Vision-based Robust UAV Navigation with Fiducial Markers in CoppeliaSim

Sourav Raxit
ID:2637757

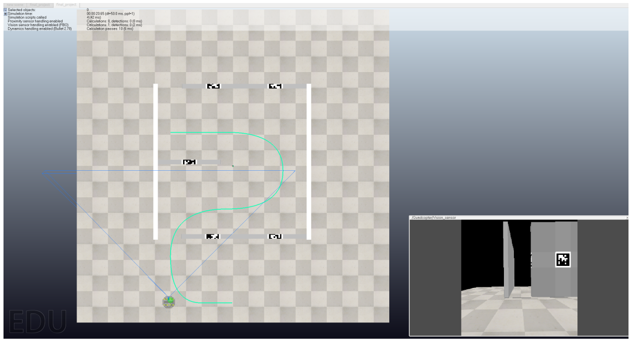
I. INTRODUCTION

The capabilities of visual fiducial markers extend beyond just augmented reality to also enable important applications in robotic navigation. Robots empowered with fiducial marker tracking can precisely determine their position and orientation relative to the camera viewing the markers. This allows robots to localize themselves in the camera's field of view without relying on GPS or other external positioning systems. Nonetheless, the reliability of detecting and differentiating multiple fiducial markers simultaneously is critical for accurate and continuous robot pose estimation.

As the robot moves through the simulated environment in CoppeliaSim, the algorithm must track each marker transition seamlessly, correlating the updated camera perspective to the robot's movement. Existing fiducial marker systems rely on hand-crafted image processing to detect black-and-white patterns like checkerboards [1]. More recent systems have explored color and topological patterns [2], [3] to improve accuracy and robustness. However, pose estimation from just corner points remains limited under noise or motion blur.

Recent deep learning-based works [1], [4], [5] show improved detection and pose estimation but have only been evaluated on simulated or limited real data. Despite achieving relatively higher accuracy than classical approaches, these methods exhibit significantly slower detection speeds compared to other marker detection systems.

The core issue with current fiducial marker-based localization systems is that they fail to generalize well to the complexities of simulated indoor navigation for robots in CoppeliaSim. This leads to problems with robustness and adaptability when deployed in practice. Estimating a robot's state accurately presents challenges when detecting markers in isolation. The noisy, fragmented observations obtained from individual markers make it difficult to fuse the detection results into a unified, coherent pose estimate. To design highly robust robot localization systems, it is critical to consider the challenges of multi-marker detection across diverse simulated conditions. In this work, we propose YoloTag, which harnesses the power of state-of-the-art deep learning object detection techniques through the integration of the high-performing YOLOv8 model. YoloTag provides an efficient machine learning architecture that enables robust multi-marker detection and 3D pose estimation across varied simulated imagery from robots operating in demanding conditions in CoppeliaSim, while meeting real-time performance requirements. By training YoloTag on this extensive dataset of simulated robot imagery, it can learn to generalize effectively and handle various complexities encountered during simulated deployment. These include dealing with varying marker poses from multiple perspectives, tracking landmarks as the robot moves and articulates, handling changing illumination conditions, overcoming partial occlusions, and being resilient to sensor noise.



```
Selected objects:                                 0
Simulation time:                                  00:00:20.65 (dt=50.0 ms, ppf=1)
Simulation scripts called                         4 (42 ms)
Proximity sensor handling enabled                 Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)              Calculations: 1, detections: 0 (2 ms)
Dynamics handling enabled (Bullet 2.78)           Calculation passes: 10 (6 ms)
```

Fig. 1: An unmanned aerial vehicle (UAV) follows an trajectory. It uses an onboard camera system and we are recording vision sensor data from this simulation environment.

YoloTag's efficient, streamlined, end-to-end deep learning architecture seamlessly fuses the information from multiple detected markers into a coherent joint 3D pose estimate. This provides precise, continuous self-localization capabilities tailored specifically for robust simulated robotic operations in CoppeliaSim.

Our key contributions are:

• Employing YOLO v8 for rapid landmark detection and pose estimation, surpassing current methods in processing speed.

• Leveraging multi-marker detection outcomes under various simulated scenarios to accurately detect and estimate poses.

• Addressing perception noise by employing a Butterworth filter to refine the estimated trajectory.

• Thoroughly assessing the system on simulated robotic platforms in indoor settings in CoppeliaSim, confirming its robustness, accuracy, and suitability for simulated applications.

II. Related Works

In this section, we explore the key related work in vision-based localization methods, the evolution of fiducial marker designs and algorithms for robust detection, and the use of fiducial markers for enabling autonomous navigation in ground and aerial vehicles.

Vision-based Localization: Vision-based localization involves estimating the location and orientation of a robot using cameras as primary sensors. Two main approaches are Relative Visual Localization (RVL) and Absolute Visual Localization (AVL). RVL techniques like Visual Odometry (VO) and Visual Simultaneous Localization and Mapping (VSLAM) incrementally estimate a robot's ego-motion and map unexplored environments by tracking visual features across frames. However, they suffer from drift over long durations. Absolute Visual Localization (AVL) aims to achieve drift-free global localization by matching imagery to geo-referenced maps, but can have limited precision.

Fiducial Marker-based Localization: Fiducial markers have enabled precise camera tracking for augmented/virtual reality and robotics. Designs like ARToolKit[6], AprilTag[7], [8], and ChromaTag[9] provide robust detection under challenges like occlusion and lighting variations. Recent work has focused on improving marker robustness to distortions through techniques like deep neural networks and alternate sensing modalities like LiDAR. Novel applications include underwater markers for navigation and end-to-end learned marker generation and detection. Algorithmic improvements have also enhanced detection efficiency and scalability. However, existing fiducial marker-based localization still faces limitations in detection speed, accuracy, and false positive rejection, particularly under challenging conditions like low resolution, occlusion, uneven lighting, and perspective distortion.

Autonomous Navigation with Fiducial Markers: Ground robots and unmanned aerial vehicles (UAVs) can leverage fiducial markers as visual landmarks for localization and mapping. By detecting markers at known locations, they can estimate their pose without external positioning systems. However, practical limitations exist like marker installation requirements and sensor constraints on UAVs.

III. Realtime Fiducial Marker Detection using Deep Learning

The realtime fiducial marker detection architecture is based on YOLOv8, which adopts an anchor-free framework, directly predicting bounding boxes within an input image I, eliminating the need for anchor box initialization and non-maximum suppression. YOLOv8 is composed of three primary components: the Backbone layer, Neck layer, and Head layer. The Backbone forms the base feature extractor, utilizing a modified CSPDarkNet-53 network. It performs 5 downsampling operations on the input image to create a multi-scale feature pyramid with levels P1 to P5. The Neck module adopts a dual-stream feature pyramid network combining FPN and PAN structures. The FPN takes a top-down approach, upsampling and merging deep features with lateral outputs from the backbone at each scale. The PAN takes a bottom-up approach, downsampling low-level features and merging them with FPN outputs at each scale. The Head contains two decoupled branches for classification and bounding box regression. YOLOv8 is trained by minimizing the total loss:

$$\mathcal{L}\text{total} = \frac{1}{N}\left(\mathcal{L}\text{conf}(p, y) + \alpha\mathcal{L}_{\text{box}}(p, \hat{p}, b, \hat{b})\right), \quad (1)$$

where N is the number of markers, α balances classification and regression, y is the ground truth label, b is the ground truth bounding box, $\hat{b}$ is the predicted bounding box, p is the ground truth probability distribution, and $\hat{p}$

is the predicted probability distribution. The classification loss is binary cross-entropy:

$$\mathcal{L}\text{conf}(p, y) = -\frac{1}{N} \sum i = 1^N \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right] \tag{2}$$

The regression loss optimizes distribution focal loss and CIoU loss:

$$\mathcal{L}\text{box} = \lambda\text{dfl} \cdot \text{DFL}(p_i, \hat{p}i) + \lambda\text{ciou} \cdot \text{CIoU}(b_i, \hat{b}_i), \tag{3}$$

IV. 4D Pose Estimation for quadrotor UAVs

Let $L = l_1, l_2, \dots, l_n$ be the set of n known markers, $\mathbf{z} = \mathbf{z}i^j i = 1^m$ be the m detected markers with four corner points $\hat{b}_i$ such that $j = 1, \dots, 4$, and $\mathbf{x} = x, y, z, \theta$ be the 4D pose of the UAV. The goal is to estimate $\mathbf{x}$ given L and $\mathbf{z}$, formulated as maximum likelihood estimation:

$$\mathbf{x}^* =_{\mathbf{x}} p(\mathbf{x}|L, \mathbf{z}) =_{\mathbf{x}} p(\mathbf{z}|\mathbf{x}, L) p(\mathbf{x}) \tag{4}$$

The perspective projection function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ transforms 3D world coordinates $l_i^j = x_i^j, y_i^j, z_i^j$ to 2D image coordinates $\mathbf{z}_i^j$:

$$\pi(l_i^j) = K \times \left[\frac{x_i^j}{z_i^j}, \frac{y_i^j}{z_i^j}, 1\right], \tag{5}$$

where K encapsulates the camera's intrinsic parameters. The EPnP algorithm computes $\mathbf{x}^*$ by minimizing the reprojection error:

$$\mathbf{x}^* = \min_{R,T} \sum_{i=1}^{m} \sum_{j=1}^{4} w_{ij} \cdot \|\mathbf{z}_i^j - \pi(Rl_i^j + T)\|^2, \tag{6}$$

where R computes θ, $T = [x, y, z]$, and w_{ij} are optional weights. Each additional landmark detection provides more information:

$$p(\mathbf{x}|L, \mathbf{z}) \propto p(\mathbf{z}_1^j|\mathbf{x}) p(\mathbf{z}_2^j|\mathbf{x}), \dots, p(\mathbf{z}_m^j|\mathbf{x}), \tag{7}$$

leading to better accuracy and precision as individual measurement errors get averaged out.

V. NOISE SUPPRESSION

The combination of YOLOv8 and the EPnP algorithm yields noisy state estimates, which can lead to substantial trajectory tracking errors over time. To obtain smooth state estimates, this noise must be filtered out.

To reduce the impact of noise on the state estimate, a Butterworth lowpass filter is implemented, which provides a maximally flat magnitude response in the passband, enabling smooth filtering around a cutoff frequency ω_c. The Butterworth filter's gradual roll-off after ω_c steadily attenuates higher frequencies where sensor noise typically resides.

To design the Butterworth filter, the time-domain signals are first converted to the frequency domain using the Fast Fourier Transform (FFT). This transformation facilitates a clearer understanding of signal characteristics and aids in subsequent filter design. The transfer function $H(s)$ in the frequency domain is given by:

$$H(s) = \frac{\omega_c}{\sum_{k=0}^{n_s} a_k s^k} \tag{8}$$

where $s = \sigma + j\omega$ represents the complex frequency, n_s is the filter order, and the coefficients a_k are determined using a recursion formula.

With known coefficients a_k, the goal is to find a new cutoff frequency ω_c in the Butterworth polynomial $B_n(s)$ based on signal characteristics, to retain true localization content while suppressing noise:

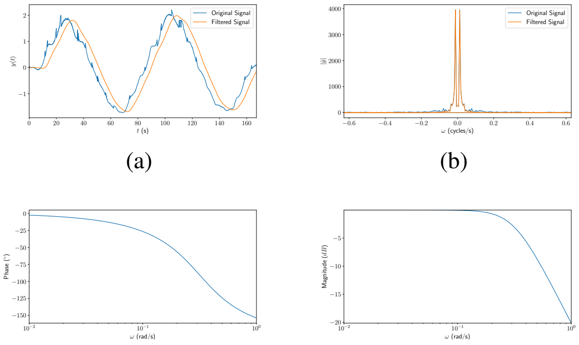
$$B_n(s) = \sum_{k=0}^{n} a_k \left(\frac{s}{\omega_c}\right)^k = \sum_{k=0}^{n} \frac{a_k}{{\omega_c}^k} s^k. \tag{9}$$

Figure 2 provides a comparative analysis of the raw and filtered signals, as well as their Fourier transformed representations and Bode magnitude and phase plots of the Butterworth filtered signal.

Specifically, Fig. 1a displays the original and filtered signals side-by-side, clearly exhibiting the noise reduction achieved, along with the resulting phase shift and decreased amplitude from the filtering process. Fig. 1b reveals how the frequency components change post-filtering, aiding in selecting appropriate cutoff frequencies for the filter design.

The Bode plots in Figs. 1c and 1d showcase how the filter passes certain frequencies unchanged while progressively attenuating others, visualizing key attributes like passband, roll-off, overall frequency response, and phase changes across frequencies.

This thorough analysis highlights the effectiveness of the Butterworth filter in smoothing trajectories and reducing noise in tag detections, underscoring its utility in enhancing data quality for trajectory analysis and related applications. The main drawback observed is the time delay in the output signal, mitigated by using a 2nd-order Butterworth filter and leveraging a 30 frames per second (fps) input frame rate and powerful computational unit.



(a) (b)

(c) (d)

Fig. 2: The Blue line depicts a raw trajectory from the EPnP algorithm while the orange line depict the corresponding filtered trajectory from the Butterworth filter in Fig. 2a. Fig.2b demonstrates how a Fast Fourier Transform (FFT) is utilized to determine an appropriate cutoff frequency for the Butterworth filter. The phase response of the filter is characterized in the phase plot (Fig. 2c), showing the phase shift introduced across frequencies —specifically, a -150° phase shift at 1 rad/s. Similarly, the magnitude response is shown in Fig. 2d, with the filter inducing a -20 dB attenuation at 1 rad/s. Analyzing these frequency responses aids in understanding the behavior of the filter and how it impacts the trajectory data.

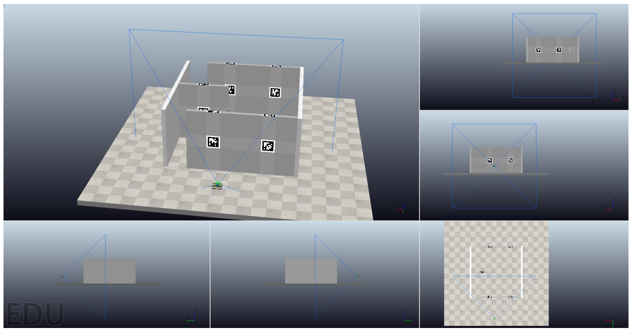


Fig. 3: Experimental Setup

VI. Experiments

A. Setup

The experiments were conducted using the CoppeliaSim simulation software on a desktop computer equipped with an Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz and an NVIDIA Quadro P2200 GPU, which has a GPU memory of 5 GB, along with 32 GB of RAM. The system operated on Ubuntu 20.04 LTS integrated with ROS Noetic. We created a simulated environment within CoppeliaSim, consisting of a 10m x 10m floor and walls that are 2.4 meters high. This setup was used to conduct various simulation experiments to evaluate the performance of our vision-based localization algorithm in a controlled virtual environment

B. Dataset Generation

To obtain training, validation, and testing images, a simulated environment was created in CoppeliaSim, measuring 10m x 10m, with walls forming a contour within the environment. Six fiducial markers, each bearing a unique ID, were placed in static positions on the walls at a height of 1m. The markers were positioned at varying distances from each other to ensure the UAV could differentiate between the landmarks when detected in the camera's field of view, enabling the estimation of its state from the tags. Multiple experiments were conducted to generate the dataset, capturing onboard camera images at 30 frames per second (FPS) with a resolution of 256 x 256 pixels. Ground truth trajectories for all experiments were recorded simultaneously within the simulated environment. The training image sequences and ground truth trajectories were recorded in a suitable format for further processing.

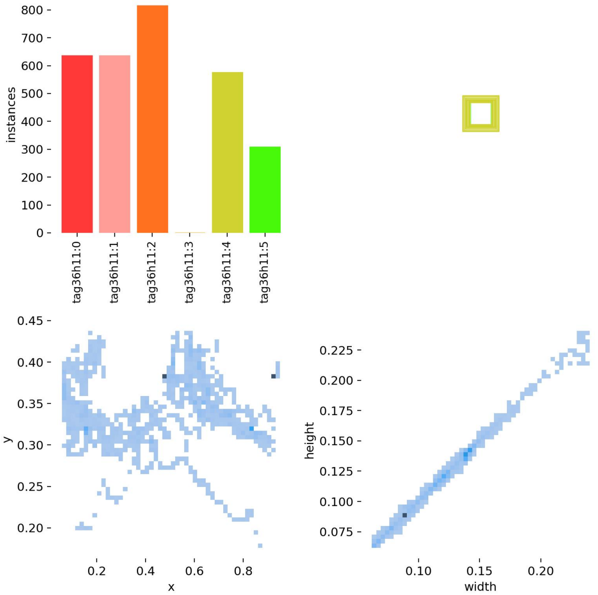
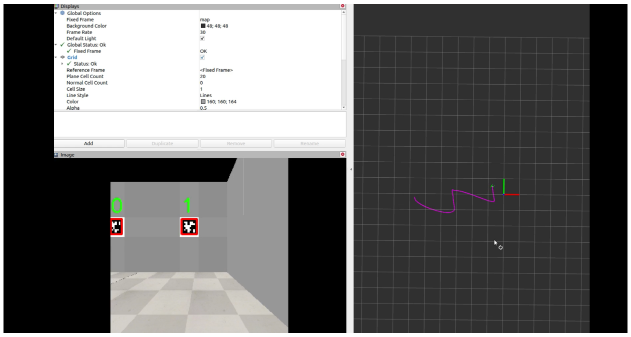


Fig. 4: Training Results

C. Trajectory Tracking Performance

YoloTag was designed for vision-based autonomous navigation tasks in real-world scenarios. To demonstrate its capabilities, trajectory tracking performance was evaluated on an irregular-shaped trajectory within the contoured environment in CoppeliaSim. For this irregular trajectory, where changes in direction occur frequently, prioritizing smoothness and adaptability is essential. Hence, minimum snap trajectories are preferable due to their ability to minimize jerk and provide smooth, continuous motion.



Global Options
Fixed Frame map
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ✓
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ✓
✓ Status: Ok
Reference Frame <Fixed Frame>
Plane Cell Count 20
Normal Cell Count 0
Cell Size 1
Line Style Lines
Color ▨ 160; 160; 164
Alpha 0.5

Fig. 5: In this figure, the YOLO object detector is employed to detect markers along with their IDs, while the UAV utilizes this information for localization.

As evident from Figure 5, the trajectory from YoloTag closely follows the ground truth trajectory. YoloTag incorporates a second-order Butterworth filter to reduce noise in the estimated trajectory, offering advantages in noise reduction, smoothing abrupt changes, and adapting to varying conditions. Additionally, the Butterworth filter incorporates temporal information for sequential detection and tracking of dynamic objects, enhancing the accuracy and stability of the estimated trajectory.

By evaluating YoloTag on this irregular-shaped trajectory within the contoured environment, its capabilities in vision-based autonomous navigation tasks, including marker detection, path following, and localization algorithms, are demonstrated in a simulated real-world scenario.

VII. Conclusion

This work introduces YoloTag, a novel fiducial marker detection architecture designed for vision-based UAV navigation tasks in simulated environments. YoloTag employs a lightweight YOLOv8 object detector for accurate marker detection and an efficient multi-marker based pose estimation algorithm to robustly compute the UAV's pose across diverse simulated conditions in CoppeliaSim. A lightweight YOLOv8 model is trained on a dataset generated from simulated experiments involving a UAV capturing onboard camera images in CoppeliaSim. A mixed annotation approach, combining an Apriltag detector and manual annotation, enables an efficient training regime. Notably, by incorporating sequential pose information through a second-order Butterworth filter, YoloTag achieves superior performance and real-time capabilities compared to existing methods, as evidenced by its performance across multiple distance metrics in the simulated environment. The proposed architecture offers a robust, efficient, and real-time solution for marker-based UAV localization and navigation in simulated GPS-denied environments within CoppeliaSim, outperforming traditional approaches. Future work will focus on employing YoloTag for UAV localization by detecting common objects in simulated environments, thereby overcoming the limitation of fiducial detection methods that are constrained to predefined markers.

References

[1] F. Bergamasco, A. Albarelli, E. Rodolà, and A. Torsello, "Runetag: A high accuracy fiducial marker with strong occlusion resilience," in *CVPR 2011*, pp. 113–120, 2011.

[2] G. Yu, Y. Hu, and J. Dai, "TopoTag: A Robust and Scalable Topological Fiducial Marker System," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 27, no. 9, pp. 3769–3780, 2021.

[3] X. Zhang, H. Guo, J. Mariani, and L. Xiao, "U-star: an underwater navigation system based on passive 3d optical identification tags," pp. 648–660, 10 2022.

[4] J. B. Peace, E. Psota, Y. Liu, and L. C. Pérez, "E2etag: An end-to-end trainable method for generating and detecting fiducial markers," 2021.

[5] M. B. Yaldiz, A. Meuleman, H. Jang, H. Ha, and M. H. Kim, "Deepformabletag: end-to-end generation and recognition of deformable fiducial markers," *ACM Transactions on Graphics*, vol. 40, p. 1–14, July 2021.

[6] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pp. 85–94, 1999.

[7] E. Olson, "Apriltag: A robust and flexible visual fiducial system," *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, 2011.

[8] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198, 2016.

[9] J. DeGol, T. Bretl, and D. Hoiem, "Chromatag: A colored marker and fast detection algorithm," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1481–1490, 2017.