Object detection and tracking for drones: A system design using dynamic visual SLAM

Dongli Wu

College of Design and Engineering, National University of Singapore, Singapore, Singapore

e1373956@u.nus.edu

Abstract. Drones can play a quite crucial role in many walks of life today. Enhancing the visual perception ability of drones is crucial to their intelligence level. Among them, it is necessary to focus on strengthening the detection, tracking and mapping capabilities of drones for dynamic objects. However, the existing visual SLAM systems carried by drones do not perform well in dynamic environments. This project designs a monocular visual SLAM system specifically for drones, aiming to achieve efficient three-dimensional mapping and target tracking, surpassing the limitations of simple static mapping and positioning. Besides, this project constructs a drone dynamic SLAM system developed on the ORB-SLAM3 structure, uses drone images to detect, track and map object motion models, and reconstructs environmental maps to obtain motion parameters with real physical scales. This project strives to optimize the input pre-processing module, improve the validity of data and output environmental maps and raster maps. The outcomes demonstrate the system's strong accuracy and adaptability in dynamic installation procedures.

Keywords: Visual simultaneous localization and mapping, Drones, Dynamic objects tracking.

1. Introduction

Drones have emerged as effective solutions in various fields, including environmental regulation, emergency response, and transportation logistics. In complex working environments, the processing capabilities of advanced drones depend on their intelligence level, which is significantly affected by their visual perception capabilities. Drones are often equipped with cameras and sensors like Inertial Measurement Unit (IMU) to provide raw data for simultaneous localization and mapping (SLAM) or Visual Odometry (VO) technologies, the foundation and core of perception, modeling, planning, and understanding [1]. However, many visual SLAM algorithms currently operate as though the scenes are static, and merely some geometric point information, not high-level semantic information, is contained in the information obtained [2]. Hence, to enhance the performance of drones in dynamic environments, an optimized algorithm specifically for drones' visual SLAM systems is needed.

Throughout the previous decade, the integration of dynamic target tracking and SLAM systems has made great progress. In the early days, dynamic target detection was performed by identifying it as a static target or by changing the external environment's texture and coloration. The above methods are still based on the framework of static SLAM. Recently, some researchers have integrated dynamic object tracking into SLAM, considering the movement of objects to improve environmental understanding beyond static mapping and localization [3]. The Single Shot MultiBox Detector (SSD) target detection framework, which was proposed considering rapidity and real-time performance, increases speed while maintaining the accuracy of detection. To enhance the neural networks' speed of operation, miniaturized networks represented by MobileNet have been proposed. These networks reduce the number of network operations by cleverly designing network structures and simplifying convolution kernels. Among the visual SLAM algorithms developed recently which employ the feature point technique, the ORB-SLAM2 algorithm is a comparatively exceptional algorithm framework. Multiple elements are frequently present in the image information stream that the visual SLAM algorithm receives. By integrating the target detection network's advantages in semantic information extraction with the exact geometry data acquired by the SLAM algorithm, the robot may acquire more hierarchical, structured, and semantic map information from its surroundings. In recent years, some studies have eliminated dynamic points within the frame of view of the camera based on the prior information of the target detection results and the measurement information of the dynamic point detection algorithm. To boost the algorithm's positioning accuracy.

However, there are still many areas for improvement in the process of combining SLAM with drones. First, the monocular camera carried by the drone lacks the ability to process information and the ability to quickly reconstruct 3D maps, which adds uncertainty when dealing with dynamic targets with speed. Besides, the images processed by conventional SLAM systems rarely contain visual phenomenon information based on the unique bird's-eye view of drones.

This project takes the complex and dynamic outdoor laboratory environment as the background, explores the construction method of semantic maps in a dynamic environment, and combines the visual SLAM system based on one RGB-D camera utilizing the SSD framework for deep convolutional neural networks based on regression prediction and multi-scale prediction to design an algorithm to remove dynamic feature points, process 2D semantic image information, and build a 3D semantic target database. Our contributions include:

- Building the drone monocular vision system to attain effective target detection, tracking, and 3D mapping. It goes beyond the limitations of static mapping-only positioning.
- Training and building a SLAM system based on drone datasets, as drone scenarios differ from traditional vehicle scenarios, necessitating a specialized approach. Experimental testing confirmed the model's effectiveness.
- Combining and optimizing the latest visual technology to achieve pre-processing of 2D semantic data and greatly improve system accuracy.

The proposed SLAM system uses dynamic feature point elimination, tracks dynamic objects, reconstructs the 3D model of the target in combination with semantic information, builds a dynamic semantic map, and obtains a grid mesh model that is beneficial for drone flight in a dynamic environment. The final analysis results in a map combining the environment and dynamic targets.

2. Related work

The dynamic visual SLAM has been well developed. In this chapter, Section 2.1 will focus on the principles, development, shortcomings, and specific implementation steps of dynamic visual SLAM. Section 2.2 will introduce the current application status of SLAM on drones.

2.1. Visual SLAM system

Visual SLAM is an algorithm that, in an unknown environment, utilizes visual sensors, including cameras, which concurrently estimate the subject's motion trajectory and reestablish the three-dimensional structure of the surrounding environment. When compared to laser SLAM, visual SLAM has the advantage of being cheaper to execute, easy installation, comprehensive environmental information, and simplified sensor integration. Consequently, it has found extensive application in the domain of mobile robotics, such as drones. Two of the primary types of conventional visual SLAM algorithms are the feature-point-based approach and the direct method. The feature point-based method first extracts the input image's feature points and their descriptors, then uses feature point matching to estimate camera motion and construct an environmental map. This method retains the image's primary information and minimizes the amount of computation, but there are problems such as long feature extraction times, ignoring some image information, and matching failure in textureless areas. The direct method is presuming the grayscale invariance of pixels, directly using the original pixel information of the image to minimize photometric error when estimating the scene's composition and camera motion. This method regards the entire process as an energy minimization problem and solves the camera pose by iteratively optimizing the energy function. The algorithm can function normally as long as it detects variations in the amount of darkness and light in the scene. The direct approach, as opposed to the feature point method, maximizes the information in the image and may effectively portray areas with little roughness [4].

Furthermore, visual SLAM algorithms can be classified into two categories: tightly coupled and loosely coupled, based on how the camera and other sensors, like the IMU, are fused together. The tightly coupled method jointly optimizes all sensor data, which is more accurate but more computationally intensive, while the loosely coupled method optimizes sensor data separately, which is slightly less accurate but more efficient.

Regardless of the visual SLAM algorithm used, core issues such as data association, motion estimation, loop detection, and pose graph optimization need to be solved. In complex environments, such as dynamic scenes and complex lighting, the robustness and accuracy of existing algorithms still need to be further improved [5].

The visual SLAM process can be divided into five key stages: sensor data acquisition, visual odometer, back-end optimization, closed-loop detection, and mapping. During the front-end phase, landmarks are identified, and the camera's pose is estimated in order to achieve real-time location by removing and matching feature points from the image. Subsequently, the back-end optimization link eliminates the accumulated error through technologies such as inter-frame common view relationships, thereby enhancing the accuracy of positioning and map construction. When a closed loop in the pose estimation is detected, that is, the similarity between two frames exceeds a predetermined threshold, the system will optimize these landmarks and poses. To enhance the expectations of map construction. The sensor data acquisition stage involves obtaining raw data from sensors, such as cameras, and performing the necessary pre-processing for subsequent work. In a multi-sensor fusion system, this stage also includes processing data from other sensors, such as IMUs and encoders [6].

2.2. Positioning and navigation for drones

The basis and premise for drones to achieve a range of intricate functions, including path planning, obstacle avoidance, flight control, and precise positioning. Three types of positioning methods are widely used at present: one is based on external devices to provide accurate location information, mainly including GPS and motion capture; one is the optical flow method for flight navigation and obstacles, and the robot is visually measured and relative motion perceived through the optical flow algorithm; and the other relies on the simultaneous localization and mapping (SLAM) of the environment by the quadcopter itself, which restores the environmental profile through sensors and simultaneously locates its position in the map to achieve navigation. Among them, GPS and IMU are the mainstream algorithms for outdoor use and have been widely used. The optical flow method can only determine the relative posture movement but cannot determine the absolute position, so it is generally used for stable hovering indoors. The SLAM method requires real-time map construction, which is divided into laser SLAM depending on lidar and visual SLAM based on monocular, binocular, or RGB-D cameras. In the scenario of indoor positioning, there have been many related studies, but it is not widely used in commercial drone applications. Among the visual SLAM methods, there are those based on VO (visual odometry) and VIO (inertial visual odometry). Among the VO-based methods, the framework that is more widely used on drones is SVO. This is a visual odometry calculation method published in 2014 by the laboratory of Professor Scaramuzza of the University of Zurich [7, 8]. In the flight experiment, this method tracked a distance of 84 m, and the position error was still controlled within ±0.1 m. The speed can reach hundreds of frames per second, which is currently a method in visual SLAM that is more suitable for drones. Among the VIO-based methods, the Hong Kong University of Science and Technology has open-sourced a VIO algorithm named VINS-Mono, which is implemented using a tightly coupled method and restores the scale through monocular and IMU with excellent results [9].

3. Methods

3.1. General Introduction

The ORB-SLAM3 framework serves as the foundation for the proposed SLAM system. It can be equipped with a monocular camera to capture images, and the environmental construction of the target object and the poster display of the camera are realized through code. At the same time, the system integrates a functional module for tracking objects for obstacle avoidance [10]. New functional modules suitable for drone positioning, tracking, and navigation are integrated into the ORB-SLAM3 framework. This method uses 2D semantic information to reconstruct a 3D environment map, enabling drones to recognize and track objects in a dynamic environment. The overall steps include the construction of the SLAM framework, the pre-processing module, pre-processing process synchronized with the SLAM framework, target three-dimensional spatial positioning, dynamic semantic map building, and implementing 2D rasterized maps.

3.2. ORB-SLAM3 Algorithm and RGB-D camera

The first model to use fisheye and pinhole lenses, ORB-SLAM3 completes visual inertial and multi-map SLAM with the addition of monocular, binocular, and depth cameras. ORB-SLAM3 is a visual-inertial SLAM system depending on tight feature coupling. The principle is shown in Figure 1. Its framework parts are tracking threads, local mapping, and closed-loop detection. In the figure, the keyframes run through the entire algorithm, connecting these three parts and containing the camera information of the image frame, ORB feature points, and IMU calibration parameters [11]. The tracking thread's tasks include keyframes creation and an approximate estimation of the camera's pose. To estimate the pose, ORB-SLAM3 incorporates IMU pre-integration into the tracking thread. In normal tracking, the distance of the object from itself is calculated by actively projecting known patterns, which improves the matching robustness.

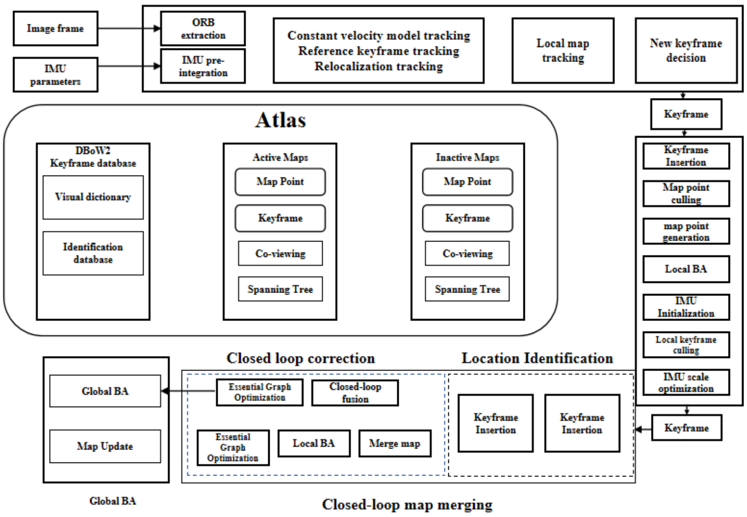

Figure 1. ORB-SLAM3 framework

After measuring the depth value, the RGB-D camera uses mechanical installation to determine the relative pose, uses the depth and color image pixels for pixel pairing, and outputs the corresponding color image and depth map. Distance information and color information can be read at the same position, thereby calculating the pixel coordinate value and generating a point cloud. Although the depth camera can actively measure the depth of each pixel using structured light, it is easily affected by sunlight and is not suitable for objects on reflective surfaces. Considering that the application scenario of the experimental camera this time is a mine tunnel without sunlight, compared with other types of cameras, the algorithm's accuracy and stability are greatly reduced in dim environments. Although the RGB-D camera [12] using TOF technology has poor resolution, it can meet the needs of SLAM for calculation accuracy and environmental information and has good performance in motion scenes, which is suitable for the research and development of visual SLAM.

3.3. Pre-processing module

The pre-processing module mainly combines cutting-edge computer vision technology to complete dynamic object detection of input image information, monocular depth estimation, and optical flow estimation in order to finish extracting the dynamic feature points. The SLAM system faces the following problems in outdoor environments: First, learn how to effectively distinguish dynamic objects from static environments with the movement of the drone itself. Secondly, to maintain efficient tracking, considering the distance between the drone and the ground target may cause the target image to account for a small proportion of the input image. The pre-processing module mainly combines cutting-edge computer vision technology to complete dynamic object detection of input image information, monocular depth estimation, and optical flow estimation to complete the extraction of dynamic feature points, thus enhancing the system's accuracy in dynamic situations outdoors.

(1) Dynamic object detection. At present, various techniques employ deep learning techniques to eliminate dynamic spots to polish SLAM, such as the famous DS-SLAM, DynaSLAM, etc. However, both of these methods combine SLAM and semantic segmentation recognition. Using semantic segmentation can more accurately select dynamic objects without wasting static points, but their running speed is greatly reduced and the computational cost is too high. Considering the foreground and the existence of static objects (vehicles) and dynamic objects (pedestrians and vehicles) in the experimental scene, the Yolo target detection technique is employed to exclude dynamic feature points. The Yolo system has high accuracy in similar environments. This project utilizes Yolov8 as the fundamental method for dynamic object detection. After introducing the attention mechanism, we use the COCO datasets for training, fine-tune it on the VisDrone 2019 datasets, and finally, obtain a dynamic object detection model with good training results.

The existing Yolov8 model performs well in target detection on conventional datasets but poorly in complex, dynamic environments [13]. Experiments have shown that, for this dataset, the recognition effect of conventional convolution can be optimized by introducing the CBAM attention mechanism. CBAM (Convolutional Block Attention Module) focuses on target object recognition and includes two separate components: Channel Attention Module (CAM) and Spatial Attention Module (SAM). This mechanism can be integrated into existing modules while saving parameters and computational complexity, thereby introducing the calculation of the attention mechanism during feature extraction. The role of the attention mechanism is to perform weighted calculations on different targets in the feature map in different dimensions, improve the algorithm's ability to extract the main features, and thus improve the accuracy of target detection. This system improves the C3 module in the backbone of Yolov8 to the CBAMC3 module; that is, the last standard convolution of C3 is changed to an attention mechanism module, and attention calculation is integrated into the feature extraction process.

(2) Monocular depth estimation. In the input image, static areas and potential dynamic areas can be distinguished. Using the monocular depth estimation method can help the system maximize the collection of dynamic feature points. This project takes advantage of an advanced computer vision method, DEPHT-ANYTHING trained on KITTI and FlyingChairs, to complete the depth estimation of

potential dynamic areas, output relative depth, calculate absolute depth, and feedback on the processed depth image.

Optical flow estimation. There may be multiple dynamic objects in the images obtained by the monocular camera of an outdoor drone, so an optical flow estimation module is presented with complete the unified tracking of the feature points of dynamic objects. This is because optical flow estimation can assign points to the dynamic objects in the dynamic scene and propagate between frames, thereby forming a dense optical flow to avoid the failure of dynamic object tracking. This system uses Matchflow [14] as the optical flow estimation method, which is trained on the FlyingChairs, Sintel, and KITTI training data sets. It can output the contours of dynamic objects in static scenes.

In general, the pre-processing module processes the input semantic information and continuously outputs the information of dynamic feature points. This useful information will play a role in subsequent modules.

3.4. Pre-processing process synchronized with SLAM framework

To achieve semantic SLAM in dynamic contexts is the main objective of this project. Applying the aforementioned techniques, the system first identifies static and dynamic objects and passes the object frame data to ORB-SLAM3. Subsequently, in ORB-SLAM3, the feature points within the dynamic objects are eliminated. This can enhance the system's efficiency in a very dynamic setting.

The prior implementation approach involved running Yolov8 first, obtaining the object frame data in.txt format, extracting this data employing ORB-SLAM3, and then performing the necessary follow-up work. Before now, the entire process was asynchronous and not real-time. This project achieves synchronization through the use of the UNIX domain socket communication mechanism. The steps are as follows:

Add the initialization of the socket at the beginning of the rgb_tum.cc file of the ORB-SLAM3 code. Then, add two functions to the rgb_tum.cc file of the ORB-SLAM code. The function of the LoadBoundingBoxFromPython function is to read the data we need from a sentence of object frame data and store it. The function of the MakeDetect_result function is to split the object frame data from a piece of object frame data one sentence at a time and then call the LoadBoundingBoxFromPython function to read them separately. To modify the detect.py code of Yolov5, two steps were done. One was to delete the code for storing pictures in files in order to speed up the program. The second was to add a socket to transfer the object frame data of one frame to Candand at a time.

3.5. Target three-dimensional spatial positioning

The target three-dimensional spatial positioning module is mainly divided into two steps, as shown in Figure 2. First, the target relative to the camera posture is determined by integrating the intrinsic information of the camera with the keyframe information and the output findings of the target detection module. Then, to finish solving the target object's position concerning the world coordinate system, the keyframe information is used to pick the acquired camera pose data.

The target three-dimensional spatial positioning module consists of three main nodes. The first driver node uses the RGB-D camera sensor to gather information, forwarding the picture and depth data to the subsequent node. The second node utilizes the ORB-SLAM3 algorithm for posture estimation and provides a matrix that transforms the camera coordinate system from the world coordinate system. The third three-dimensional spatial positioning node obtains the first two nodes' picture and posture data and realizes the three-dimensional coordinate solution of the target through the integrated pre-processing module.

This module creates three subscription topics, which respectively receive the camera pose, color information, and depth image, and synchronize the timestamps of the three. The synchronized key frame image, the corresponding depth image, and the camera's position in relation to the global coordinate system are passed to the pre-processing module. The detection module outputs the pixel coordinates of the detection image and the target detection frame and afterwards utilizes the pixel coordinates to calculate the target's three-dimensional position about the camera coordinate system, camera internal parameters, and depth values. Among them, the depth value is determined by the pixel coordinates of the target center point, the depth image, and the depth factor. In order to reduce the impact of the fluctuation of the single-point depth value, this paper uses the average depth value of the 5×5 neighborhood of the target point. Finally, the three-dimensional coordinates of the target in the world coordinate system are solved when paired with the camera's position with respect to the coordinate system. The following is the basic formula for determining the camera pose:

$$z = \frac{d}{depth} \tag{1}$$

$$x = \frac{(u-c_x)z}{f_x} \tag{2}$$

$$y = \frac{(v-c_y)z}{f_y} \tag{3}$$

In the formula, (x, y, z) is the object coordinate of the camera, so (u, v) is used to find the pixel coordinate.

3.6. Dynamic semantic map building

If a moving object enters the camera's field of vision and causes significant movements during the SLAM map creation process, this will impact the estimation of the camera's location and posture. The moving object's trajectory will appear on the point cloud map simultaneously, and the data will be preserved regardless of whether the point cloud map is transformed into another format. The inability to detect whether a moving object can pass safely prevents direct navigation using maps containing dynamic object information. How to find the area of dynamic objects in the image sequence and eliminate the influence of these areas has been the focus of research in the past two years. This project improves the ORB-SLAM3 framework by adding dynamic and static point detection algorithms, target detection algorithms, and dynamic map construction algorithms to realize the semantic map construction process in dynamic environments. This project completes the target recognition and positioning experiment in dynamic environments.

3.7. Implementing 2D rasterized maps

In order to facilitate the navigation of drones, a 2D raster map is implemented. The construction of the raster map is based on the construction and preservation of the dense point cloud map. Based on ORB-SLAM3, a dense point cloud map is constructed in real-time. On the basis of the point cloud map, an octree map and a two-dimensional raster map containing occupancy information are constructed to facilitate subsequent obstacle avoidance, navigation, and other functions. The following problems may occur during the process:

- Problems that may be encountered in the real-time display of Octomap in the ROS environment.
- The point cloud is perpendicular to the grid.
- The octree map is not fully displayed.
- The ground is also displayed as occupied.

Solutions include optimizing display settings, ensuring coordinate systems are aligned, adjusting Octomap resolution and updating regions, and avoiding false detections through ground filtering or adjusting occupancy thresholds.

4. Results

4.1. Experiment setup

After the algorithm is built and optimized in Ubuntu 18.04, simulation experiments are performed using an open-source dataset (https://github.com/lemonhi/drone_dataset/tree/main). The datasets are captured by the RBG monocular camera carried by the DJI Mini 4 drone. The datasets obtain basic information on the streets and roads from a bird's-eye view. The data set analyzed in the experiment is shown in Figure 2.

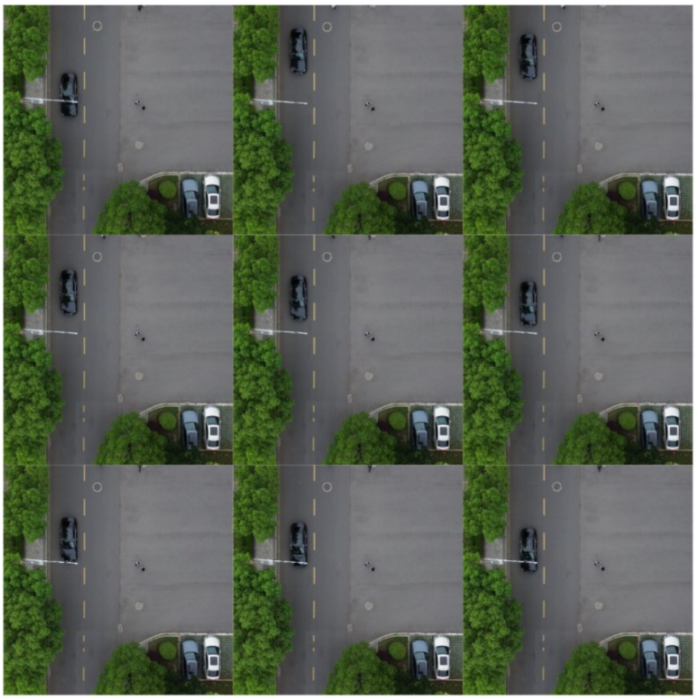

Figure 2. Partial datasets

4.2. Experiment results

First, use the pre-processing module to analyze the datasets. Some results of object detection frame recognition, monocular depth estimation, and optical flow estimation are respectively shown in Figures 3, 4, and 5.

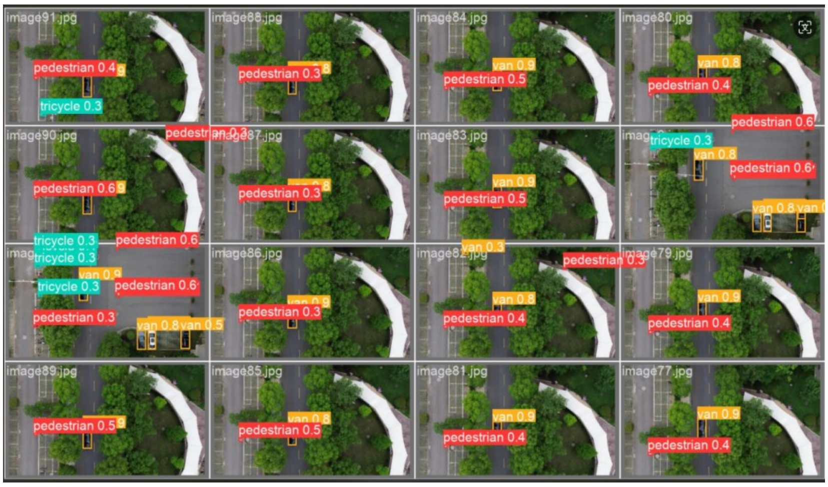

Figure 3. Partial object detection

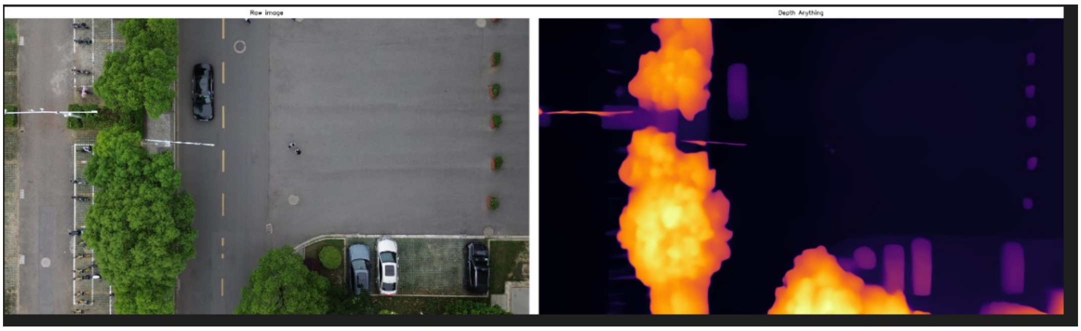

Figure 4. Partial monocular depth estimation

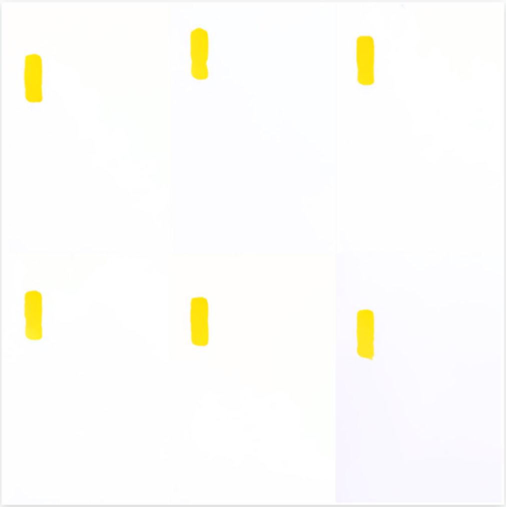

Figure 5. Partial depth estimation

Figure 6 shows the results of the experiment on the dynamic system after the datasets were built. As shown in the figure, the black background is the result of 3D remodeling, the red line is the movement route of the camera, and the blue line is the movement route of the dynamic object (moving vehicle).

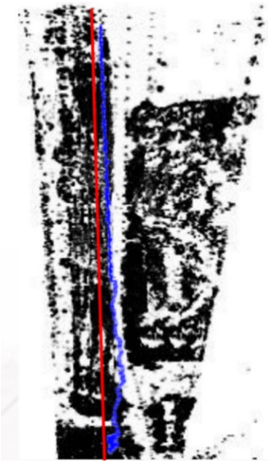

Figure 6. Illustration of the results on the drone datasets

Figure 7 shows the 2D grid path of the drone used in the pre-test after the debugged rasterization module is combined with MATLAB output, and Figure 8 shows the output path diagram of the dynamic object (moving vehicle) in the datasets studied in this paper.

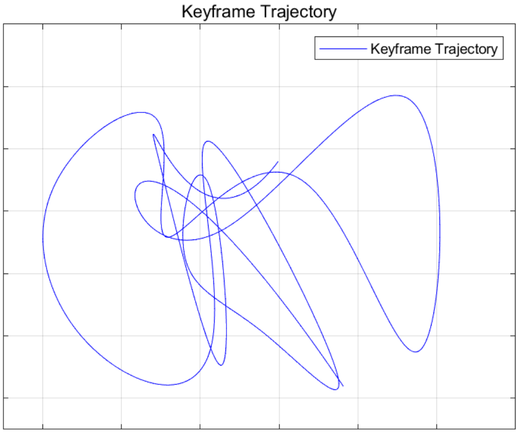

Figure 7. pre-test result

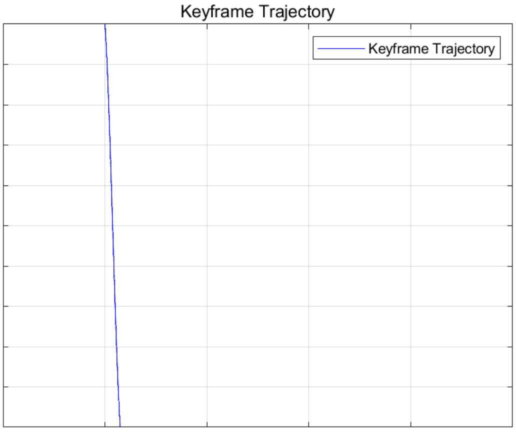

Figure 8. path diagram of the dynamic object

5. Discussion

5.1. Discussion of optimized object detection

Compared with the original Yolov8 experimental results, the improved model can more effectively obtain the environment's semantic information and more accurately identify the target object, with a significant improvement in the accuracy of the dynamic object (a moving car) shown in Figure 9. The comparison with other methods is shown in Table 1.

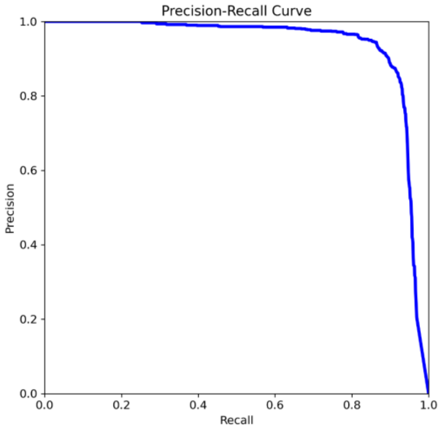

Figure 9. Accuracy of dynamic object

Table 1. Comparison of different Yolo

Method	mAP@.5	mAp@.5:.95
Yolov3	0.431	0.213
Yolov5	0.492	0.293
Yolov8	0.573	0.315
Ours	0.893	0.701

5.2. Comparison of this SLAM system with other SLAM

First, the VDO-SLAM system is used together with this solution to experiment on the KITTK datasets, and the results are compared. VDO-SLAM is also a system based on monocular vision and has certain similarities with this solution. The results are shown in Table 2.

Table 2. Comparison of different SLAM

Method	VDO-SLAM[15]				Ours			
	Camera Pose		Object Trace		Camera Pose		Object Trace	
	E_r(deg)	E_t(m)	E_r(deg)	E_t(m)	E_r(deg)	E_t(m)	E_r(deg)	E_t(m)
00	0.1830	0.1847	2.0021	0.3827	0.0732	0.0756	1.6821	0.3794
01	0.1772	0.4982	1.1833	0.3589	0.00672	0.1255	1.0476	0.4372
02	0.0496	0.0963	1.6833	0.4121	0.0387	0.0583	1.0405	0.4103
03	0.1065	0.1505	0.4570	0.2032	0.0736	0.1322	1.0459	0.3870
04	0.1741	0.4951	3.1156	0.5310	0.0656	0.1467	1.9683	0.6202

The table displays the translation and rotation error values for the two approaches. By comparison, this method has better accuracy. At the same time, this method may not be able to accurately propose dynamic feature points when processing dynamic scenes, so it has similar accuracy to the other method.

5.3. Discussion on the practicality of this solution

This project mainly proposes a dynamic SLAM system for the bird's-eye view of drones, which can be confirmed to be highly effective in experimental results. This solution combines a variety of pre-processing modules to obtain more robust analysis data, and achieves communication synchronization between the pre-processing module and the SLAM system, which is beneficial to its practical application and effectiveness on real drone platforms. Better integration. This solution also combines the object-wise tracking method for the elimination of dynamic and static feature points, which can significantly improve the system's processing capabilities for dynamic scenes.

6. Conclusion

This project aims to establish a drone-based dynamic SLAM system, and uses orb-SLAM3 as the basic framework to set up the pre-processing flow of the input image (object detection frame, depth map and dense optical flow estimation), and realizes the visualization of the dynamic environment on the map and the output of the processed raster map, thus realizing the relatively complete construction of the drone-based SLAM system. This topic endeavors to optimize the input pre-processing module, improve the effectiveness of data and repair the output result of the whole system. Meanwhile, this system has good adaptability to dynamic environments.

This solution also has many areas that can be optimized in the subsequent results. a) In the process of camera positioning, the existence of dynamic key points will interfere with the positioning of the camera. The key to improving the camera's positioning accuracy in a dynamic environment is to discover how to remove dynamic points from the positioning process. This project proposes a moving point identification algorithm based on the optical flow algorithm, but its accuracy is low, and it is also prone to many false detections. b) When there is an object occlusion, it will affect the accuracy of the system, and more refined segmentation can be used.

References

[1] Xi, Y., Liu, J., Wang, Z., & Chen, C. (2022). Learning-empowered resource allocation for air slicing in drone-assisted cellular V2X communications. *IEEE Systems Journal*, 17, 1008–1011.

[2] Bescós, B., Fácil, J. M., Civera, J., & Neira, J. (2018). DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3, 4076–4083.

[3] Brata, K., Funabiki, N., Panduman, N., & Fajrianti, E. (2024). An enhancement of outdoor location-based augmented reality anchor precision through VSLAM and Google Street View. *Sensors*, 24(4), 1161.

[4] Barros, A. M., Michel, M., Moline, Y., & Corre, G. (2022). A comprehensive survey of visual SLAM algorithms. *Robotics*, 11(1), 24.

[5] Zhang, B., Dong, Y., Zhao, Y., & Qi, X. (2024). DynPL-SLAM: A robust stereo visual SLAM system for dynamic scenes using points and lines. *IEEE Transactions on Intelligent Vehicles*.

[6] Chen, W., Shang, G., Ji, A., et al. (2022) An overview on visual slam: From tradition to semantic. *Remote Sensing*, 14(13): 3010.

[7] Shan, M., Wang, F., Lin, F., Gao, Z., Tang, Y. Z., & Chen, B. M. (2015). Google map aided visual navigation for drones in GPS-denied environment. In *Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics*, 114–119.

[8] Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, 15–22.

[9] Kim, Y. (2021). Aerial map-based navigation using semantic segmentation and pattern matching. *arXiv*, arXiv:2107.00689.

[10] Campos, C., Elvira, R., Gómez Rodríguez, J. J., Montiel, J. M. M., & Tardós, J. D. (2021). ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37, 1874–1890.

[11] de Jesus, K. J., Pereira, M. O. K., Emmendorfer, L. R., & Gamarra, D. F. T. (2023). A comparison of visual SLAM algorithms ORB-SLAM3 and DynaSLAM on KITTI and TUM monocular datasets. In *Proceedings of the 2023 36th SIBGRAPI Conference on Graphics, Patterns and Images*, 109–114.

[12] Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2014). 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30, 177–187.

[13] Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). drone-Yolov8: A small-object-detection model based on improved Yolov8 for drone aerial photography scenarios. *Sensors*, 23(16), 7190.

[14] Dong, Q., Cao, C., & Fu, Y. (2023). Rethinking optical flow from geometric matching consistent perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 1337–1347.

[15] Zhang, J., Henein, M., Mahony, R., & Ila, V. (2020). VDO-SLAM: A visual dynamic object-aware SLAM system. *arXiv*, arXiv:2005.11052.