Article

MoNA Bench: A Benchmark for Monocular Depth Estimation in Navigation of Autonomous Unmanned Aircraft System

Yongzhou Pan [1,2,†], Binhong Liu [1], Zhen Liu [1], Hao Shen [1], Jianyu Xu [1], Wenxing Fu [1] and Tao Yang [1,*]

[1] Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China; pan.yongzhou@u.nus.edu (Y.P.); binhongliu@mail.nwpu.edu.cn (B.L.); liu.zhen@mail.nwpu.edu.cn (Z.L.); xjy8@mail.nwpu.edu.cn (J.X.); wenxingfu@nwpu.edu.cn (W.F.)
[2] School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China
[*] Correspondence: yangtao@nwpu.edu.cn
[†] Current address: College of Design and Engineering, National University of Singapore, Singapore 117575, Singapore.

Abstract: Efficient trajectory and path planning (TPP) is essential for unmanned aircraft systems (UASs) autonomy in challenging environments. Despite the scale ambiguity inherent in monocular vision, characteristics like compact size make a monocular camera ideal for micro-aerial vehicle (MAV)-based UASs. This work introduces a real-time MAV system using monocular depth estimation (MDE) with novel scale recovery module for autonomous navigation. We present MoNA Bench, a benchmark for Monocular depth estimation in Navigation of the Autonomous unmanned Aircraft system (MoNA), emphasizing its obstacle avoidance and safe target tracking capabilities. We highlight key attributes—estimation efficiency, depth map accuracy, and scale consistency—for efficient TPP through MDE.

Keywords: UAV; autonomous navigation; monocular depth estimation; path planning; flight safety; target tracking

check for updates

Citation: Pan, Y.; Liu, B.; Liu, Z.; Hao, S.; Xu, J.; Fu, W.; Yang, T. MoNA Bench: A Benchmark for Monocular Depth Estimation in Navigation of Autonomous Unmanned Aircraft System. *Drones* **2024**, *8*, 66. https://doi.org/10.3390/drones8020066

Academic Editor: Oleg Yakimenko

Received: 31 December 2023
Revised: 9 February 2024
Accepted: 12 February 2024
Published: 16 February 2024

1. Introduction

Efficient trajectory and path planning (TPP) plays a fundamental role in defining the autonomy of unmanned aircraft systems (UASs) in adversarial environments. To achieve this capability, UASs commonly integrate sensors such as GPS, LiDAR, stereo camera, and RGB-D camera, which are proficient at directly capturing metric depth information from the geographic coordinates or the surrounding environment. In contrast, monocular vision lacks depth measurement capabilities, presenting challenges such as scale ambiguity.

Despite encountering the challenges, the monocular camera remains the sensor of choice for micro-aerial vehicles (MAVs), which are often deployed in narrow indoor environments, confronting substantial resource constraints and strict payload limitations. In this context, the monocular camera stands out as the optimal selection due to its unique characteristics, including compact size, lightweight design, low energy consumption, and the ability to provide rich information content such as texture and semantics. Consequently, if reliable depth information becomes attainable, the monocular camera will serve as the preferred perception device for MAV-based UAS, offering a balanced solution that addresses resource constraints while delivering valuable visual information.

Efficient TPP is indispensable for autonomous UASs, ensuring robust localization and navigation capabilities, particularly in tasks like real-world target search and tracking. The achievement of efficient TPP in monocular UASs relies on the dependable depth recovery from monocular images and the reconstruction of the surrounding 3D environment, facilitated by recent advancements in computer vision. Through deep learning-based monocular depth estimation (MDE) methods, it is able to extract dense depth maps from monocular image sequences. The cutting-edge MDE algorithms currently exhibit outstanding performance, as evidenced by evaluations on widely recognized benchmarks, showcasing exceptional prediction accuracy.

In this work, we develop a real-time monocular MAV system (as shown in Figure 1), utilizing monocular depth estimation to support efficient trajectory and path planning. Building upon our prior research [1], we introduce an innovative scale recovery module to calculate the scale factor of the environment, reconstruct the 3D scenario, and assess the performance of the integrated MDE algorithms. With the acquired scale information, the improved framework not only excels in autonomous obstacle avoidance but also supports safe target tracking—a task demanding a precise metric scale. Our experiments validate the effectiveness of the proposed system, highlighting the significant attributes that MDE algorithms should possess for efficient TPP: estimation efficiency, depth map accuracy, and scale consistency.

The contributions of this paper are threefold:

- We design a real-time monocular MAV-based unmanned aircraft system. Our system accomplishes efficient path planning to enable the effective implementation of autonomous obstacle avoidance and safe target tracking.
- We introduce MoNA Bench, a benchmark for monocular depth estimation in autonomous navigation for unmanned aircraft systems. We develop a series of deployable performance evaluation experiments for monocular depth estimation and identify significant attributes that MDE algorithms should possess for efficient trajectory and path planning.
- To benefit the community, we release the complete source code of the proposed benchmark at: https://github.com/npu-ius-lab/MoNA-Bench (accessed on 31 December 2023).

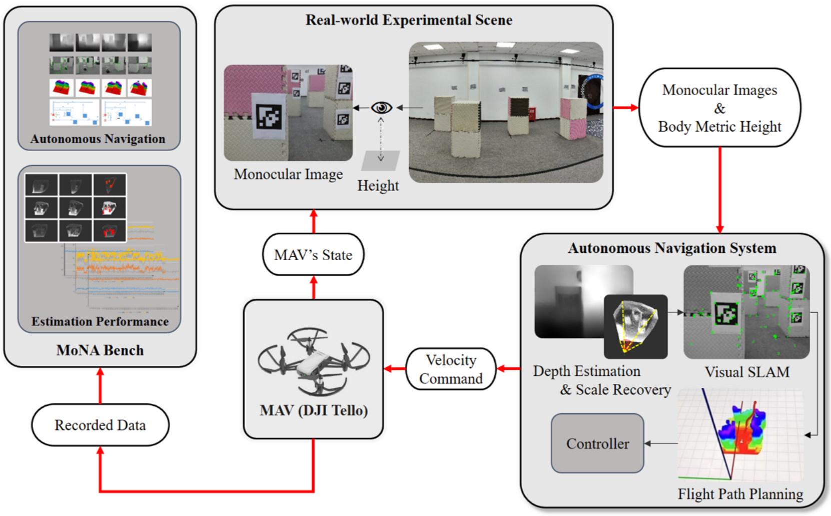


Figure 1. The basic framework of the proposed MoNA Bench.

The paper is organized as follows: In Section 2, we briefly reviewed key technologies employed in system construction, including monocular depth estimation, trajectory planning, pose estimation, etc. Section 3 provides an in-depth explanation of our approach. In Section 4, we present the results, evaluation, and analysis of our experiments. Lastly, Section 5 summarizes the paper and outlines prospects for improvement.

2. Related Work

2.1. Monocular Depth Estimation

Monocular depth estimation aims to recover depth information from images captured by monocular cameras, predominantly relying on deep learning. Various algorithms have emerged in this field, with supervised and self-supervised learning being the primary approaches. Supervised methods, while achieving superior accuracy and scene perception by training on precise ground-truth depth maps, face limitations due to the need for costly and susceptible calibrated depth sensors. On the other hand, self-supervised methods, utilizing ego-motion as the supervisory signal, eliminate the requirement for ground-truth depth maps. However, they lack an inherent understanding of the scene scale, introducing dynamic scale disparities between predicted and actual depth maps. This scale ambiguity can significantly impact the accuracy of depth estimation, particularly in real-world applications.

Eigen et al. [2] pioneered deep learning-based MDE in 2014, training their model through supervised learning. Alhashim et al. [3] developed an encoder-decoder architecture with skip connections to capture object boundaries faithfully. They employed a pre-trained truncated DenseNet-169 [4] as the encoder through transfer learning, achieving promising results on datasets like NYUv2 [5] and KITTI [6]. Ranftl et al. [7] proposed a cross-dataset model training method to improve their model's generalization capabilities.

The self-supervised learning method has gained considerable attention for its independence from ground truth, with notable research efforts aiming to address its inherent scale ambiguity. PackNet-SfM [8] introduced weak velocity supervision from IMU, enabling their network to acquire real-world scale understanding. Bian et al. [9] developed a geometry consistency loss and a self-discovered mask to guarantee depth prediction consistency across frames. For indoor performance improvement, an auto-rectify network [10] was further designed, incorporating novel loss functions to automatically rectify images during training.

2.2. Efficient Flight Trajectory Planning

Hard-constrained methods and gradient-based optimization methods represent the primary approaches to flight trajectory planning. Hard-constrained methods often yield trajectories in close proximity to obstacles, posing challenges for efficient flight. In contrast, gradient-based trajectory optimization methods treat trajectory generation as a nonlinear optimization problem, balancing considerations of smoothness, safety, and dynamic feasibility.

Fast-Planner [11] utilized topological path parallel optimization to address local minima issues, facilitating precise online planning for local obstacle avoidance trajectories. Building upon this foundation, Fast-Tracker [12] introduced an agile target active safety tracking system for unmanned aerial vehicles (UAVs). The system incorporated modules for target motion prediction and trajectory planning, allowing for the derivation of time-space optimal and collision-free safety tracking trajectories for the UAV. Additionally, it maintained continuous target position prediction even after losing track of the target.

2.3. Target Detection and Pose Estimation

While natural features serve a broad range of applications, artificial features such as QR codes remain essential in the development and testing phases of robot systems. These features offer consistent and dependable target information, facilitating the assessment of robot systems in real-world environments.

ARToolkit [13] emerged as one of the pioneering systems for artificial feature tracking; however, its robustness faced challenges, especially in varying illumination conditions. AprilTag [14,15] presented a high-speed visual fiducial system that exhibited remarkable robustness against factors like lighting variations, occlusions, and lens distortion. This system employed 2D bar code style features (tags) and efficiently achieved full 6 degrees of freedom (DoF) localization of tags from a single image.

2.4. Monocular Vision-Based Autonomous Obstacle Avoidance

Significant strides have been made in autonomous obstacle avoidance technology for UAVs based on monocular vision. Currently, depth learning occupies a predominant role in monocular vision-based obstacle avoidance systems. Some impressive research [16–18] established their monocular UAV autonomous obstacle avoidance systems with similar frameworks. These systems typically consisted of three modules: monocular depth estimation, autonomous collision avoidance, and UAV velocity control. To enhance performance, these methods used RGB videos recorded from real-world scenarios for model training. In contrast, the system proposed in our earlier study [1] did not require captured scene data. Instead, it employed pre-trained models trained on large public datasets like NYUv2 [5] to estimate depth maps for unknown scenes, without introducing scene-specific information during training.

3. Approach

3.1. Overview

Our research utilized the DJI RoboMaster TT Tello Talent (RYZE Tech Co., Ltd., Shenzhen, China). MAV as our hardware platform, weighing only around 80 g (including propellers and battery). It features a 5 MP monocular camera with an 82.6° field of view and can record videos at up to 720 P/30 fps. Additionally, it is equipped with a time-of-flight (ToF) infrared distance sensor for precise body height detection. The MAV can communicate with the ground server via its built-in 2.4 GHz WiFi module and transmit stable video streaming within a 100 m range using 5 GHz WiFi when its expansion module is mounted. The details of our MAV are depicted in Figure 2.

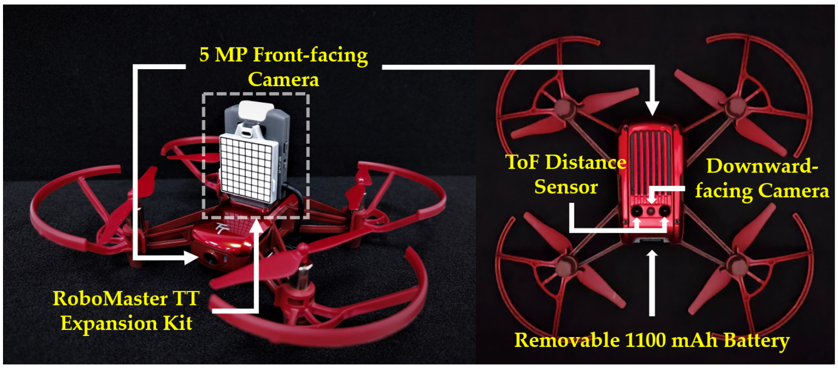


Figure 2. The built-in sensor layout and the appearance of the MAV with its extension kit.

Ensuring stable autonomous navigation for monocular MAVs in unknown indoor environments requires tackling the scale ambiguity in monocular depth estimation as the primary challenge. Addressing this issue is essential to maintain the consistency between the environmental scale and the scale of predicted target poses for constructing safe target tracking. With reliable perception established, MAV localization and path planning can be performed using RGB images and estimated depth maps. Lastly, the MAV can seamlessly track the generated path through appropriate controllers.

Based on the outlined processing sequences, we have implemented an unmanned aircraft system depicted in Figure 3. The proposed system consists of four key modules: depth estimation, autonomous navigation, pose estimation, and velocity control. Within the autonomous navigation module, two fundamental functionalities are encapsulated: obstacle avoidance and target tracking. The procedural workflow of the system is summarized below:

1. System Connection. The MAV establishes a wireless connection with the ground server through WiFi, transmitting a continuous stream of RGB images. Upon receiving the activation command, the MAV initiates takeoff and maintains a stable hover at an altitude of approximately 0.9 m.
2. Depth Estimation. The depth estimation network operates on the ground server, integrating extensively validated networks like MonoDepth, MiDaS, and SC-DepthV2. To recover metric information from the physical world for subsequent navigation, a novel scene scale recovery submodule is developed. This submodule incorporates MAV height information and executes recovery through four steps: depth map to

point cloud conversion, point cloud ground segmentation, point cloud coordinate transformation, and scale factor calculation. Additionally, to facilitate subsequent computations, both RGB images and dense depth maps are resized to a resolution of 640 × 480.

3. Autonomous Navigation. To enable autonomous MAV navigation, a visual SLAM (Simultaneous Localization and Mapping) submodule localizes the camera's 6 degrees of freedom (DoF) pose through processed RGB images, dense depth maps, and camera intrinsics. Subsequently, the coordinate transformation submodule converts the estimated camera pose to MAV body odometry. Following this, the flight trajectory is generated with two distinct options:
   - Obstacle Avoidance. In this mode, the system employs RGB images, predicted dense depth maps, camera intrinsics, and MAV body odometry to generate a 3D occupancy grid map. This map serves as the foundation for constructing a local trajectory and establishing a front tracking point when a flying target is specified.
   - Target Tracking. In this mode, alongside RGB images, predicted dense depth maps, camera intrinsics, and MAV body odometry, the system requires a subscription to the target position obtained from the 6-DOF pose estimation module to generate the specialized trajectory for tracking the designated target.
4. Pose Estimation. Accurate target tracking hinges on determining the real-world positions of designated targets. In our system, AprilTag has been selected as the tracking target due to its reliable and distinctive visual features. The target's pose estimation is accomplished by analyzing RGB image sequences, allowing the system to precisely locate and track the designated target throughout its movements.
5. Velocity Control. Upon generating the flight trajectory, the system provides adaptability by offering a choice between two controllers to calculate the MAV's velocity command: a PID controller and a path-following controller. This flexibility ensures efficient control and responsiveness tailored to the specific requirements of the mission or task at hand.

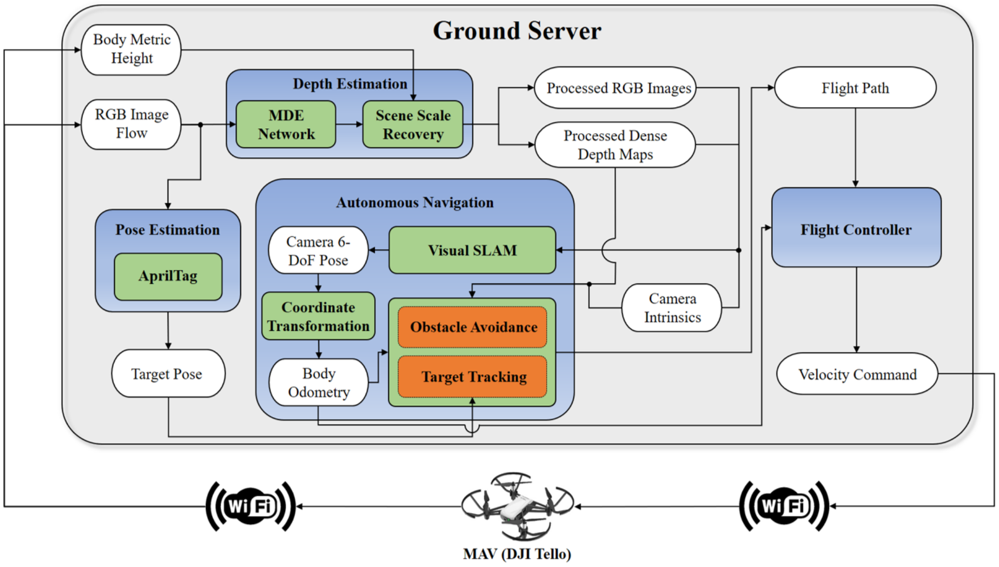


Figure 3. The pipeline of the proposed system.

3.2. Monocular Depth Estimation

Accurate depth predictions form the foundation for monocular UASs to perceive the surrounding environment. In this work, we explore and compare both supervised and self-supervised learning algorithms, including MonoDepth [3], MiDaS [7], and SC-DepthV2 [10], to understand their performance and limitations in MDE for UASs. After obtaining high-quality predicted depth maps, we incorporate MAV body height information for real-time metric scale recovery.

3.2.1. MDE Algorithms

MonoDepth. MonoDepth is a supervised learning algorithm designed to overcome the common issue of blurry approximations in low-resolution depth estimation. The algorithm introduces a simple network architecture based on transfer learning, featuring fewer network parameters and training iterations. Simultaneously, it excels in capturing object boundaries, enabling high-precision and high-quality depth estimation. The loss function of MonoDepth is defined by Equation (1):

$$L(d, \hat{d}) = \lambda L_{depth}(d, \hat{d}) + L_{grad}(d, \hat{d}) + L_{SSIM}(d, \hat{d}) \tag{1}$$

where d represents the groundtruth depth map, and $\hat{d}$ represents the predicted depth map. The pixel-wise loss L_{depth} is calculated based on the depth values. To incorporate the image gradient of the depth image, L_{grad} is introduced. Additionally, L_{SSIM} is a structural similarity (SSIM) [19] term, which is commonly employed in image reconstruction tasks.

MiDaS. Improving the generalization of supervised learning-based MDE algorithms demands abundant and diverse training data. However, the distinct characteristics among existing depth datasets present significant challenges for cross-dataset training. To enable the algorithm's adaptation to a broad spectrum of dynamic and diverse environments, Ranftl et al. [7] proposed a model training strategy, facilitating the mixing of data from multiple datasets during training. Equation (2) defines the loss function of MiDaS:

$$L_l = \frac{1}{N_l} \sum_{n=1}^{N_l} L_{ssi}(\hat{\boldsymbol{d}}^n, (\hat{\boldsymbol{d}}^*)^n) + \alpha L_{reg}(\hat{\boldsymbol{d}}^n, (\hat{\boldsymbol{d}}^*)^n) \tag{2}$$

where $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{d}}^*$ represent the predicted and ground-truth depth maps, respectively; $L_{ssi}(\hat{\boldsymbol{d}}, \hat{\boldsymbol{d}}^*)$ is incorporated into the loss function to ensure scale and shift invariance; To adapt to the disparity space, $L_{reg}(\hat{\boldsymbol{d}}, \hat{\boldsymbol{d}}^*)$ is introduced as a multi-scale, scale-invariant gradient matching term; The training set size is defined as N_l, and α is a hyperparameter with a value of 0.5.

SC-DepthV2. Self-supervised MDE algorithms rely on unlabeled videos for training, simplifying the data collection process compared to supervised methods. However, without proper scale constraints, these algorithms may produce scale-inconsistent depth estimation results, leading to ambiguity in per-frame scale and posing challenges for providing camera trajectories over long video sequences.

SC-Depth is a self-supervised MDE algorithm and introduces a geometric consistency loss to guarantee global scale consistency in depth predictions. Additionally, the algorithm incorporates an induced self-discovered mask to handle unidentified moving objects and occlusions in the scene, ensuring the validity of the underlying static scene assumption in geometric image reconstruction. SC-Depth formulates its loss function as defined in Equation (3):

$$L = \alpha L_P^M + \beta L_S + \gamma L_G \tag{3}$$

where L_P^M represents the weighted photometric loss by a self-discovered mask M, and an SSIM term similar to MonoDepth is included. L_S stands for the smoothness loss, while L_G is a geometric consistency loss to ensure scale consistency. The weights for these losses are determined by hyperparameters $[\alpha, \beta, \gamma]$.

Furthermore, for improved indoor performance, Bian et al. proposed an auto-rectify network SC-DepthV2. This network automatically rectifies images affected by complex camera motion in an end-to-end fashion, addressing the challenge posed by rotational motion acting as noise during training.

3.2.2. Scale Recovery

The recovery of scale information is crucial for achieving precise perception of spatial scenes, serving as a key element in ensuring consistent scale for autonomous obstacle avoidance and target tracking. In our UAS, this is accomplished through a series of steps.

Depth map to point cloud conversion. Converting a depth map into a point cloud involves associating each pixel in the depth map with a corresponding 3D point in the real-world coordinate system. For any point $\boldsymbol{P}$ with homogeneous pixel coordinates $[u, v, 1]^{\mathrm{T}}$ defined in camera coordinate, and known camera intrinsic matrix $\boldsymbol{K}$, the relationship is expressed as:

$$Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{KP} \tag{4}$$

According to Equation (4), with the known camera intrinsics, the corresponding point cloud can be derived from the predicted depth map. The conversion from a depth map to a point cloud can be effortlessly implemented by calling ROS built-in package.

Point cloud ground segmentation. This step is primarily accomplished using the RANSAC (Random Sample Consensus) algorithm. RANSAC is a robust estimation method that iteratively selects a set of samples from the dataset, fits a model based on these samples, and evaluates the fit of other data points to this model. Notably, RANSAC can operate directly on raw point cloud data without the need for 2D plane projection, making it a widely used technique for segmenting the ground plane in 3D space.

A fundamental prerequisite for applying the RANSAC algorithm to point cloud ground segmentation is the abundance of ground information in the scene. In practical applications, the substantial volume of raw point cloud data generated from depth map conversion may hinder computational efficiency, resulting in increased computational iterations. To optimize algorithm performance and system operational speed while maintaining computational accuracy, it is advisable to implement random down-sampling of the point cloud. The pseudocode for the RANSAC-based ground plane segmentation algorithm is outlined in Algorithm 1.

Point cloud coordinate transformation. The ground segmentation of point clouds occurs in the camera coordinate system, while the height information of the MAV is defined in the body coordinate system. To recover the scale factor from the extracted ground point cloud, a coordinate transformation should be applied to convert the point cloud from the camera frame to the body frame. Define the camera extrinsics as the transformation matrix $\boldsymbol{T}$, which includes the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. To transform the coordinates of a ground point $\boldsymbol{P}$ from the camera coordinate system to the body frame coordinate system, assuming the true physical world coordinates of point $\boldsymbol{P}$ in the body frame are $[X_b, Y_b, Z_b]^{\mathrm{T}}$, denoted as the vector $\boldsymbol{P_b}$, according to the basic rules of 3D rigid body coordinate Euclidean transformation, we have:

$$\begin{bmatrix} \boldsymbol{P} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{P_b} \\ 1 \end{bmatrix} = \boldsymbol{T}\begin{bmatrix} \boldsymbol{P_b} \\ 1 \end{bmatrix} \tag{5}$$

Scale factor calculation. Through point cloud ground segmentation and point cloud coordinate transformation, we obtained the relative position of the ground point cloud in the body frame. Let the total number of points in the ground point cloud be n, and the z-axis coordinate in the body frame for each point be Z_b^m, where m represents the m-th point in the ground point cloud. The relative height h_R is defined as follows:

$$h_R = \frac{1}{n}\sum_{m=1}^{n} Z_b^m \tag{6}$$

Defining the actual height obtained by the onboard ToF infrared distance sensor as the metric height h_M, the scale factor s can be expressed as:

$$s = \left| \frac{h_M}{h_R} \right| \tag{7}$$

By following the aforementioned four steps, we compute the scale factor bridging real-world depth and monocular estimated depth, thereby achieving monocular depth estimation in metric scale. Figure 4 visually illustrates the changes in the point cloud before and after scale recovery.

Algorithm 1: RANSAC-based Point Cloud Ground Plane Segmentation

```
Data: $\mathcal{N}$ - Original point cloud ;
      $m$ - Downsample size ;
      $d$ - Distance threshold ;
      $k$ - Maximum iterations ;
      $\vec{x}$ - Normal vector of the ground plane;
      $s_{min}$ - Threshold for normal vector similarity.
Result: $\mathcal{N}_{ground}$ - Ground points.
$\mathcal{N}_{sampled}$ = Randomly sample $m$ points from the original point cloud $\mathcal{N}$ ;
mostNum = 0;
betterModel = null ;
// Ground plane parameters in the camera frame
bestModel = {0, 1, 0, 0} ;
// Ground plane segmentation
while iterations < $k$ do
    maybeInliers = 3 randomly selected points from $\mathcal{N}_{sampled}$;
    maybeModel = plane parameters fitted to maybeInliers;
    confirmedInliers = empty set;
    for every point in $\mathcal{N}_{sample}$ do
        if point fits maybeModel with an error $< d$ then
            add point to confirmedInliers;
        end
    end
    if the number of elements in confirmedInliers $>$ mostNum then
        betterModel = plane parameters fitted to all the points in confirmedInliers;
        mostNum = the number of elements in confirmedInliers;
    end
    increment iterations;
end
// Validity check for segmentation
$\vec{x}_{plane}$ = the normal vector of bestModel;
similarity = cosine_similarity($\vec{x}_{plane}$, $\vec{x}$);
if similarity $> s_{min}$ then
    bestModel = betterModel;
end
return $\mathcal{N}_{ground}$ based on bestModel;
```

3.3. Autonomous Navigation

Building upon monocular depth estimation and scale recovery, the proposed UAS estimates the body odometry through long video sequences to support efficient flight trajectory and path planning. Utilizing the obtained dense depth maps, we employ RGB-D visual localization to determine the 6-DoF camera pose. Once receiving the target point, whether manually designated or autonomously tracking a specific target, our UAS efficiently generates a smooth and collision-free flight trajectory. It then directs the MAV to follow the generated trajectory, accomplishing the predefined task.

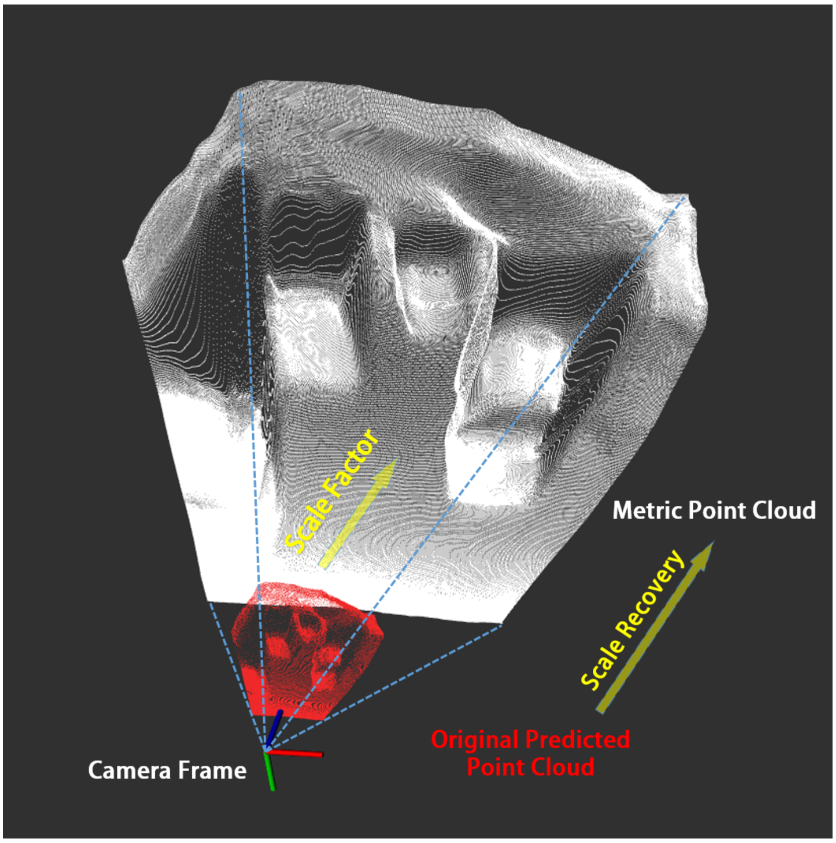


Figure 4. Point clouds predicted before and after scale recovery.

3.3.1. Visual SLAM

A monocular camera-based visual SLAM system necessitates structural initialization through camera movement. In contrast, RGB-D camera-based visual SLAM systems can be initialized solely with the first frame image. In our proposed UAS, the introduction of monocular depth estimation with real-world scale effectively overcomes the limitations of traditional monocular visual SLAM systems.

The initial position of the camera sensor determines where the world coordinate system is established. On an MAV, there is often a mounting angle between the monocular camera and the horizontal direction of the MAV body. This implies that, after system initialization, an unknown mounting angle relationship exists between the x-axis of the world coordinate system and the horizontal direction. As the UAV moves horizontally in the real environment, the mounting angle introduces a z-axis component in the estimated pose of the MAV in the world coordinate system.

Our UAS utilizes ORB-SLAM2 [20] for the MAV's 6-DoF localization. Expanding on this, we achieve sensor mounting angle calibration by horizontally moving the MAV and computing the trigonometric correlation between the MAV's z-axis movement component and the x-axis direction in the world coordinate system. This process establishes the angular relationship between the monocular sensor and the MAV body. The calibration is depicted in Figure 5, illustrating the MAV's pose estimation in the world coordinate system before and after calibration.

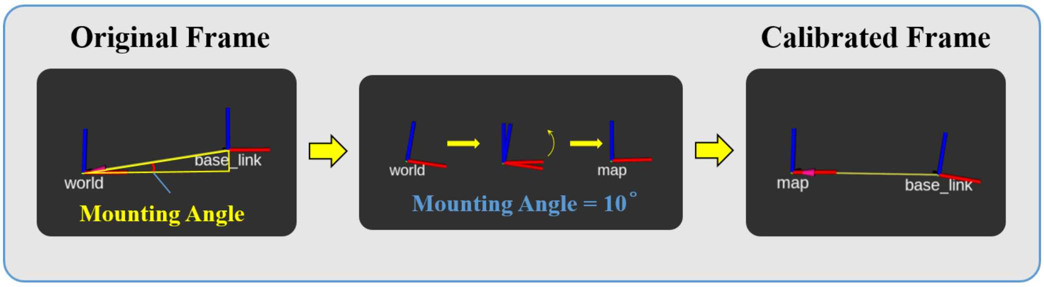


Figure 5. Sensor mounting angle calibration.

3.3.2. Pose Estimation

The accuracy of pose estimation significantly influences the success of autonomous target tracking for MAV. In our developed UAS, we integrate the well-established and thoroughly tested visual tag system AprilTag to predict the target pose. It is noteworthy that the AprilTag module operates independently, offering the system target pose estimates on a real-world scale. This suggests that when the scale of the predicted target pose aligns with the MAV pose estimation in practical experiments, it serves as evidence of the effectiveness of the system's depth estimation and scale recovery.

3.3.3. Path Planning

With accurately determined and unified poses at the metric scale for both the MAV and the target, the UAS can proficiently accomplish autonomous navigation in a real environment. Within our system, the integration of Fast-Planner [11] and Fast-Tracker [12] enables the generation of local trajectories for the MAV, effectively meeting various task requirements, including obstacle avoidance and target tracking. These trajectory planners in the system adhere to a conventional workflow, enabling real-time generation of collision-free flight trajectories that are both spatially and temporally optimized.

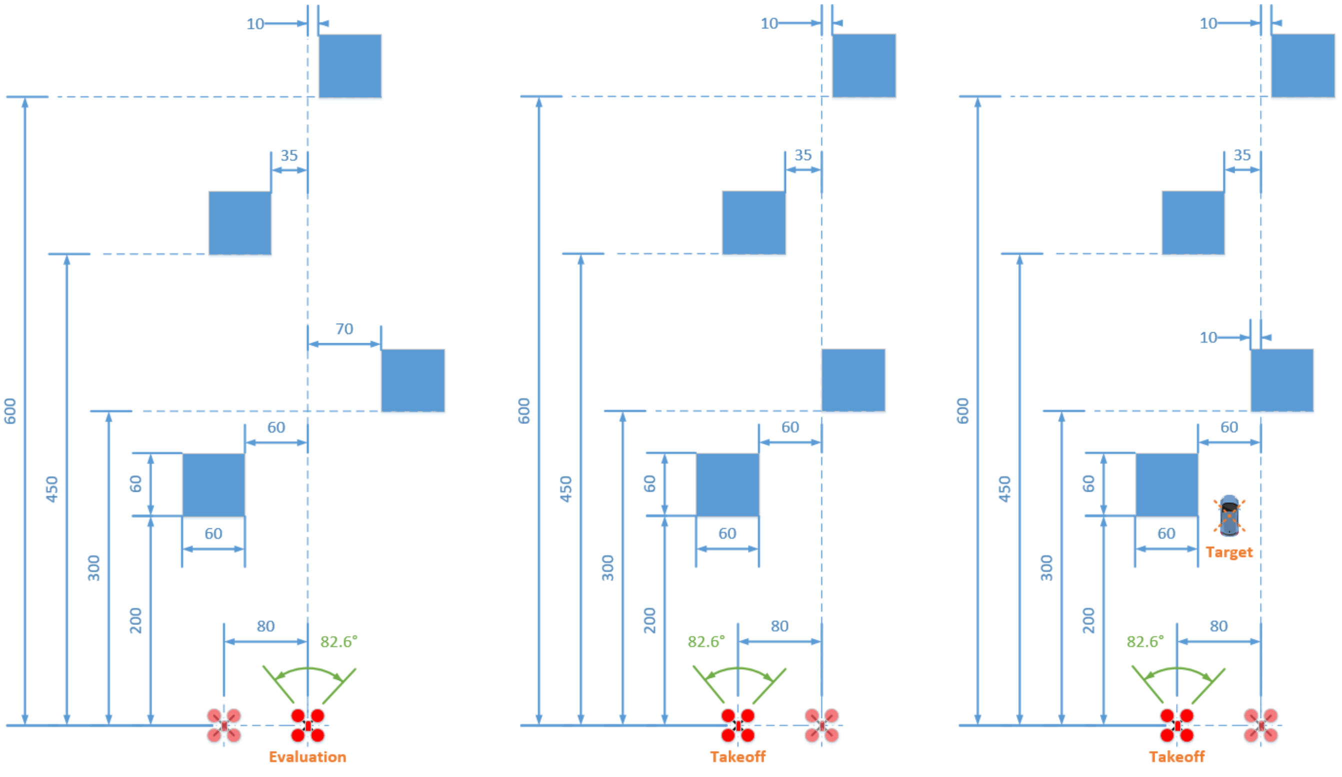
4. Experiments

4.1. Experimental Configurations

In our experiment, we chose the DJI RoboMater TT Tello Talent as our MAV. Our computational processes are facilitated by a ground server featuring an Intel i7-10875H CPU (Intel Corporation, Santa Clara, CA, US). and an NVIDIA GeForce RTX 2070 Super GPU (NVIDIA Corporation, Santa Clara, CA, US). with 8 GB of memory. The entire system is implemented in Ubuntu and ROS (Robot Operating System), specifically Ubuntu 18.04 and ROS Melodic.

We crafted three different experimental scenarios, as illustrated in Figure 6, to construct our MoNA Bench. These experimental setups require well-lit conditions to ensure optimal performance of MDE algorithms. In the frontal region of the MAV, four obstacles are placed at vertical distances of 2 m, 3 m, 4.5 m, and 6 m from the MAV. For the evaluation of MDE algorithm performance, the MAV is suspended at the 'Evaluation' point, with a body height of 1 m above the ground. As for the navigation experiments, including the obstacle avoidance and target tracking experiments, the MAV takes off and departs from a position located 0.8 m to the left of the performance evaluation test location. Before the MAV commences the experiment, it hovers at an approximate height of 0.8 m.

In our experiments, only pre-trained models were employed. Specifically, both MonoDepth and SC-DepthV2 models undergo training on the NYUv2 dataset. In contrast, MiDaS utilizes a total of six datasets for the joint training of its model. Unlike our previous work [1], this experimental section refrains from applying any preprocessing to the predicted depth maps generated by the algorithms.



(**a**) Estimation performance experiment (**b**) Obstacle avoidance experiment (**c**) Target tracking experiment

Figure 6. The fundamental experiments comprising MoNA Bench.

4.2. Depth Estimation Experiments

Utilizing the monocular UAS developed in this paper, we conducted performance tests for three monocular depth estimation algorithms: MonoDepth, MiDaS, and SC-DepthV2. The experiments involved collecting distance estimates for various obstacles from 500 consecutive monocular images generated by each algorithm. The algorithm performance evaluation was achieved by comparing these estimates with the ground-truth distances of the obstacles along the MAV's axial direction. In the designed experimental scenarios, the actual test results for each algorithm are depicted in Figure 7. A summary of the performance for each algorithm is provided below:

MonoDepth. MonoDepth displays a relatively strong performance in predicting distances. The algorithm exhibits relatively inferior distance prediction for the 2 m obstacle compared to SC-DepthV2, while it provides distance predictions for obstacles at longer distances (4.5 m and 6 m) that are closer to the true distances.

MiDaS. The distance prediction performance of MiDaS is highly unstable. The algorithm fails to successfully identify the 2 m obstacle, while the predicted distances for obstacles at slightly longer distances (3 m, 4.5 m, and 6 m) exhibit extremely abrupt inter-frame changes and deviate significantly from the actual values.

SC-DepthV2. SC-DepthV2 demonstrates the best prediction performance. The algorithm provides highly accurate predictions for nearby obstacles (2 m and 3 m obstacles), while its prediction for obstacles at longer distances (4.5 m and 6 m) is relatively less accurate.

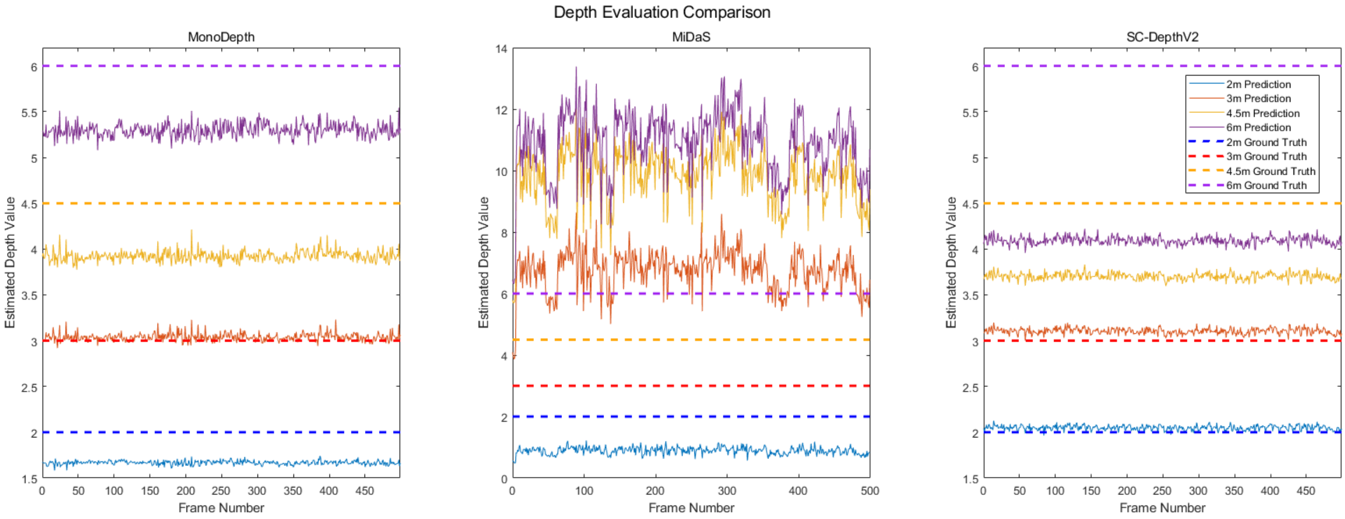


Figure 7. Comparison of depth estimation outcomes across algorithms.

Figure 8 presents the results of experiments involving point cloud generation, ground segmentation, and distance estimation for the algorithms. As observed from the graph, both MonoDepth and SC-DepthV2 accurately capture the spatial positions of diverse obstacles. In contrast, the point cloud generated by MiDaS exhibits a "V"-shaped structure. The irrational spatial structure of MiDaS point clouds poses a challenge, making the original predicted depth map from this algorithm difficult to directly apply in the context of autonomous obstacle avoidance and target tracking for MAVs. In addition, the dramatic frame-to-frame scale variation presents another challenge in achieving stable scale recovery through raw predicted depth maps from MiDaS. This shortcoming results in MiDaS exhibiting different navigation performance compared to our previous work [1].

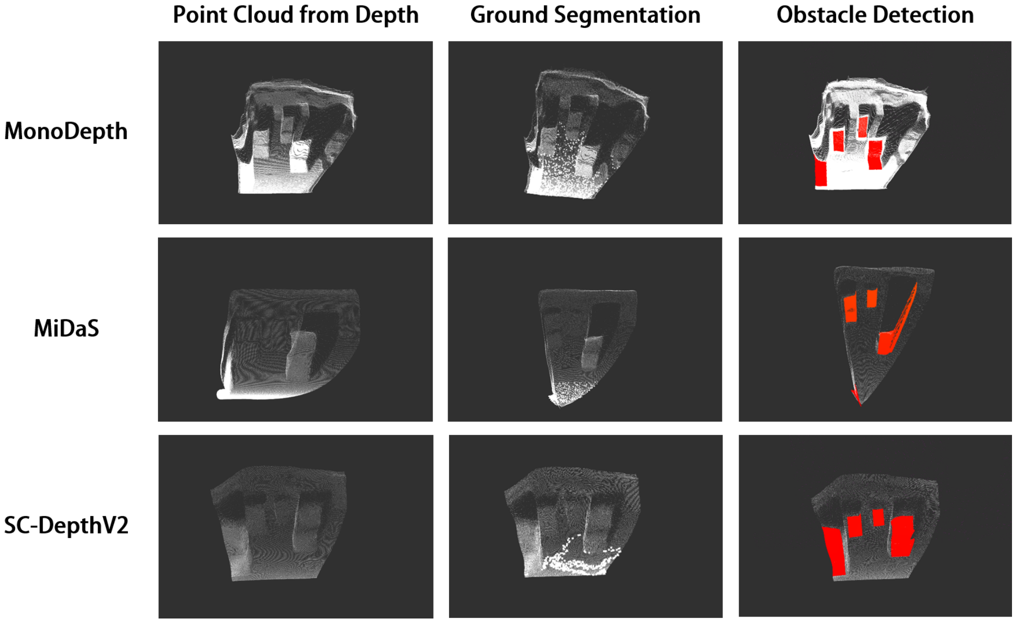


Figure 8. Experiment results: point cloud generation, ground segmentation, and distance estimation.

Due to the essential demand for real-time and efficient data processing by UAS, algorithm evaluation in our proposed MoNA Bench involves considerations of computational cost, efficiency, and accuracy. For depth estimation experiments, key metrics include the GPU occupancy, the coefficient of variation (CV) for the continuous scale factor sequence s, and the average relative error (ARE) between predicted obstacle distances $\hat{y}$ and ground-truth distances y. The CV is calculated as:

$$CV = \left(\frac{\sigma_s}{\mu_s}\right) \times 100 \tag{8}$$

where σ_s represents the standard deviation of the scale factor sequence s, and μ_s is its mean. Additionally, the ARE is defined as:

$$\text{ARE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{9}$$

Based on the aforementioned evaluation metrics, Table 1 provides a performance comparison for the selected algorithms. In summary, MonoDepth and SC-DepthV2 can accurately obtain the spatial positions of various obstacles from monocular images, while the performance of MiDaS is less satisfactory. Furthermore, due to SC-DepthV2's relatively more accurate perception of nearby obstacles and lower computational cost compared to MonoDepth, it can be considered the optimal choice among the three algorithms.

Table 1. Algorithm performance comparison for depth estimation experiments

Algorithms	**GPU Occupancy (%)**	**CV Estimation * (%)**	**2 m ARE (%)**	**3 m ARE (%)**	**4.5 m ARE (%)**	**6 m ARE (%)**
MonoDepth	81–89	1.5	16.6	1.5	12.8	11.7
MiDaS	31–35	9.7	55.7	124.6	117.8	81.9
SC-DepthV2	32–34	1.0	2.5	3.5	17.7	31.8

* CV Estimation: Coefficient of variation for scale factor in depth estimation experiments.

4.3. Autonomous Navigation Experiments

The autonomous navigation experiments are divided into two components: obstacle avoidance and safe target tracking. This division is designed to thoroughly investigate the essential characteristics necessary for efficient TPP in MDE algorithms. At the same time, it serves as a validation of the effectiveness of the proposed UAS.

4.3.1. Obstacle Avoidance

For the evaluation of each algorithm, we collected 700 frames of continuous scale factor sequences and their respective publication frequencies. Additionally, the success rate of obstacle avoidance is assessed in 9 flights, with successful avoidance defined as navigating around 3 obstacles. The results, illustrated in Figure 9, reveal that failures in obstacle avoidance are primarily attributed to three errors:

1. Localization Error. Due to the high computing resource demand, certain MDE algorithms encountered challenges in processing received image data continuously and sequentially in real-time when the ground server reached its performance limits. This discrepancy led to the loss of feature points and, subsequently, localization failure.
2. Detection Error. Discrepancies between the predicted depth map of spatial obstacles and the actual distribution of obstacles in space contributed to collisions between the MAV and obstacles. Detection errors were also observed when the depth map failed to clearly distinguish the boundaries of obstacles, such as the top part of an obstacle and the gap between obstacles.
3. Others. This category includes unexpected wobbling or instability of the MAV during the experiments.

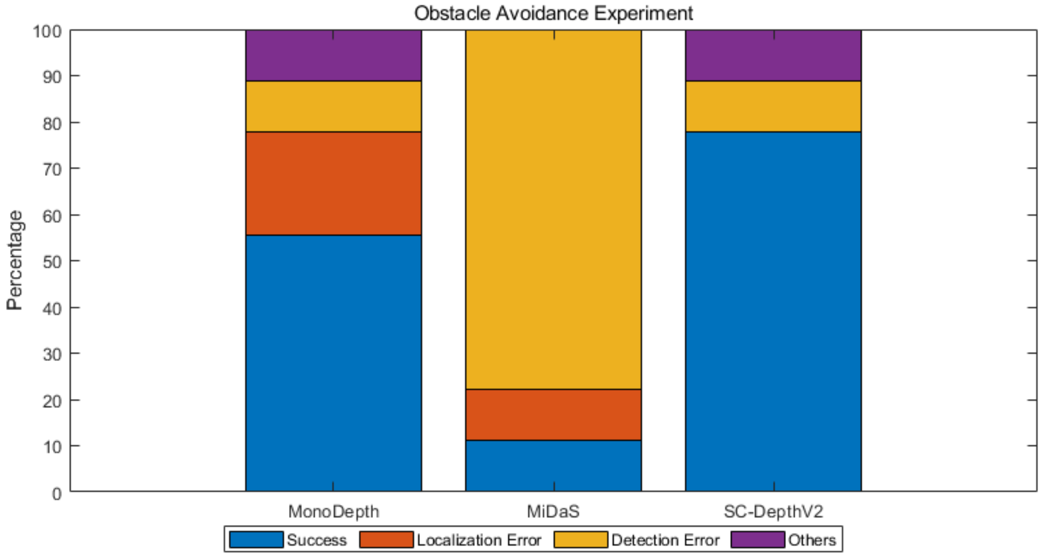


Figure 9. Comparison of obstacle avoidance outcomes across algorithms.

According to our experiments, the autonomous obstacle avoidance performance of each algorithm can be summarized as follows:

MonoDepth. Despite facing localization errors due to its high demands on ground server performance and resulting in a low frame rate for depth map predictions, MonoDepth achieved a commendable success rate, thanks to its ability to generate high-quality predicted depth maps, contributing to its effectiveness in obstacle avoidance. This capability allowed for the generation of a flight trajectory through the gaps between obstacles.

MiDaS. The unstable scale of depth maps in MiDaS, causing substantial frame-to-frame variations and resulting in a V-shaped 3D grid map, led to a considerable number of detection errors. Due to the inaccuracies in the grid map, MiDaS encountered difficulties in guiding the MAV through passages between obstacles. It could only occasionally generate a trajectory around obstacles from the outside to navigate the MAV. Without proper preprocessing, its support for efficient TPP was limited.

SC-DepthV2. This algorithm demonstrated the highest success rate in obstacle avoidance among the three. During its operation, SC-DepthV2 initially generated the first obstacle on the 3D grid map and gradually revealed the obscured obstacles during flight. The algorithm effectively distinguished these obstacles, generating a flight trajectory through their gaps.

The experimental videos are available through our open-source repository, showcasing the generation of obstacle avoidance trajectories in MAV flights. Additionally, throughout the experiments, we observed that MonoDepth and SC-DepthV2 maintained relatively stable real-time calculations of the scale factor during operation. The mean scale factors were close to manually adjusted fixed scale factors from our previous work [1].

4.3.2. Safe Target Tracking

The safe target tracking experiment serves as a validation for the effectiveness of the proposed monocular UAS. The system's effectiveness is established if the MAV can safely track the target's pose, provided in metric scale by AprilTag, while autonomously avoiding obstacles. During the experimental testing, we observed that MiDaS produced spatial structures for point clouds that were irrational, preventing the generation of depth maps from meeting the requirements for efficient TPP. This observation aligns with the conclusions discussed in Section 4.2.

Concerning MonoDepth, its higher computational demands, especially in terms of GPU occupancy, led to data packet losses during operation. These issues hindered its ability to meet the real-time requirements for efficient TPP. In contrast, we successfully conducted monocular MAV safety tracking experiments using SC-DepthV2. Figure 10 depicts the predicted depth map, feature points, and planned trajectory in the safe target tracking experiment.

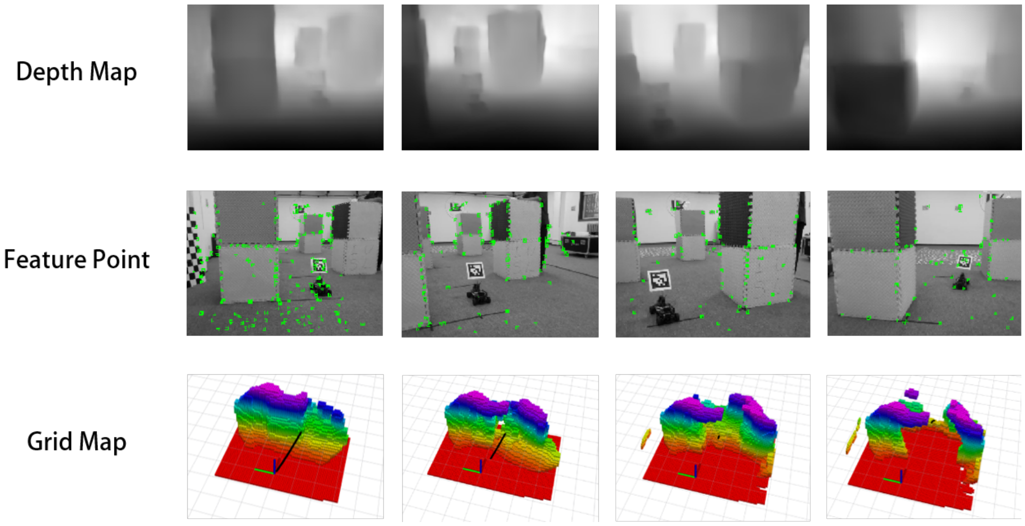


Figure 10. Safe target tracking experiment.

In spite of this, it's worth noting that the system still places high computational demands, particularly on the ground station's CPU, approaching 100% utilization and causing a significant computational load.

Throughout the experiments, we observed that the real-time computation of the scale factor by SC-DepthV2 maintained stability and consistently matched the scale factors obtained in obstacle avoidance scenarios. Building on this observation, we fixed the scale factor to the mean of the continuous scale factor sequence and conducted the safe target tracking experiment with this fixed factor. The results were consistent with those obtained when dynamically updating the scale factor, allowing the MAV to simultaneously track targets safely and autonomously avoid obstacles. Importantly, this approach led to a reduction in the ground station's computational load, with CPU utilization dropping to 80%, resulting in a more stable system operation. This experiment provides strong evidence that fixing the scale factor is a viable strategy in monocular MAV autonomous navigation, particularly when the algorithm maintains scale consistency, thereby effectively achieving efficient TPP.

4.3.3. Navigation Performance

Combining the metrics obtained from obstacle avoidance experiments, including the average depth map prediction publication frequency, the data utilization rate, the CV for the continuous scale factor sequence, and the obstacle avoidance success rate, along with the algorithm's support for safe target tracking, we evaluate the autonomous navigation performance of each algorithm. The data utilization rate is defined as the ratio of the average monocular RGB image reception frequency to the average depth map prediction publication frequency. In our experiments, the average monocular RGB image reception frequency is 30 Hz. The GPU occupancy in navigation experiments is similar to that in depth estimation experiments, as only the MDE modules asks for GPU processing.

Table 2 provides a comparison of navigation performance for the selected algorithms, where the "Safe Tracking" column indicates whether the algorithm supports the MAV's safe tracking of the target, represented as a boolean value.

Table 2. Algorithm performance comparison for autonomous navigation experiments

Algorithms	APF * (Hz)	DUR ** (%)	CV Navigation *** (%)	Success Rate (%)	Safe Tracking
MonoDepth	8.2	26.67	6.3	55.6	False
MiDaS	30.0	100	39.4	11.1	False
SC-DepthV2	30.0	100	5.0	77.8	True

* APF: Average publication frequency. ** DUR: Data utilization rate. *** CV Navigation: Coefficient of variation for scale factor in obstacle avoidance experiments.

4.4. MoNA Bench

Through depth estimation experiments and autonomous navigation experiments, we extensively explored the performance evaluation metrics for MDE algorithms across various task scenarios. Ultimately, we identified that achieving efficient TPP requires MDE algorithms to excel in three key aspects: MDE efficiency, MDE accuracy, and scale consistency. As a result, we summarized these findings in Table 3 and introduced MoNA Bench—a benchmark designed to evaluate the performance of MDE in autonomous navigation for UASs. Based on our assessment, SC-DepthV2 emerged as the top-performing algorithm among the three.

Table 3. Algorithm performance comparison for depth estimation experiments

MoNA Bench							
MDE Efficiency			**MDE Accuracy**	**Scale Consistency**		**Navigation Capability**	
GPU Occupancy	APF	DUR	Distance AREs *	CV Estimation	CV Navigation	Success Rate	Safe Tracking

* Distance AREs: Average relative errors for 2 m, 3 m, 4.5 m, and 6 m.

5. Discussions and Conclusions

In this work, we presented a real-time micro-aerial vehicle-based unmanned aircraft system designed for efficient trajectory and path planning through monocular depth estimation. Building upon our prior research [1], our system integrated an innovative scale recovery module to address the scale ambiguity challenge, leading to accurate depth predictions and enabling the reconstruction of real-world scale. This improvement not only enhanced autonomous obstacle avoidance but also supported safe target tracking under a precise metric scale.

Meanwhile, the established MoNA Bench provided a benchmark for evaluating monocular depth estimation in micro-aerial vehicle autonomous navigation. It underscored the essential attributes that estimation algorithms should possess for efficient trajectory and path planning: estimation efficiency, depth map accuracy, and scale consistency.

Looking forward, the challenges of single ground station performance limitations prompt future exploration into lightweight monocular MAV navigation algorithms for swarm collaboration. Additionally, transitioning from artificial to natural features for complex scenarios is a noteworthy direction for further investigation.

Author Contributions: Conceptualization, T.Y. and W.F.; methodology, T.Y. and Y.P.; software, Y.P. and B.L.; validation, Z.L., H.S. and J.X.; writing—original draft preparation, Y.P.; writing—review and editing, T.Y.; supervision, T.Y.; project administration, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62103332 and the Fundamental Research Funds for the Central Universities, NPU, under Grant 31020200QD045.

Data Availability Statement: The original contributions presented in the study are included in the article. The data analyzed in this study is from the experimental setting presented here: https://github.com/npu-ius-lab/MoNA-Bench.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

TPP	Trajectory and Path Planning
UAS	Unmanned Aircraft System
MAV	Micro-aerial Vehicles
MDE	Monocular Depth Estimation

References

1. Pan, Y.; Wang, J.; Chen, F.; Lin, Z.; Zhang, S.; Yang, T. How Does Monocular Depth Estimation Work for MAV Navigation in the Real World? In Proceedings of the 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022), Xi'an, China, 23–25 September 2022; pp. 3763–3771.
2. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; pp. 2366–2374.
3. Alhashim, I.; Wonka, P. High quality monocular depth estimation via transfer learning. *arXiv* **2018**, arXiv:1812.11941.
4. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
5. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 746–760.
6. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
7. Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1623–1637. [CrossRef]
8. Guizilini, V.; Ambrus, R.; Pillai, S.; Raventos, A.; Gaidon, A. 3d packing for self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2485–2494.
9. Bian, J.W.; Zhan, H.; Wang, N.; Li, Z.; Zhang, L.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised scale-consistent depth learning from video. *Int. J. Comput. Vis.* **2021**, *129*, 2548–2564. [CrossRef]
10. Bian, J.W.; Zhan, H.; Wang, N.; Chin, T.J.; Shen, C.; Reid, I. Auto-Rectify Network for Unsupervised Indoor Depth Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 9802–9813. [CrossRef] [PubMed]
11. Zhou, B.; Gao, F.; Wang, L.; Liu, C.; Shen, S. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3529–3536. [CrossRef]
12. Han, Z.; Zhang, R.; Pan, N.; Xu, C.; Gao, F. Fast-tracker: A robust aerial system for tracking agile target in cluttered environments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 328–334.
13. Kato, H.; Billinghurst, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), San Francisco, CA, USA, 20–21 October 1999; pp. 85–94.
14. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407.
15. Krogius, M.; Haggenmiller, A.; Olson, E. Flexible layouts for fiducial tags. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1898–1903.

16. Zhang, Z.; Xiong, M.; Xiong, H. Monocular depth estimation for UAV obstacle avoidance. In Proceedings of the 2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT), Changchun, China, 6–7 December 2019; pp. 43–47.
17. Liu, Q.; Zhang, Z.; Xiong, M.; Xiong, H. Obstacle Avoidance of Monocular Quadrotors with Depth Estimation. In Proceedings of the International Conference on Autonomous Unmanned Systems, Changsha, China, 24–26 September 2021; pp. 3194–3203.
18. Yonchorhor, J. (The) Development of the Scale-Aware Monocular Depth Estimation Aided Monocular Visual SLAM System for Real-Time Robot Navigation. Master's Thesis, Korea Advanced Institute of Science & Technology (KAIST), Daejeon, Republic of Korea, 2021.
19. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
20. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.