

Adaptive Visual Obstacle Avoidance Algorithm for Aerial Drones Integrating Deep Learning and Spatial Geometric Transformations

Haitian Sun^{1, a}

¹School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215000, China

^aHaitian.Sun22@student.xjtlu.edu.cn

Abstract: This study proposes a new visual obstacle avoidance method for drones based on the YOLO and PnP algorithms. Most existing obstacle avoidance methods are based on sensors such as lidar and ultrasonic sensors, or visual obstacle avoidance methods based on monocular or binocular cameras. In contrast, this study only uses a monocular camera on a two-axis gimbal on an aerial drone to detect faces in the picture, protecting the human head, which is the object that may be the most harmful, with the lowest hardware and algorithm costs, and providing effective obstacle avoidance strategy support for the autonomous flight safety of aerial drones.

Keywords: UAV, Visual Obstacle Avoidance, YOLOv8, PnP Algorithm.

1. Introduction

Quadcopter drones have developed rapidly in recent years due to their flexible maneuverability and wide application prospects. In particular, drones have demonstrated their unique advantages in the fields of security monitoring, disaster relief, and intelligent inspection. However, one of the biggest challenges faced by drones during autonomous flight is obstacle recognition and avoidance. In general aerial photography drone usage scenarios, when shooting images of a crowd, the distance to people is very close. If an accidental collision occurs, the most likely target of injury is often humans, especially the human head.

In 2018, a 9-year-old boy was accidentally hit in the face by a drone propeller [1], resulting in severe eye and facial injuries. The injuries included hemorrhage in the right eye, partial tear of the upper eyelid, full-thickness corneal laceration, and scleral laceration. The patient recovered some vision after surgery, but still had visual defects, with partial loss of the upper nasal and upper temporal fields of the right eye, and the final vision was 20/400. This incident shows that drones can cause extremely serious damage to the human body, especially the head, and needs to be taken seriously.

An obstacle avoidance system for the human head that can be deployed on all aerial photography drones or even all drones is particularly important.

Traditional obstacle avoidance technologies mainly rely on laser radar (LIDAR) [2] or ultrasonic sensors [3-5]. These obstacle avoidance methods have several major disadvantages. First, sensors such as laser radar are very heavy [6], which will increase the flight burden of the drone and reduce the flight endurance [7]. Second, the large amount of point cloud data generated by the laser radar and the echo signal of the ultrasonic sensor need to be processed in a complex manner, which places high demands on the computing power of the drone [2]. Aerial drones do not have separate computing units and have limited computing resources, so this processing requirement will lead to performance bottlenecks.

In recent years, with the rapid development of computer

vision and deep learning technology, vision-based target detection and recognition methods have received widespread attention. Vision-based algorithms are divided into monocular and binocular. Binocular vision can provide relatively accurate depth information, but the cost of the camera and computing unit is much higher than that of monocular vision. The biggest disadvantage of monocular vision is that it cannot directly obtain the depth information of the image. The position of the target point in three-dimensional space can be calculated through methods such as the PnP algorithm. Most purely visual obstacle avoidance methods tend to detect all types of objects, such as the optical flow method used by the research teams of Elawady, Sadek and Kidane, which uses a monocular camera to detect and avoid obstacles in front of the camera [8-10]. Aguilar's research team used feature point recognition to avoid obstacles [11]. By extracting key feature points in the image and using them to calculate the position and relative motion of obstacles, an efficient obstacle avoidance strategy was achieved. Meng et al. proposed a 6D pose prediction method based on a single CNN architecture through deep learning technology, which can be quickly and accurately applied to object detection and obstacle avoidance tasks. Yang et al. proposed a method of using a lightweight probabilistic convolutional neural network (pCNN) for monocular depth estimation and obstacle avoidance [12], which optimized the processing efficiency of the camera and reduced the computational overhead. However, it is obvious that detecting the entire picture and all objects in the picture still requires certain computational power of the computing unit, and it is not an extremely low-cost obstacle avoidance system that can be deployed on all aerial drones or even all drones.

This study proposes an obstacle avoidance system based on face recognition. This method aims to reduce the potential damage caused by aerial drones to humans, especially the head, with the lowest hardware and algorithm costs. Specifically, in general aerial drone usage scenarios, if a collision occurs, the greatest damage may be caused to humans, especially the head. Therefore, the obstacle avoidance method that only uses the two-axis gimbal camera

of the aerial drone to recognize faces and avoid obstacles can effectively reduce the significant damage that aerial drones may cause to humans, especially the head. This method does not require lidar or ultrasonic sensors and extra cameras, and also reduces the pressure on the computing unit as much as possible. This study introduces the concept of sensor fusion. By combining the attitude information provided by the IMU (inertial measurement unit) sensor with visual data, the system's accuracy in measuring the target distance is further improved. As a classic computer vision algorithm, the PnP (Perspective-n-Point) algorithm can accurately calculate the position and attitude of the camera by combining the feature points in the image with the feature point positions in the world coordinate system. Combining the data of the IMU sensor with the PnP algorithm can better correct the actual attitude of the drone, thereby achieving accurate measurement of the target distance.

The DJI Mavic series aerial photography drone used in this article is a portable, six-degree-of-freedom rotating multi-rotor aircraft with high flexibility and stability. The mechanical structure of this drone consists of four motor-driven rotors, which are divided into two groups: one group rotates clockwise and the other group rotates counter clockwise to ensure balance and stability during flight. This design enables the drone to perform precise attitude control in three-dimensional space, including six degrees of freedom movement such as front and back, left and right, up and down, and rotation. This drone is equipped with a high-definition camera with a resolution of 1080p or 4K, supports high-definition video recording at 30 frames per second, and has H.264 video encoding format and JPEG photo shooting functions. The camera's viewing angle is about 78.8°, and the smaller focal length (26mm (35mm equivalent)) enables the camera to obtain a wider field of view and include more environmental details. Equipped with a two-axis mechanical gimbal, it can stabilize the camera in realtime during the flight of the drone. By adjusting the pitch and roll angles, the image jitter caused by flight is reduced to ensure the smoothness and clarity of the output image.



Figure 1. The UAV and the camera

The innovation of this study is that a UAV face recognition and position calculation method combining the YOLOv8 model, PnP algorithm and sensor fusion is proposed. After detecting the face and determining its distance in the real world, the system will automatically trigger the obstacle avoidance strategy when the face distance is less than the safety threshold. The obstacle avoidance strategy calculates the optimal obstacle avoidance direction of the UAV by

analyzing the positions of all detected faces in the image to ensure the safety of flight. The purpose of this study is to improve the safety of autonomous flight of UAVs in human-dense scenes with the lowest hardware and algorithm costs.

The structure of this paper is arranged as follows: Section II introduces the methods and techniques used in this study, including the detection principle of the YOLOv8 model, the mathematical derivation of the PnP algorithm, the application of IMU sensors, and the specific implementation of sensor fusion. Section III describes the experimental design and scene settings in detail. Section IV shows the experimental results and conducts in-depth discussions. Finally, Section V summarizes the research results and proposes future research directions.

2. Methodology

This section introduces in detail the core technical methods used in this study, including face detection, position calculation and sensor fusion, and the specific implementation of obstacle avoidance strategy.

2.1. Face Detection

The YOLO (You Only Look Once) model has become a research hotspot in the field of visual target detection because of its fast and accurate detection capabilities. YOLOv8 is a new generation of deep learning model launched by the Ultralytics team known for its innovative YOLOv5 model, released on January 10, 2023 [13]. It can not only be used for object detection in images, but also supports tasks such as image classification and object discrimination. YOLOv8 is the result of further improvements based on YOLOv5, significantly optimized for performance and ease of use. Ultralytics has made several upgrades to it, including an optimized backbone network, anchor-free detection heads, and a new loss function, making YOLOv8 more efficient than its predecessor. In addition to these enhancements, YOLOv8 natively supports image classification tasks. Its biggest advantage is that it combines extremely high speed and accuracy while maintaining a lightweight design, ensuring that it can adapt to various hardware platforms and application scenarios [14]. By integrating the global information of the image, YOLOv8 can simultaneously predict the locations and categories of multiple targets in one forward pass. Therefore, YOLOv8 has huge application potential in UAV autonomous obstacle avoidance.

By integrating the global information of the image, YOLOv8 can simultaneously predict the locations and categories of multiple targets in one forward pass. Therefore, YOLOv8 has huge application potential in UAV autonomous obstacle avoidance. In this study, the YOLOv8 model was used for face recognition and five key points were accurately marked: two eyes, nose and two corners of the mouth. In this way, it can provide an effective foundation for subsequent facial feature analysis and related applications, especially in tasks that require precise facial alignment and expression recognition. Taking advantage of the efficient features of YOLOv8, we conducted experiments in a variety of environments and verified its robustness and accuracy under different lighting and angles.

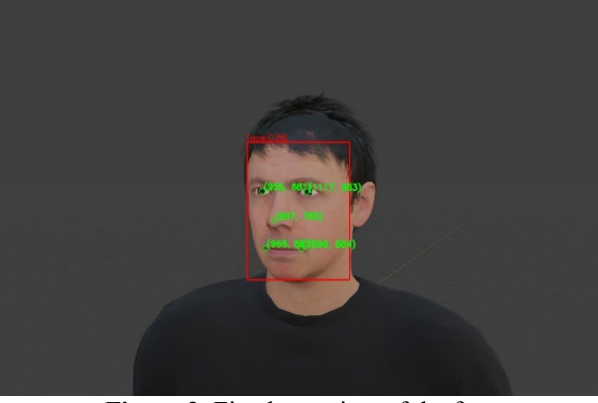


Figure 2. Five key points of the face

2.2. Position calculation and sensor fusion

The PnP (Perspective-n-Point) algorithm is a classic computer vision algorithm used to deduce the six-dimensional pose (i.e., three-dimensional position and orientation) of a camera from two-dimensional image points and corresponding three-dimensional space points. By establishing a mathematical model of the perspective projection relationship between a known three-dimensional space point and its two-dimensional projection point in the image, the PnP algorithm can solve the rotation matrix and displacement vector of the camera, thereby accurately estimating the position and orientation of the camera in three-dimensional space. The algorithm is widely used in fields such as augmented reality [15], autonomous driving [16], and positioning in indoor environments [17,18], providing an accurate estimation of the camera pose, thereby achieving precise positioning and tracking of the target object.

In the pose estimation problem, different PnP algorithms have their own advantages and disadvantages. Choosing the right algorithm is crucial for accuracy and efficiency. The following are several types of PnP algorithms.

- Iterative algorithm based on Levenberg-Marquardt (LM) optimization method: a classic iterative method that finds the optimal solution through continuous iteration. It is suitable for general situations, especially when the initial estimate is good, it can quickly converge to the correct solution.

- EPnP (Efficient Perspective-n-Point): an efficient non-iterative PnP algorithm suitable for processing large-scale 3D point sets. It calculates the pose through a polynomial method with high computational efficiency.

- P3P (Perspective-Three-Point): suitable for pose estimation problems when there are only three 3D points. The algorithm determines a unique solution through three points, but in some cases, multiple solutions may be generated, requiring further disambiguation.

- DLS (Direct Least Squares): a PnP algorithm based on direct least squares that can estimate the pose in a linear model of the position relationship between the camera and the object. It solves the pose by minimizing the cost function.

- UPnP (Universal PnP): a PnP algorithm that does not require initial pose guessing. It calculates the camera pose through a scheme based on singular value decomposition, which is suitable for situations where there is no initial pose estimate.

- AP3P (Algebraic P3P): It is a P3P algorithm based on algebraic methods, suitable for processing a small number of 3D point sets, and obtains a stable solution by optimizing algebraic equations.

Considering that 3D face recognition algorithms with more control points can be introduced in the future, this study uses the EPnP algorithm [19]. The EPnP algorithm is an efficient method for solving PnP problems. The core idea of this algorithm is to represent 3D points as a combination of 4 control points and transform them through this "intermediate medium." Optimization is only performed on the 4 control points. The computational complexity of EPnP is $O(n)$, whereas the complexity of other algorithms is typically $O(n^3)$. For PnP problems with a large number of feature points, EPnP is highly efficient.

In this study, only five feature points on the face are currently extracted, but the number will gradually be expanded to the mainstream 98 feature points to improve the accuracy and robustness of face recognition. Since the EPnP algorithm is more efficient in handling situations with multiple points, selecting it as the method for face pose estimation in this study can significantly enhance the computational speed while maintaining accuracy. This choice lays a solid foundation for future, more complex 3D face recognition tasks.

The detailed process of the EPnP algorithm is as follows:

Notation: A 3D point is denoted by p , such as p_i^w and p_i^c . Control points are denoted by c , such as c_i^w and c_j^c . Points with the superscript w are expressed in the world coordinate system (e.g., p_j^w and c_j^w), while those with c are in the camera coordinate system (e.g., p_i^c and c_j^c).

(1) Control Points and Barycentric Coordinates: The EPnP algorithm first selects four control points in the world coordinate system, denoted by $c_1^w, c_2^w, c_3^w, c_4^w$. These control points can represent any 3D point p_i^w using barycentric coordinates

α_i :

$$p_i^w = \sum_{j=1}^4 \alpha_j c_j^w, \sum_{j=1}^4 \alpha_j = 1 \quad (1)$$

Here, $p_i^w = [X^w, Y^w, Z^w]^T p$ is a 3D point, and $c_i^w = [X^w, Y^w, Z^w]^T$ is a control point. The coefficients α_j are known as homogeneous barycentric coordinates, which express a 3D point as a linear combination of the control points.

(2) Selecting Control Points: The control points are chosen based on the centroid of the reference points. The first control point is:

$$c_1^w = \frac{1}{n} \sum_{i=1}^n p_i^w \quad (2)$$

The remaining control points are determined by solving an eigenvalue problem:

$$c_j^w = c_1^w + \lambda_{c,j-1}^2 V_{c,j-1}, j = 2,3,4 \quad (3)$$

(3) Transforming Control Points to the Camera Coordinate System: Using the intrinsic matrix K and the extrinsic parameters $[R \ t]$, the control points in the world coordinate system are transformed to the camera coordinate system:

$$c_j^c = [R \ t][c_j^w] \quad (4)$$

(4) Setting up Linear Equations: By using the projection equation and the weighted sum of control points, a system of linear equations can be established:

$$\forall i, \mathbf{w}_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = K \mathbf{p}_i^c = K \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^c \quad (5)$$

After eliminating the depth term w_i , two linear equations can be derived:

$$M\mathbf{X} = 0, \text{ where } \mathbf{X} = [c_1^c, c_2^c, c_3^c, c_4^c]^T \quad (6)$$

Here, \mathbf{X} contains the coordinates of the control points in the camera coordinate system.

(5) Solving the Linear System: The solution of $M\mathbf{X} = 0$ gives the coordinates of the control points in the camera coordinate system. This is typically achieved by finding the null space of MTM .

(6) Non-Linear Optimization: Although the linear solution provides an initial estimate, non-linear optimization is usually applied to minimize their projection error and improve accuracy:

$$\text{Error}(\beta) = \sum_{(i,j) \text{ s.t. } i < j} \left(\|\mathbf{c}_i^c - \mathbf{c}_j^c\|^2 - \|\mathbf{c}_i^w - \mathbf{c}_j^w\|^2 \right)^2 \quad (7)$$

(7) Camera Pose Estimation: Finally, the camera's rotation matrix R and translation vector t are estimated using the Iterative Closest Point (ICP) algorithm or other pose estimation methods.

When using the PnP algorithm, the 6D posture of the target point can also be inferred from the fixed position of the camera. In this study, the PnP algorithm is used to calculate the position of the face detected by the drone camera relative to the drone in three-dimensional space. However, in practical applications, since the two-axis gimbal of the drone is generally in a state where the Roll axis is self-stabilized (i.e., automatically keeps level) but the Pitch axis is at an angle with the horizontal plane, the camera's viewing angle may produce a pitch angle offset, thereby affecting the accuracy of the distance measurement. In order to solve this problem, this study introduces the concept of sensor fusion, and further corrects the results calculated by the PnP algorithm by combining the gimbal pitch axis angle data provided by the IMU sensor.

The IMU sensor is an inertial measurement unit that integrates an accelerometer, a gyroscope, and a magnetometer, and can provide the attitude information of the drone in real time. In this study, the pitch axis angle data provided by the IMU sensor is used to correct the calculation results of the PnP algorithm. Specifically, after the PnP algorithm calculates the preliminary position result, the system will combine the data of the IMU sensor and correct the actual position and posture of the face through matrix transformation, thereby obtaining a more accurate face distance measurement result.

2.3. Obstacle avoidance strategy

(1) Preliminary determination of obstacle avoidance direction: The obstacle avoidance strategy of drones is an

important part of ensuring the safety of their autonomous flight. In this study, the system will automatically trigger the obstacle avoidance strategy based on the face recognized in the picture. The obstacle avoidance system implements a calculation process of three-dimensional obstacle avoidance points based on the detected target point data. The initial obstacle avoidance direction is determined by calculating the uniformity and density of the distribution of faces relative to the drone in three-dimensional space. Specifically, if the distribution of faces in a certain direction is sparse, then this direction will be considered a relatively safe obstacle avoidance direction, and this safe direction will be set as the movement direction of the drone. On the contrary, if the distribution of faces in a certain direction is dense, then this direction will be considered a high-risk area, and the drone should avoid flying in this direction. If there is a dense distribution of faces in all directions, it is considered that there is no safe direction, and the drone will stop moving immediately. After calculating the avoidance direction, a verification step will be performed.

This obstacle avoidance system ensures the autonomous navigation capability of the drone in a complex environment by calculating three-dimensional obstacle avoidance points based on the detected target point data. In order to determine the initial obstacle avoidance direction, the spatial distribution characteristics of the face points are first defined.

Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of detected target points in 3D space, where each target point is represented as $p_i = (x_i, y_i, z_i)$. We use *uniformity* and *density* metrics to evaluate the distribution of faces, defined as follows:

- Uniformity metric U :

$$U = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2} \quad (8)$$

where \bar{x} , \bar{y} , \bar{z} are the mean coordinates of the target points.

- Density metric D :

$$D = \frac{1}{n} \sum_{i=1}^n ((x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2) \quad (9)$$

By calculating the uniformity U and density D , we can determine the safe direction of movement based on the thresholds T_U and T_D :

- If $D < T_D$, the direction is considered safe, denoted as \vec{v}_{safe} . - If $D \geq T_D$, the direction is deemed a high-risk area.

Constraint on obstacle avoidance direction: To ensure safe drone flight, the angle θ between the avoidance direction and the horizontal plane must be positive. The angle θ is computed as:

$$\theta = \arctan \left(\frac{v_z}{\sqrt{v_x^2 + v_y^2}} \right) \quad (10)$$

where $\vec{v}_{\text{safe}} = (v_x, v_y, v_z)$ is the safe avoidance direction vector. We require that:

$$\theta > 0 \Rightarrow vz > 0 \quad (11)$$

If all directions are found to be high-risk, the drone halts its movement.

(2) *Validation Step*: After determining the preliminary obstacle avoidance direction, a cylinder-based validation strategy is applied. The simple motion path of a quadrotor drone with Yaw-axis rotation can be approximated as a cylinder with the drone's circumference as the base. Let R be the cylinder's radius (safety distance threshold), and c be the center position of the drone. The cylindrical path of the drone can be described by the following inequality:

$$\sqrt{(x - c_x)^2 + (y - c_y)^2} \leq R \quad (12)$$

The algorithm iteratively checks whether all target points p_i are outside the cylinder:

$$\forall p_i \in P : \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} > R \quad (13)$$

If all target points p_i satisfy the above condition, the program confirms the current avoidance point as valid and exits the loop. Otherwise, the drone must adjust its movement direction and recalculate potential avoidance points.

3. EXPERIMENT

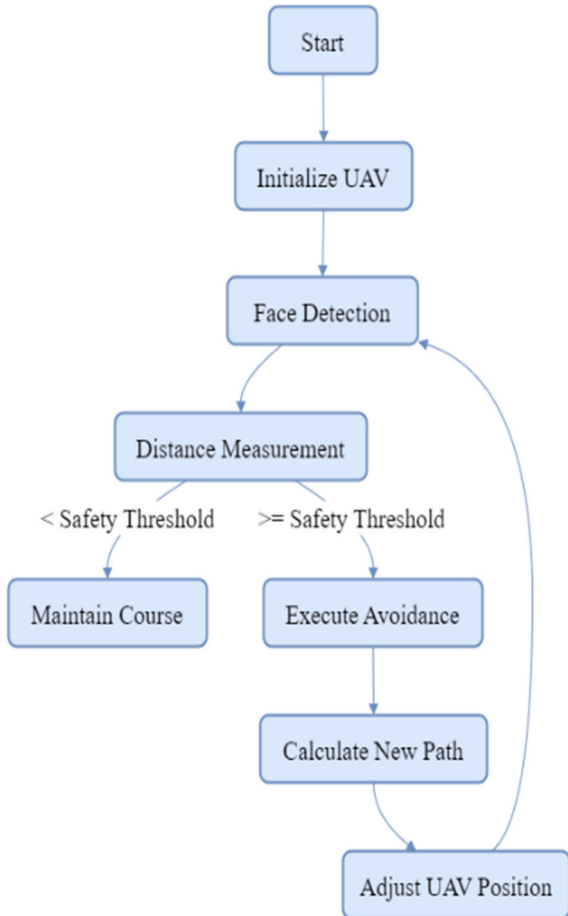


Figure 3. Experimental process

3.1. Experimental Setup

To evaluate the proposed drone obstacle avoidance system based on face detection, this study designed extensive experiments in a virtual simulation environment, ensuring safety and controllability while testing under realistic scenarios.

Platform and Environment:

- UAV Platform: DJI Mavic series with onboard 1080p/4K camera and IMU sensors.
- Virtual Environment: Developed in Blender, including narrow streets with complex obstacles (e.g., buildings, moving pedestrians).
- Face Models: Diverse virtual characters with varied skin tones and lighting conditions for robustness testing.
- Flight Parameters: Altitudes of 1, 2, 3 meters and speeds of 0.5, 1.0, 1.5 m/s.

3.2. Evaluation Metrics

- **Mean Average Precision (mAP)**: Evaluates the face detection accuracy using the PASCAL VOC standard.
- **Absolute Position Error (APE)**: Quantifies the difference between estimated and true positions:

$$APE = \frac{1}{N} \sum_{i=1}^N \|\hat{P}_i - P_i\| \quad (14)$$

where \hat{P}_i is the estimated position, P_i is the true position, and N is the number of test samples.

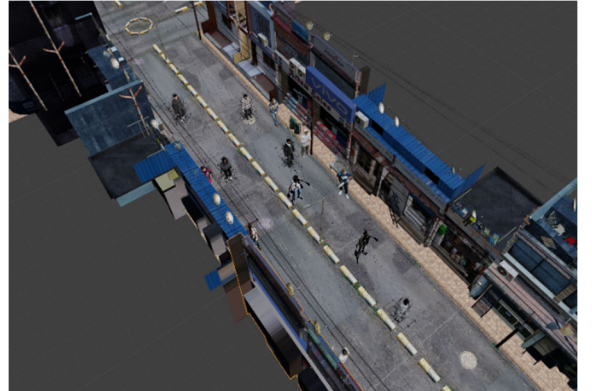


Figure 4. Virtual environment setup

- **Success Rate**: The percentage of successful obstacle avoidance actions:

$$Success Rate = \frac{Number\ of\ Successes}{Total\ Tests} \times 100\% \quad (15)$$

- **Response Time**: Time taken for the system to react after detecting an obstacle, measured in milliseconds (ms).

3.3. Experimental Results

Table I summarizes the key metrics obtained under varying flight conditions.

3.4. Result Analysis

(1) *Face Detection Performance*: The highest mAP (0.87) was achieved at a flight altitude of 2 meters. This suggests an optimal balance between view angle and image clarity, as shown in Table I.

Table I. Performance Evaluation Under Different Flight Conditions

Condition	mAP	APE (m)	Success Rate (%)	Response Time (ms)
Altitude 1m	0.85	0.15	95	200
Altitude 2m	0.87	0.12	93	180
Altitude 3m	0.82	0.18	90	220
Speed 0.5 m/s	0.83	0.17	92	210
Speed 1.0 m/s	0.86	0.14	94	190
Speed 1.5 m/s	0.81	0.20	88	230

A slight decrease in detection accuracy (mAP = 0.82) at 3 meters altitude indicates reduced image resolution due to increased distance from the target.

(2) *Position Estimation Accuracy:* The APE values increase with flight speed, with the highest error (0.20 m) at 1.5 m/s, indicating challenges in maintaining visual-inertial fusion accuracy under high-speed motion.

(3) *Obstacle Avoidance Effectiveness:* The system showed high obstacle avoidance success rates, particularly at lower altitudes and speeds. The highest success rate (95%) was achieved at an altitude of 1 meter. However, performance decreased at higher speeds (1.5 m/s), with the success rate dropping to 88%, suggesting potential areas for optimization.

(4) *System Response Time:* The fastest reaction time (180 ms) was observed at an altitude of 2 meters, where the drone had optimal visibility and processing efficiency. The response time increased significantly at a speed of 1.5 m/s (230 ms), potentially impacting real-time obstacle avoidance.

4. Conclusion

The obstacle avoidance method for drone face recognition and distance measurement based on YOLOv8, PnP algorithm and sensor fusion proposed in this study demonstrates its application potential in autonomous flight safety assurance. Compared with traditional obstacle avoidance methods, the system in this study can not only perform efficient face detection, but also accurately measure the distance between the face and the drone, thereby effectively avoiding obstacles. However, the experimental results also reveal the limitations of the system in some extreme scenarios. For example, in an environment with extremely complex or drastic changes in light, the detection accuracy of the YOLOv8 model may decrease. In this case, the detection error may affect the subsequent distance measurement and obstacle avoidance decision.

In addition, the data of the IMU sensor may drift to a certain extent when the drone moves at high speed, which may also lead to an increase in the measurement error.

Future research can be improved in the following aspects. First, algorithms for 3D face recognition with more control points can be introduced, such as Google's Mediapipe. The use of the EPNP algorithm, which is very efficient for PnP problems with more feature points, makes it possible to calculate the face posture and position of more control points. Secondly, more advanced sensor fusion algorithms, such as particle filters, can be combined to further correct the errors in IMU data. In addition, more intelligent obstacle avoidance strategies can be explored, and the tracking of the motion trajectory of the face can be considered. If the detected face has a clear motion trend, the system will combine its motion

trajectory to predict its future position, thereby optimizing the obstacle avoidance path. To achieve this goal, the extended Kalman filter (EKF) can be used [20]. When processing nonlinear visual measurement data, the algorithm optimizes the obstacle state estimation through prediction and update steps, thereby effectively supporting obstacle avoidance and waypoint tracking tasks. In this way, the drone can not only avoid static obstacles, but also effectively deal with dynamic targets.

References

- [1] Moskowitz, E. E., Siegel-Richman, Y. M., Hertner, G., & Schroepel, T. (2018). Aerial drone misadventure: A novel case of trauma resulting in ocular globe rupture. *American Journal of Ophthalmology Case Reports*, 10, 35–37.
- [2] Peng, Y., Qu, D., Zhong, Y., Xie, S., Luo, J., & Gu, J. (2015). The obstacle detection and obstacle avoidance algorithm based on 2-D lidar. 2015 IEEE International Conference on Information and Automation, 1648–1653.
- [3] Borenstein, J., & Koren, Y. (1988). Obstacle avoidance with ultrasonic sensors. *IEEE Journal on Robotics and Automation*, 4(2), 213–218.
- [4] Gupta, N., Makkar, J. S., & Pandey, P. (2015). Obstacle detection and collision avoidance using ultrasonic sensors for RC multirotors. 2015 International Conference on Signal Processing and Communication (ICSC), 419–423.
- [5] Suherman, S., Putra, R. A., & Pinem, M. (2020). Ultrasonic Sensor Assessment for Obstacle Avoidance in Quadcopter-based Drone System. 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), 50–53.
- [6] Nieuwenhuisen, M., Droschel, D., Schneider, J., Holz, D., Labe, T., & Behnke, S. (2013). Multimodal obstacle detection and collision avoidance for micro aerial vehicles. 2013 European Conference on Mobile Robots, 7–12.
- [7] Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., & Hebert, M. (2012). Learning Monocular Reactive UAV Control in Cluttered Natural Environments (No. arXiv:1211.1690). arXiv.
- [8] Mori, T., Scherer, S. (2013) First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In: IEEE International Conference on Robotics and Automation. 1750-1757.
- [9] Souhila, K., & Karim, A. (2007). Optical Flow Based Robot Obstacle Avoidance. *International Journal of Advanced Robotic Systems*, 4(1), 2.
- [10] Gosiewski, Z., Ciesluk, J., & Ambroziak, L. (2011). Vision-based obstacle avoidance for unmanned aerial vehicles. 2011 4th International Congress on Image and Signal Processing, 2020–2025.
- [11] Aguilar, W., Casaliglla, V., & P^olit, J. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.

- [12] Yang, X., Chen, J., Dang, Y., Luo, H., Tang, Y., Liao, C., Chen, P., & Cheng, K.-T. (2021). Fast Depth Prediction and Obstacle Avoidance on a Monocular Drone Using Probabilistic Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems*, 22(1), 156–167.
- [13] Glenn Jocher, Ayush Chaurasia, Alex Stoken, et al. (2022) ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). Zenodo.
- [14] Sohan, M., Sai Ram, T., & Rami Reddy, Ch. V. (2024). A Review on YOLOv8 and Its Advancements. In I. J. Jacob, S. Piramuthu, & P. Falkowski-Gilski (Eds.), *Data Intelligence and Cognitive Informatics*. Springer Nature, Singapore. 529–545.
- [15] Makibuchi, N., Kato, H., & Yoneyama, A. (2013). Vision-based robust calibration for optical see-through head-mounted displays. 2013 IEEE International Conference on Image Processing, 2177–2181.
- [16] Wang, W., Wang, F., Liu, P., He, Y., & Zhang, X. (2016). A monovision-based 3D pose estimation system for vehicle behavior prediction. 2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 1–6.
- [17] Deretey, E., Ahmed, M. T., Marshall, J. A., & Greenspan, M. (2015). Visual indoor positioning with a single camera using PnP. 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1–9.
- [18] Kubířkov^á, H., Jedlička, K., Fiala, R., & Beran, D. (2020). Indoor Positioning Using PnP Problem on Mobile Phone Images. *ISPRS International Journal of Geo-Information*, 9(6), 368.
- [19] Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2), 155–166.
- [20] Watanabe, Y., Calise, A., & Johnson, E. (2007). Vision-Based Obstacle Avoidance for UAVs. *AIAA Guidance, Navigation and Control Conference and Exhibit*. Hilton Head, South Carolina.