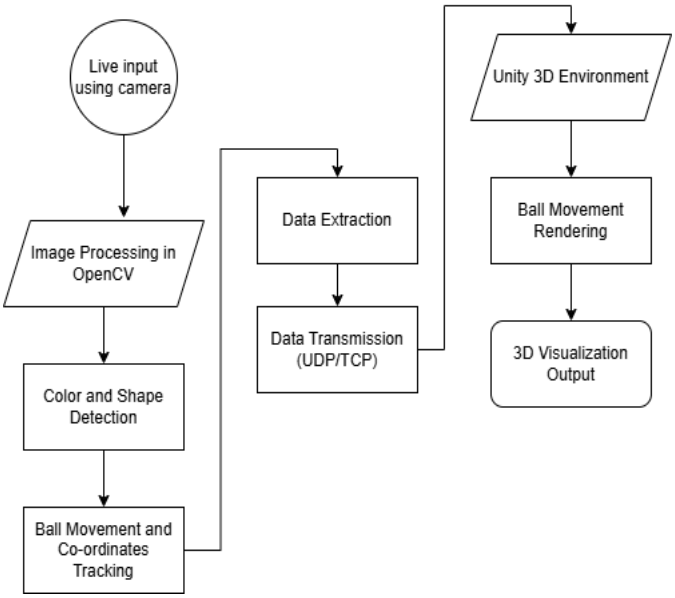

Diagram -1: Architecture

- **Input Acquisition**: This module captures real-time video frames from a camera or video file. The frames serve as the primary data source for detecting and tracking the ball within the scene.
- **Pre-processing**: In this stage, the acquired frames undergo pre-processing to enhance the quality and suitability for analysis. Techniques such as Gaussian blur are applied to reduce noise, and color space conversion (e.g., BGR to HSV) is performed to facilitate color-based object detection.[3]
- **Ball Detection**: Utilizing color-based segmentation, the system isolates the ball from the background. By defining specific color ranges in the HSV color space, a mask is created to highlight the ball, allowing for its identification within the frame.[3]
- **3D Position Estimation**: To determine the ball's position in three-dimensional space, depth estimation techniques are employed. This may involve stereo vision setups or depth sensors to calculate the distance of the ball from the camera, enabling accurate 3D localization.
- **Trajectory Prediction**: Once the 3D position is established, the system predicts the ball's trajectory using motion analysis. This prediction is essential for applications requiring anticipation of the ball's movement, such as interactive gaming or simulations.[8]
- **Virtual Environment Integration**: The final component involves mapping the predicted

trajectory and current position of the ball into a virtual environment. This integration ensures that the virtual representation accurately reflects the real-world dynamics of the ball, providing an immersive experience for the user.[6]

5. IMPLEMENTATION

1. **Environment Setup:**
   - Installed Dependencies: Ensuring Python is installed, along with necessary libraries such as OpenCV, Socket.
   - Configured Hardware: Set up a camera or video input device to capture real-time footage.
   - Captured Video Frames: Utilized OpenCV to access the video stream and read frames sequentially.[1]
2. **Pre-processing:**
   - Color Space Conversion: Converted frames from BGR to HSV color space to facilitate color-based object detection.
3. **Ball Detection:**
   - Color Thresholding: Defined HSV color ranges corresponding to the ball's color and create a mask to isolate the ball.
   - Contour Detection: Identified contours in the masked image to locate potential ball candidates.
   - Contour Filtering: Filtered contours based on size and shape to accurately identify the ball.[3]

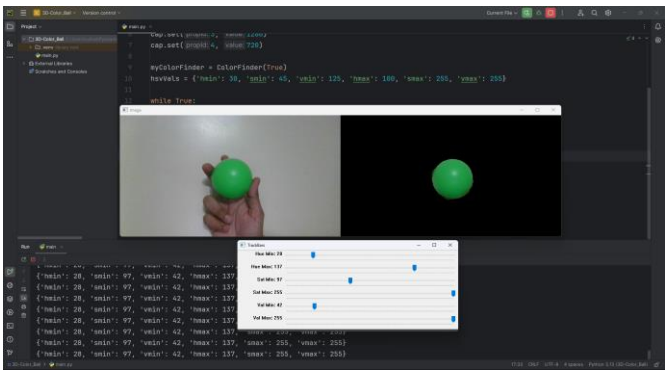

Fig -1: Ball Detection

4. **3D Position Estimation:**
   - Coordinate Mapping: Combined 2D positional data with depth information to compute the ball's 3D coordinates.
5. **Trajectory Prediction:**
   - Motion Analysis: Analyzed the ball's movement over time to predict its future positions.
   - Model Implementation: Applied mathematical functions, such as vectors, to enhance prediction accuracy.

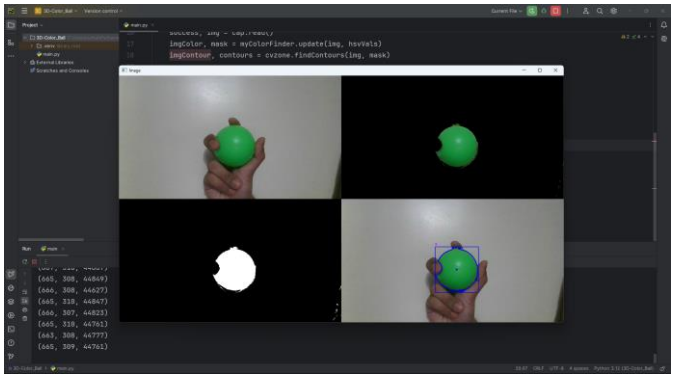

Fig -2: Trajectory Prediction

6. **Virtual Environment Integration:**
   - Real-Time Rendering: Mapped the ball's 3D coordinates and predicted trajectory into the virtual environment to simulate its motion accurately.[6]
7. **Testing and Validation:**
   - System Calibration: Adjusted parameters to ensure the system accurately tracks and represents the ball's movement.
   - Performance Evaluation: Tested the system under various conditions to assess robustness and reliability.

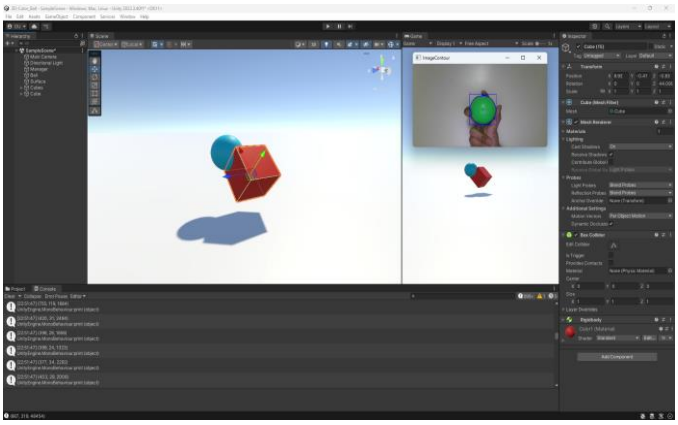

Fig -3: Testing and Validation

6. RESULTS

The 3D Ball Tracking in Virtual Environment project, implemented using OpenCV and Python, achieved significant milestones in real-time object tracking and virtual simulation. The system effectively detected and tracked a colored ball in real-time, maintaining high accuracy even during rapid movements.

By employing depth estimation techniques, the project accurately calculated the ball's three-dimensional coordinates, enabling precise mapping within a virtual environment. The integration of motion analysis algorithms allowed for effective prediction of the ball's future positions, enhancing the realism and interactivity of the virtual simulation.

These outcomes demonstrate the project's potential applications in areas such as virtual reality simulations, sports analytics, and interactive gaming, where real-time 3D object tracking is essential.[4][7]

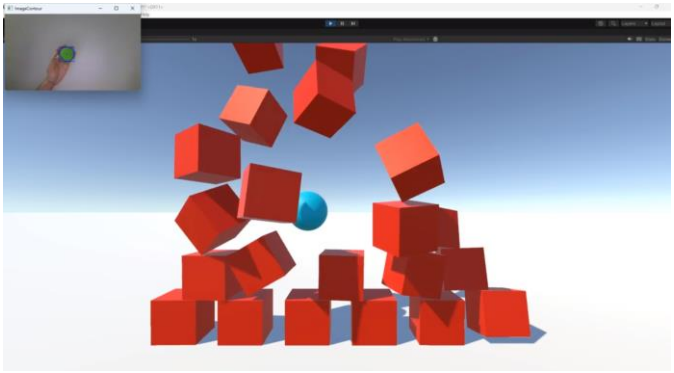

Fig -4: Output

7. CHALLENGES AND LIMITATIONS

- Ensuring accurate object detection and motion tracking under changing lighting conditions.
- Managing real-time data transmission with low latency.
- Handling occlusion or obstruction of the tracked object.
- Achieving seamless integration between OpenCV and Unity.
- Addressing performance issues for large-scale or complex 3D environments.
- Making sure it is compatible across different hardware and software platforms.
- Balancing computational efficiency with tracking accuracy.

8. FUTURE DIRECTIONS

The future scope of the 3D object tracking and visualization system presents numerous opportunities for further enhancement and application. One promising direction is the incorporation of advanced object detection algorithms, such as deep learning models, which could improve accuracy and enable the system to track more complex objects in diverse environments. Additionally, expanding the system to handle multiple objects tracking simultaneously could unlock new possibilities in fields like sports analytics, robotics, and interactive gaming.[4]

There is also potential for deeper integration with augmented reality (AR) technologies, allowing for more immersive, real-time interactive experiences in applications such as gaming, education, and training simulations. Furthermore, optimizing the system's communication to reduce latency and improve precision would be crucial for high-speed tracking in areas like robotics or real-time sports analysis.[7]

9. CONCLUSIONS

This project successfully demonstrates the integration of OpenCV and Unity for real-time 3D object tracking and visualization. By leveraging OpenCV's computer vision capabilities and Unity's 3D rendering power, the system accurately tracks and visualizes the movement of objects in real time, bridging the gap between physical and virtual spaces. The proposed system has shown significant potential in enhancing applications in sports analytics, robotics, and augmented reality, offering valuable insights through dynamic and interactive visualizations. Despite challenges in maintaining accuracy and minimizing latency, the project provides a foundation for future advancements in real-time motion tracking and virtual representation, contributing to the development of more immersive and responsive systems in various domains.

10. REFERENCES

[1] Bradski, G. (2000). "The OpenCV Library." Dr. Dobb's Journal of Software Tools. Source: opencv.org

[2] Kari Pulli (NVIDIA), Anatoly Baksheev (NVIDIA), Kirill Kornyakov (NVIDIA), Victor Eruhimov (NVIDIA). "Realtime Computer Vision with OpenCV" in Communications of the ACM. (June 2012)

[3] https://pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/

[4] https://towardsdatascience.com/ball-tracking-in-volleyball-with-opencv-and-tensorflow-3d6e857bd2e7

[5] https://learn.microsoft.com/en-us/dotnet/api/system.net.sockets.udpclient?view=net-6.0

[6] https://docs.unity3d.com/Manual/UnityManual.html

[7] Lee, J., & Lee, K. (2018). "A Study of Augmented Reality Applications in Sports Training." International Journal of Sports Science & Coaching, 13(5), 777-784. DOI: 10.1177/1747954118803269

[8] https://medium.com/@hrdeejay18/ball-tracking-and-prediction-using-opencv-and-cvzone-478b3557f413