

Article

Pigeon-Inspired Depth-Reasoning-Driven Decision Framework for Autonomous Traversal Flight of Quadrotors in Unmapped 3D Spaces

Yongbin Sun ^{1,2,*}  and Rongmao Su ¹

¹ National Key Laboratory of Aircraft Integrated Flight Control, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China; rmsu@buaa.edu.cn

² State Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing 100083, China

* Correspondence: ybsun@buaa.edu.cn

Abstract

Autonomous traversal flight in unknown 3D environments remains challenging due to mapping bottlenecks and computational latency. Inspired by pigeons navigating cluttered forests through instantaneous visual perception rather than constructing global metric maps, this paper presents a pigeon-inspired depth-reasoning-driven decision framework for agile quadrotor traversal in unmapped spaces without explicit map construction. To ensure feasibility, we leverage a robust state estimation backbone enhanced by deep-learning-based feature matching, providing stable pose feedback under aggressive maneuvers. The core contribution is a pigeon-inspired depth-reasoning framework that translates raw sensory depth data into a hybrid optimization framework, integrating both hard safety constraints and soft geometric smoothness constraints, directly emulating the three avian mechanisms: gap selection via instantaneous depth gradients, path selection that minimizes posture changes, and a safety field driven by the looming effect. By bypassing time-consuming mapping and spatial discretization processes, the framework significantly reduces perception-to-control latency. Finally, validated via simulations and real-world experiments on a resource-constrained quadrotor platform, our map-less approach achieves superior decision frequencies and comparable safety margins to those of state-of-the-art map-based planners. This framework offers a practical, high-frequency solution for autonomous flight where computational resources and environmental knowledge are strictly limited.

Keywords: pigeon-inspired; Visual-Inertial Odometry; autonomous traversal flight; quadrotors; unmapped space; depth reasoning; constraint-fusion planning



Academic Editor: Antonio Concilio

Received: 6 March 2026

Revised: 13 April 2026

Accepted: 14 April 2026

Published: 19 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the ongoing advancement of aerial robotics, unmanned aerial vehicles (UAVs), especially quadrotors, have been increasingly involved in practical applications due to their remarkable maneuverability and flexibility. The potential of autonomous unmanned robots extends beyond quadcopters, permeating numerous fields. Significant theoretical progress has emerged, such as the development of VLM and VLA [1] in visual autonomous driving. Moreover, trajectory generation and quadcopter control [2] have become trending research topics. Nevertheless, unlike theoretical studies, the safety of unmanned systems in engineering applications has always been crucial for practical use. This highlights the importance of the generalization and robustness of system algorithms.

In practical applications, sensors are indispensable for continuously monitoring the external environment. Currently, a variety of sensors, including visual cameras, event cameras [3], LiDAR, and radar are widely utilized. Especially for exploration tasks in unmapped spaces, such as cave exploration, rubble search, and jungle traversal, flight planning under map-free conditions heavily relies on the external information captured by sensors, which directly impacts operational effectiveness. Each sensor type has distinct characteristics. While visual cameras offer more comprehensive information for subsequent tasks compared to other sensors, their positioning efficiency is inferior to that of radar sensors. However, the human navigation system heavily depends on the processing of visual information [4], such as visual cues and scene details, rather than pure positional data, to form a holistic spatial representation. This biological phenomenon finds a compelling parallel in nature: pigeons navigating through densely cluttered forests rely predominantly on instantaneous visual perception and optical flow analysis, constructing only an implicit spatial understanding rather than maintaining precise metric maps of their surroundings.

1.1. Related Works

1.1.1. Visual–Inertial Odometry

In comparison to the Visual Odometry (VO) system that relies solely on visual positioning, the Visual–Inertial Odometry (VIO) system, which integrates an Inertial Measurement Unit (IMU), can achieve a pronounced complementary effect between the IMU and camera [5,6]. Equipped with the ability to capture high-frequency data, IMUs are particularly advantageous in short-term and fast-changing motions. Moreover, their low dependence on external environmental information endows them with high robustness across diverse environments. In contrast, visual positioning technology excels in long-term and slowly changing movements, enabling it to provide precise information when the environment remains relatively stable. The VINS-Mono algorithm [7,8] successfully integrated inertial navigation with visual positioning, effectively leveraging the strengths of both systems to enhance overall performance. Similarly, ORB-SLAM3 [9] began incorporating IMUs for fusion, thereby improving its own positioning capabilities. At the present stage, visual positioning technology has attained a certain level of maturity in terms of theory, with relevant theories being well-established. However, its application capability in real-world environments remains a significant challenge. The performance of visual technology is highly subject to environmental conditions, which is a major factor restricting its robustness.

Mainstream visual positioning techniques, including DSO [10], MSCKF [11], ROVIO [12], ICE-BA [13], and other algorithms, predominantly employ direct or optical flow methods to minimize the photometric error in the front-end odometer part to solve the least-squares optimization problem. While direct methods leverage per-pixel image information for camera pose estimation and reduce reliance on feature point extraction and matching, they partially circumvent issues from feature scarcity or matching inaccuracies. Nonetheless, the strong assumptions of photometric invariance and spatial continuity underpinning these methods may compromise convergence in real-world scenarios, particularly during visual positioning in crossing or highly maneuverable conditions.

ORB-SLAM [9,14,15] algorithms are based on indirect methods for front-end processing. Leveraging the descriptive and matching capabilities of ORB feature points, they effectively handle diverse complex real-world scenarios while meeting real-time requirements. Compared to other approaches, indirect methods rely on feature point matching [16–20] to acquire spatial information, offering a high degree of invariance. This enables them to provide superior feature information in more complex environments, making them more robust in practical applications. Recent advancements have increasingly leveraged deep learning to augment the robustness and information density of perception modules.

DeepV2D [21] proposes a differentiable Structure-from-Motion module to estimate dense depth and camera motion from video streams. Meanwhile, DROID-SLAM [22] introduces a differentiable bundle adjustment layer to enable the recurrent iterative optimization of pose and depth estimates.

However, indirect methods also have drawbacks. The performance of indirect methods is closely tied to the quality of the feature points. In most cases, the visual positioning component provides status information for subsequent tasks. Since feature points represent sparse information, it is impossible to construct a dense map. This also complicates subsequent trajectory estimation tasks.

1.1.2. Motion Planning

The soft-constrained trajectory planning method [2,23,24] incorporates flexible constraints into the trajectory planning process. This approach permits certain constraints to be violated within a defined scope to enhance other planning objectives, such as trajectory smoothness, energy efficiency, and time optimization. Nonetheless, it has notable drawbacks. It cannot strictly confirm that all constraints are satisfied and sensitive to parameter selection. This sensitivity means that environment-specific adjustments often require significant time and effort. Typically, this method assumes the environment is known and relatively static [25]. In complex or dynamic environments, soft constraints may not adapt quickly enough, leading to potential trajectory planning failures.

However, due to the low adjustability of hard constraints during the planning process, the generated paths cannot achieve better results than soft constraints, so the performance has always been mediocre. But this also means that in the actual execution process, the behavior of the trajectory planning system dominated by hard constraints is predictable, and there will be no violation of constraints. Moreover, due to the strictness of hard constraints, the planned trajectory is highly repeatable in multiple executions, which is very important for industrial production and automation tasks that require consistency [26]. Under the condition of satisfying hard constraints, optimization algorithms such as linear programming and nonlinear programming can be used to further optimize the performance indicators of the trajectory and improve the overall performance of the system. Currently, using reinforcement learning and other methods for trajectory planning [27,28] is also a popular direction, but the drawbacks are also very obvious. Under limited training conditions, it cannot adapt to complex environments and requires high investment in the early stages of the algorithm.

Prior map information or an established local cost map is necessary for trajectory planning [29]. For instance, CHOMP [30] leverages covariant gradient descent for continuous trajectory optimization, enabling the rapid generation of smooth, dynamically feasible trajectories. Similarly, optimization-based local planners such as FASTER [31] have significantly enhanced UAV safety and trajectory quality during high-speed flight by integrating safe backup trajectories with sampling strategies in free-known spaces. A majority of existing trajectory generation methods are fundamentally based on a soft constraint approach that is dominated by known environmental information from a third-party perspective. These methods focus on optimizing specific performance metrics using gradient-based optimization techniques. However, such approaches often result in trajectories with high redundancy and are unable to guarantee the safety of the planned trajectory in complex and unmapped spaces.

1.1.3. Bio-Inspired Flight Behaviors

Beyond the engineering-oriented approaches discussed above, a rich body of ethological research has unveiled sophisticated flight strategies in birds that can directly inspire

the design of autonomous UAVs [32]. Raptors, for instance, exhibit remarkable target tracking capabilities: they avoid the confusion effect in dense prey aggregations by locking onto fixed points, a strategy that ensures successful interception [33]. Similarly, hawks steer attacks using a guidance system specifically tuned for the close pursuit of erratically maneuvering targets [34].

Collective behavior in avian flocks also provides rich inspiration for multi-UAV coordination. Studies on mixed-species flocks of songbirds reveal that collective decision-making emerges from simple social interaction rules, enabling the group to respond adaptively to environmental changes [35]. Moreover, the presence of predators significantly influences flocking dynamics, leading to phase transitions in group behavior [36]. These findings underpin recent advances in bird-inspired self-propelled mechanisms for UAV swarm control [37], where phase transition principles are employed to achieve robust formation flight.

In addition to flocking and predation, individual flight maneuvers such as perching have been extensively studied. The optimization of avian perching maneuvers reveals fundamental principles of agility and precision, balancing kinetic energy management with trajectory planning [38]. Recent work on morphing-wing drones demonstrates how these biological strategies can be successfully replicated in engineered systems, achieving agile perching through wing articulation mechanisms inspired by birds [39].

1.2. Key Contributions of This Article

In the above context, autonomous exploration of unknown spaces by unmanned robots faces several challenges. First, current visual-information-based positioning methods are not robust and are greatly affected by the environment. This causes noticeable positioning errors and raises safety concerns in practical use. Second, with limited time and onboard computing resources, spatial mapping is delayed by another thread. This compromises the real-time nature of information. Existing work cannot ensure guidance using safe and reliable trajectories. Finally, current trajectory planning and optimization rely on known spatial information. The computational feasible domain in the environment is large, leading to high trajectory redundancy and heavy dependence on prior information. These challenges stand in stark contrast to the efficiency of biological flight—pigeons traverse unknown, cluttered environments at high speeds without explicit mapping, relying instead on a tight coupling between visual perception and motor control.

In this paper, inspired by this biological paradigm, the above problems are addressed through the design of a pigeon-inspired depth-reasoning-driven decision framework. Within the visual front-end, a deep learning framework is incorporated to enhance feature point matching. This is based on the indirect method, which is less susceptible to environmental influences, thereby bolstering the robustness of the visual positioning system.

In this work, we define unmapped navigation as a system that does not construct or maintain a globally consistent metric map (e.g., voxel grids, TSDF, or octomaps) for path planning. Instead, the system relies on a local instantaneous depth buffer and short-term visual-inertial states. This not only reduces computational load but also maintains positioning robustness, effectively addressing the non-real-time issue of dense mapping in delayed threads. For subsequent trajectory planning, a pigeon-inspired deep-vision-based autonomous navigation method is proposed. Drawing upon three mechanisms observed in pigeon flock flight—gap selection via instantaneous depth gradients [40], path selection that minimizes posture changes [41], and a safety field driven by the looming effect [42]—this method integrates depth perception with parametric trajectory generation, enabling quadrotors to navigate unmapped space and generate waypoints.

In our approach, hard constraints are primarily utilized while also integrating soft constraints. Based on the sparse path points obtained through the aforementioned method,

we generate a controllable trajectory for the quadcopter's safe flight. This enables autonomous and robust crossing flight of the quadcopter in a completely unknown complex environment. To validate its effectiveness, a series of experiments were designed and conducted. The contributions of this paper are as follows.

(1) A deep learning-based feature-matching state estimation method is integrated into the framework for autonomous traversal flight of quadrotors in unmapped 3D space, providing the necessary stability for the pigeon-inspired depth-reasoning-driven decision framework during maneuvering.

(2) A novel pigeon-inspired depth-reasoning-driven decision framework that directly generates trajectory constraints from raw sensory depth data. By bypassing explicit mapping and spatial discretization, which mirrors the avian ability to interpret visual scenes without constructing metric representations, the proposed method significantly reduces the latency from perception to control.

(3) A trajectory generation strategy integrating hybrid constraints, where raw depth information is transformed into hard safety boundaries, while soft smoothness terms are optimized to ensure the dynamic feasibility and continuity of the flight path in unmapped 3D spaces, which replicate the instantaneous trade-off between collision avoidance and flight efficiency observed in nature.

(4) Extensive comparative evaluations in high-fidelity simulations and feasibility validation on a resource-constrained physical platform.

The rest of the paper is organized as follows. Depth-guided VIO localization is introduced in Section 2. The pigeon-inspired depth-reasoning-driven decision framework is given in Section 3. The experimental results and analysis is discussed in Section 4. Section 5 summarizes this whole article.

2. Depth-Guided VIO Localization

The objective of VIO is to estimate the state of the mobile carrier by fusing data from the camera and IMU. Traditional VIO relies on geometric features, which perform poorly in weak texture environments, dynamic scenes, or intense motions, leading to reduced tracking reliability.

2.1. Problem Definition

As shown in Figure 1a, the infrared camera captures the infrared light reflected by an object to generate an infrared grayscale image. By comparing the images from the left and right infrared cameras, the disparity can be computed, which in turn allows for the determination of the distance between the object and the camera. Concurrently, the depth stream can be aligned with the color video stream to produce an image that contains both depth and color information.

We formalize the depth-guided VIO localization problem, which centers on fusing explicit depth measurements into the VIO framework to enhance robustness and accuracy. The objective is to devise a state estimation system that optimizes the integration of data from three complementary modalities: visual features $\mathcal{Z}$, inertial measurements $\mathcal{I}$, and depth information $\mathcal{D}$. The system state defined at time t is

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{p}_t & \mathbf{v}_t & \mathbf{q}_t & \mathbf{b}_t^a & \mathbf{b}_t^g & \mathbf{f}_t^{1:N} \end{bmatrix}^T \quad (1)$$

where $\mathbf{p}_t \in \mathbb{R}^3$ is the position, $\mathbf{v}_t \in \mathbb{R}^3$ is the velocity, $\mathbf{q}_t \in SO(3)$ represents the attitude information, $\mathbf{b}_t^a$ and $\mathbf{b}_t^g$ are IMU accelerometer and gyroscope biases, and $\mathbf{f}_t^{1:N} \in \mathbb{R}^3$ are the 3D coordinates of N tracked visual features. Given the following timing measurements.

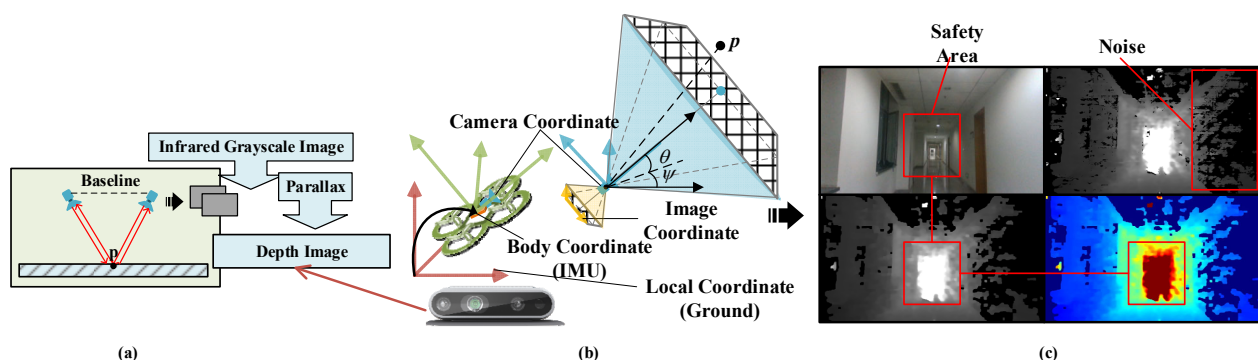


Figure 1. The state definition of the carrier and the sensor, the arrows in the figure represent the corresponding right-handed coordinate system. (a) Measurement principle diagram of the depth camera. (b) Some state information definitions in algorithms. (c) Image information received by the camera, including color images and depth maps.

$$\mathbf{Z}_t = \left\{ \mathbf{z}_t^i \right\}_{i=1}^N = \left\{ u_t^i, v_t^i \right\}_{i=1}^N \quad (2)$$

$$\mathcal{I}_t = \left\{ \mathbf{a}_\tau, \boldsymbol{\omega}_\tau \right\}_{\tau=t_0}^t \quad (3)$$

$$\mathcal{D}_t = \left\{ d_t^j \right\}_{j=1}^M \quad (4)$$

where d is the depth value aligned with the visual feature, u and v are the pixel coordinates, $\mathbf{a}$ and $\boldsymbol{\omega}$ are the acceleration and angular velocity values measured by the IMU. The state estimation problem can be reduced to the maximum a posteriori (MAP) estimation problem in a sliding window of states $\chi = \{\mathbf{x}_{t-k}, \dots, \mathbf{x}_t\}$.

$$\chi^* = \underset{\chi}{\operatorname{argmax}} \left(\prod_{\tau=t-k}^t p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, \mathcal{I}_\tau) \prod_{i,j} p(\mathbf{z}_\tau^i | \mathbf{x}_\tau, \mathbf{f}^i) p(d_\tau^j | \mathbf{x}_\tau, \mathbf{f}^j) \right) \quad (5)$$

where $p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, \mathcal{I}_\tau)$ represents the IMU motion prior, $p(\mathbf{z}_\tau^i | \mathbf{x}_\tau, \mathbf{f}^i)$ is the visual reprojection likelihood, and $p(d_\tau^j | \mathbf{x}_\tau, \mathbf{f}^j)$ is the depth measurement likelihood.

2.2. Deep Learning Network-Based Vision

In our system, the visual-inertial state estimation is built upon a robust frontend utilizing SuperGlue [43,44] as shown in Figure 2. The decision to employ a learning-based matcher rather than traditional hand-crafted descriptors (e.g., ORB) is informed by its well-documented resilience to motion blur, extreme lighting variations, and low-texture environments—conditions that are frequently encountered during high-speed quadrotor flight. Extensive prior studies [45,46] have already demonstrated that deep learning network-based VIO significantly outperforms traditional methods in terms of tracking success rates and long-term stability. Consequently, we treat the VIO module as a robust perception backbone, focusing our investigation on how this stable state estimation enables high-frequency, map-less trajectory reasoning on resource-constrained hardware like the Jetson Xavier NX.



Figure 2. Feature extraction and matching results. The lines in the figure represent feature point pairs, and the colors represent confidence levels.

2.3. Sensor Fusion on $SE(3)$ Manifold

2.3.1. IMU Pre-Integration

The high-frequency characteristics of IMU data are not synchronous with those of low-frequency sensors like vision. Moreover, the measurements of IMU are susceptible to bias and noise. IMU Pre-integration [47] can pre-integrate IMU measurements between adjacent keyframes. It can generate observations decoupled from the initial state, thus greatly improving the efficiency of optimization. This section elaborates on the mathematical form of IMU pre-integration based on the theories of Lie groups and Lie algebras. Rotation group is $SO(3) = \{\mathbf{R} \in \mathbb{R}^3 | \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$. Lie Algebra is $\mathfrak{so}(3) = \{\boldsymbol{\phi} \in \mathbb{R}^3 | \hat{\boldsymbol{\phi}} \in \mathbb{R}^{3 \times 3}\}$, where $\hat{\boldsymbol{\phi}}$ is an antisymmetric matrix. Lie groups and Lie algebras satisfy the following exponential and logarithmic mappings.

$$\text{Exp}(\boldsymbol{\phi}) = \mathbf{R} \in SO(3) \quad \text{Log}(\mathbf{R}) = \boldsymbol{\phi} \in \mathfrak{so}(3) \quad (6)$$

The IMU measurement at time t is

$$\begin{cases} \hat{\boldsymbol{\omega}}_t^b = \boldsymbol{\omega}_t^b + \mathbf{b}_t^g + \mathbf{n}^g \\ \hat{\mathbf{a}}_t^b = \mathbf{a}_t^b + \mathbf{R}_{w,t}^b \mathbf{g}^w + \mathbf{b}_t^a + \mathbf{n}^a \end{cases} \quad (7)$$

where white noise $\mathbf{n}_a \sim \mathcal{N}(0, \sigma_a^2)$, $\mathbf{n}_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$, $\mathbf{g}$ is the acceleration due to gravity. In the period $[t_i, t_j]$, the incremental truth value discrete model of the state is

$$\begin{cases} \mathbf{R}_j = \mathbf{R}_i \cdot \prod_{k=i}^{j-1} \text{Exp}\left(\left(\hat{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \mathbf{n}_k^g\right)\Delta t\right) \\ \mathbf{v}_j = \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathbf{R}_k \left(\hat{\mathbf{a}}_k - \mathbf{b}_k^a - \mathbf{n}_k^a\right)\Delta t \\ \mathbf{p}_j = \mathbf{p}_i + \sum_{k=i}^{j-1} \left[\mathbf{v}_k\Delta t + \frac{1}{2}\mathbf{R}_k \left(\hat{\mathbf{a}}_k - \mathbf{b}_k^a - \mathbf{n}_k^a\right)\Delta t^2\right] \end{cases} \quad (8)$$

where $\Delta t_{ij} = \sum_{k=i}^{j-1} \Delta t$. Introducing discrete IMU pre-integrated quantities decoupled from the initial state

$$\begin{cases} \Delta \mathbf{R}_{ij} \triangleq \mathbf{R}_i^T \mathbf{R}_j = \prod_{k=i}^{j-1} \text{Exp}\left(\left(\hat{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \mathbf{n}_k^g\right)\Delta t\right) \\ \Delta \mathbf{v}_{ij} \triangleq \mathbf{R}_i^T \left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \sum_{k=i}^{j-1} \Delta \mathbf{R}_{ik} \left(\hat{\mathbf{a}}_k - \mathbf{b}_k^a - \mathbf{n}_k^a\right)\Delta t \\ \Delta \mathbf{p}_{ij} \triangleq \mathbf{R}_i^T \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t^2\right) = \sum_{k=i}^{j-1} \left[\Delta \mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta \mathbf{R}_{ik} \left(\hat{\mathbf{a}}_k - \mathbf{b}_k^a - \mathbf{n}_k^a\right)\Delta t^2\right] \end{cases} \quad (9)$$

The state deviation caused by the bias of the IMU is

$$\begin{cases} \Delta \mathbf{R}_{ij} \left(\mathbf{b}_i^g + \delta \mathbf{b}_i^g\right) \approx \Delta \mathbf{R}_{ij} \left(\mathbf{b}_i^g\right) \text{Exp}\left(\frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{b}_i^g} \delta \mathbf{b}_i^g\right) \\ \Delta \mathbf{v}_{ij} \left(\mathbf{b}_i^g + \delta \mathbf{b}_i^g, \mathbf{b}_i^a + \delta \mathbf{b}_i^a\right) \approx \Delta \mathbf{v}_{ij} \left(\mathbf{b}_i^g, \mathbf{b}_i^a\right) + \frac{\partial \Delta \mathbf{v}_{ij}}{\partial \mathbf{b}_i^g} \delta \mathbf{b}_i^g + \frac{\partial \Delta \mathbf{v}_{ij}}{\partial \mathbf{b}_i^a} \delta \mathbf{b}_i^a \\ \Delta \mathbf{p}_{ij} \left(\mathbf{b}_i^g + \delta \mathbf{b}_i^g, \mathbf{b}_i^a + \delta \mathbf{b}_i^a\right) \approx \Delta \mathbf{p}_{ij} \left(\mathbf{b}_i^g, \mathbf{b}_i^a\right) + \frac{\partial \Delta \mathbf{p}_{ij}}{\partial \mathbf{b}_i^g} \delta \mathbf{b}_i^g + \frac{\partial \Delta \mathbf{p}_{ij}}{\partial \mathbf{b}_i^a} \delta \mathbf{b}_i^a \end{cases} \quad (10)$$

The recursive Jacobian matrix of the system state variable with bias is

$$\begin{cases} \frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{b}_i^g} = \Delta \mathbf{R}_{i,j-1} \frac{\partial \mathbf{R}_{i,j-1}}{\partial \mathbf{b}_i^g} - \mathbf{J}_r^{j-1} \Delta t \\ \frac{\partial \mathbf{v}_{ij}}{\partial \mathbf{b}_i^g} = \frac{\partial \mathbf{v}_{i,j-1}}{\partial \mathbf{b}_i^g} - \left[\Delta \mathbf{R}_{i,j-1} \left(\mathbf{a}_{j-1} - \mathbf{b}_i^a\right) \frac{\partial \Delta \mathbf{R}_{i,j-1}}{\partial \mathbf{b}_i^g} \Delta t\right] \\ \frac{\partial \mathbf{v}_{ij}}{\partial \mathbf{b}_i^a} = \frac{\partial \mathbf{v}_{i,j-1}}{\partial \mathbf{b}_i^a} - \Delta \mathbf{R}_{i,j-1} \Delta t \\ \frac{\partial \mathbf{p}_{ij}}{\partial \mathbf{b}_i^g} = \frac{\partial \mathbf{p}_{i,j-1}}{\partial \mathbf{b}_i^g} + \left[\frac{\partial \mathbf{v}_{i,j-1}}{\partial \mathbf{b}_i^g} \Delta t - \frac{1}{2}\Delta \mathbf{R}_{i,j-1} \left(\mathbf{a}_{j-1} - \mathbf{b}_i^a\right) \frac{\partial \Delta \mathbf{R}_{i,j-1}}{\partial \mathbf{b}_i^g} \Delta t^2\right] \\ \frac{\partial \mathbf{p}_{ij}}{\partial \mathbf{b}_i^a} = \frac{\partial \mathbf{p}_{i,j-1}}{\partial \mathbf{b}_i^a} + \left[\frac{\partial \mathbf{v}_{i,j-1}}{\partial \mathbf{b}_i^a} \Delta t - \frac{1}{2}\Delta \mathbf{R}_{i,j-1} \Delta t^2\right] \end{cases} \quad (11)$$

where $\mathbf{J}_r(\cdot)$ is the Jacobian matrix of the right-multiplied small perturbation $\delta \boldsymbol{\phi}$, using the BCH approximation.

$$\mathbf{J}_r^k = \mathbf{J}_r \left(\left(\boldsymbol{\omega}_k - \mathbf{b}_i^g \right) \Delta t \right) \quad (12)$$

2.3.2. Tightly Coupled Pose Estimation

As shown in Figure 3, a visual-inertial bundle adjustment (BA) framework is employed. To achieve the maximum a posteriori estimation, the sum of the prior and the Mahalanobis norm of all measurement residuals is minimized [48].

$$\min_{\chi} \left\{ \|\mathbf{r}_p - \mathbf{H}_p \chi\|^2 + \sum_{k \in B} \|\mathbf{r}_B \left(\mathbf{z}_{b_{k+1}}^{b_k}, \chi \right)\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2 + \sum_{(i,j) \in C} \rho \left(\|\mathbf{r}_C \left(\mathbf{z}_{c_i}^{f_j}, \chi \right)\|_{\mathbf{P}_{c_i}^{f_j}}^2 \right) + \sum_{l \in D} \rho \left(\|\mathbf{r}_D \left(\mathbf{z}_{d_l}^{f_l}, \chi \right)\|_{\mathbf{P}_{d_l}^{f_l}}^2 \right) \right\} \quad (13)$$

where $\rho(\cdot)$ can be selected as either the Huber or Cauchy kernel function to target different levels of outliers. $\mathbf{r}_p$ represents the residual of the prior information. $\mathbf{r}_B$ represents the

residual error estimated by the IMU. $\mathbf{r}_C$ is the camera reprojection residual. $\mathbf{r}_D$ is the residual of depth estimation. The specific definitions of each residual are as follows.

$$\mathbf{r}_B(\tilde{\mathbf{z}}_{b_{k+1}}^k, \chi) = \begin{bmatrix} \mathbf{r}_{\Delta \mathbf{R}_{ij}} \\ \mathbf{r}_{\Delta \mathbf{v}_{ij}} \\ \mathbf{r}_{\Delta \mathbf{p}_{ij}} \\ \delta \mathbf{b}^a \\ \delta \mathbf{b}^g \end{bmatrix} \triangleq \begin{bmatrix} \text{Log}(\Delta \hat{\mathbf{R}}_{ij}^T \Delta \tilde{\mathbf{R}}_{ij}) \\ \Delta \tilde{\mathbf{v}}_{ij} - \Delta \hat{\mathbf{v}}_{ij} \\ \Delta \tilde{\mathbf{p}}_{ij} - \Delta \hat{\mathbf{p}}_{ij} \\ \mathbf{b}_{k+1}^a - \mathbf{b}_k^a \\ \mathbf{b}_{k+1}^g - \mathbf{b}_k^g \end{bmatrix} \quad (14)$$

where $\tilde{\mathbf{z}}_{b_{k+1}}^k, \Delta \tilde{\mathbf{R}}_{ij}, \Delta \tilde{\mathbf{v}}_{ij}, \Delta \tilde{\mathbf{p}}_{ij}$ indicate the measurement value of IMU, $\Delta \hat{\mathbf{R}}_{ij}^T, \Delta \hat{\mathbf{v}}_{ij}, \Delta \hat{\mathbf{p}}_{ij}$ represent the observation value of the camera, which serves as the reference value of the IMU data.

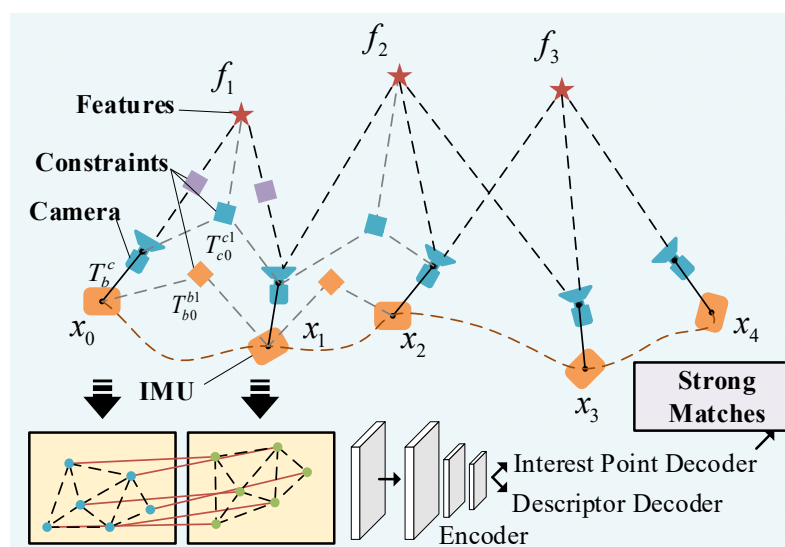


Figure 3. Information fusion based on tight coupling. The feature point information is fused into the estimation of IMU and depth information using a strong matching method.

$$\mathbf{r}_C(\tilde{\mathbf{z}}_{c_i}^{f_j}, \chi) = \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_{c_i}^{f_j} \\ \hat{v}_{c_i}^{f_j} \end{bmatrix} \right) - \mathbf{T}_b^c \mathbf{T}_w^{b_{ij}} \mathbf{T}_{b_i}^w \mathbf{T}_c^b \frac{1}{\lambda_{f_j}} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_{c_{i'}}^{f_j} \\ \hat{v}_{c_{i'}}^{f_j} \end{bmatrix} \right) \quad (15)$$

where π_c^{-1} is the back-projection matrix of camera. λ_{f_j} is the estimated value of the depth of the feature point. $\begin{bmatrix} \hat{u}_{c_i}^{f_j} \\ \hat{v}_{c_i}^{f_j} \end{bmatrix}$ and $\begin{bmatrix} \hat{u}_{c_{i'}}^{f_j} \\ \hat{v}_{c_{i'}}^{f_j} \end{bmatrix}$ are the observed pixel coordinates of the consensus point for the i time and other frames, respectively.

$$\mathbf{r}_D(\tilde{\mathbf{z}}_{d_i}^{f_l}, \chi) = \lambda_{f_l} - d_l \quad (16)$$

where d_l is the depth of the feature points measured by the depth camera.

3. Pigeon-Inspired Depth-Reasoning-Driven Decision Framework

When executing flight maneuvers, avian flocks frequently encounter intricate environmental configurations, as depicted in Figure 4a. Under such circumstances, the necessity for rapid, robust, and safe flight is paramount. Consequently, birds rely solely on visual information to make local behavioral decisions. These mechanisms allow pigeons to perceive the looming effect of obstacles and identify traversable gaps through instantaneous depth-gradient reasoning. Similarly, in the context of quadrotor crossing flight, the visual guidance framework we have developed can rapidly delineate the decision-making area

and generate a flight decision trajectory based on prevailing state information, as illustrated in Figure 4b. This capability, directly inspired by the pigeon's real-time visual processing for gap detection, enables the execution of autonomous crossing flight missions within unmapped and complex spatial environments.

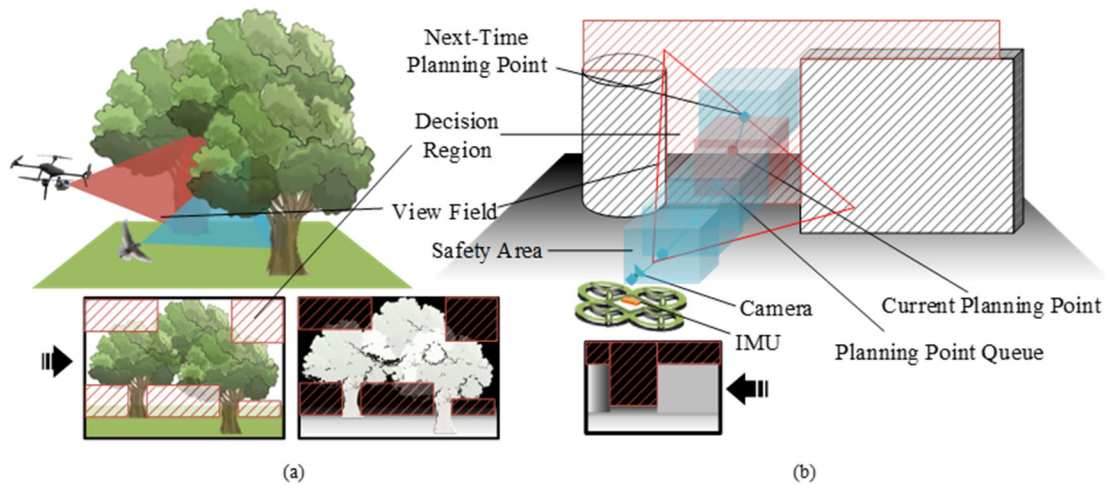


Figure 4. Pigeon-inspired vision-guided autonomous trajectory traversal of a quadrotor. (a) Similarity between quadcopter and pigeon vision in practice. (b) Reference state of the quadrotor during obstacle avoidance.

3.1. Basic Vision Model

Firstly, some information of the visual model and used quadrotor model are defined. During the entire flight process, the following four coordinate systems are used: the local coordinate system p_w fixed to the ground, the body coordinate system p_b , the camera coordinate system p_c , and the imaging coordinate system p_{uv} , as shown in Figure 1b. The correlation between the coordinate systems is shown in the following formula. T and K_c are the external parameter matrix and internal parameter matrix of the camera, respectively.

$$p_c = T_w^c \cdot p_w, \quad T = \begin{bmatrix} R_w^c & t_w^c \\ 0^T & 1 \end{bmatrix} \quad (17)$$

$$p_{uv} = K_c \cdot p_c, \quad K_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

The camera parameter vector are as Equation (19), where the horizontal and vertical focal length are f_x, f_y , offset denote c_x, c_y , field of view are FOV_x, FOV_y , and depth detection range present $d_{\min}, d_{\max}$.

$$C = \left\{ f_x \quad f_y \quad c_x \quad c_y \quad FOV_x \quad FOV_y \quad d_{\min} \quad d_{\max} \right\} \quad (19)$$

During the flight of a quadrotor, Euler angles Θ , encompassing the roll angle φ , pitch angle θ , and yaw angle ψ , serve often as the conventional parameters for denoting the attitude information. During the acquisition of visual information by the front-mounted camera, the roll angle merely induces an in-plane rotation of the captured image without altering its spatial information. Once the image undergoes horizontal correction based on

the roll angle, the pitch angle and yaw angle alone are considered in the ensuing visual guidance computation.

$$\begin{cases} \Theta = \{ \varphi & \theta & \psi \} \\ \Theta' = \{ \theta & \psi \} \end{cases} \quad (20)$$

$$\chi_t = \{ p \quad \Theta' \} = \{ x \quad y \quad z \quad \theta \quad \psi \} \quad (21)$$

In accordance with the camera's parameters, specifically the field of view angle FOV_c and depth detection range d_c , the field-of-view cone can be defined as follows.

$$\begin{cases} FOV_c = [-FOV_x/2, +FOV_x/2] \times [-FOV_y/2, +FOV_y/2] \\ d_c = [d_{min}, d_{max}] \\ p_c \in \Omega_c = d_c \cdot e^{i \cdot FOV_c} \end{cases} \quad (22)$$

Obstacles within the field-of-view cone are observable and can participate in the current decision-making process. At the current moment, the quadrotor processes external visual information I_t , integrates it with its own state χ_t and the expected point position $\tilde{p}$, and outputs the expected decision state $\tilde{\chi}_t$. Figure 1c presents the data captured by a real-world depth camera. In practical applications, the captured image undergoes filtering and processing to enhance its usability.

$$\tilde{\chi}_t = f(\chi_t, I_t, \tilde{p}) \quad I_t : \Omega_{img} \rightarrow \mathbb{R} \quad (23)$$

At its core, the obstacle avoidance strategy we proposed hinges on visual data to reroute around obstacles on the direct path from the current position to the target position. Depth information can easily separate the information in the image into foreground and background, as shown in Figure 5b. As shown in Figures 4b and 5b, visual data enables the identification of obstacle-free regions, the creation of safe flight areas for quadcopters, and the division of the decision-making space.

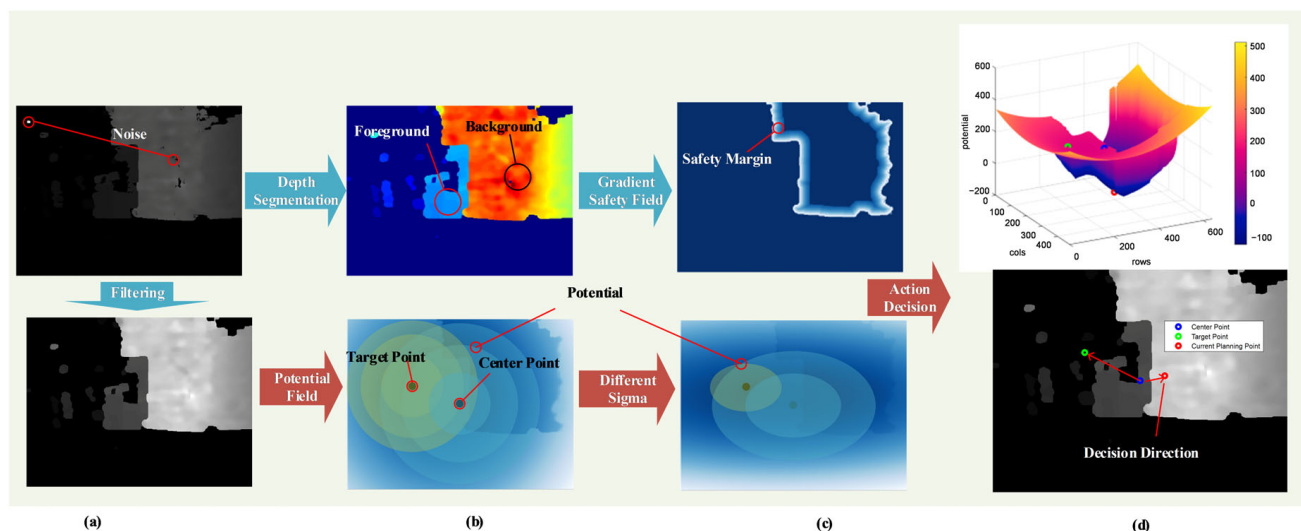


Figure 5. The processing and decision-making process of visual information in pigeon-inspired depth-reasoning-driven decision framework. (a) Depth image input by the original camera: The input information that can be used after depth information filtering. (b) Foreground and background of threshold segmentation after pseudo color display: The input information that can be used after depth information filtering. (c) Image of safety gradient field function and the image of non-uniform equipotential field function. (d) Decision potential field functions and results.

3.2. Behavioral Constraints Judgment

Pigeons instantaneously identify passable gaps by analyzing depth gradients on the retina. Like the pigeon's binocular disparity processing, we perform a real-time thresholding of the depth map to separate obstacles (foreground) from free space (background). The resulting binary mask is then analyzed to identify passable gaps without constructing any global map. The image will provide information about the location of obstacles. By integrating the quadrotor's current position with the target point's position, many logical decisions and actions can be initiated. The decision tree (Figure 5) with four decision attributes evaluates whether such a gap exists toward the target or current heading, replicating the avian "gap-first" decision process.

3.2.1. Heading Direction Judgment

Analogous to pigeons requiring a destination, such as the nest, during flight, the position of the target point remains one of the factors to be consistently considered throughout the flight of a quadrotor. Therefore, it is necessary to judge based on the deviation between the current forward direction and the target point position. If the target point is not in the field-of-view cone of the current camera, it will be considered that the heading is deviated at this time.

$$u_t^1 = \begin{cases} 1 & , p_c^{target} \in \|p_c^{target}\| \cdot e^{i \cdot FOV_c} \\ 0 & , p_c^{target} \notin \|p_c^{target}\| \cdot e^{i \cdot FOV_c} \end{cases} \quad (24)$$

3.2.2. Target-Point-Direction Obstacle Judgment

We select a high-confidence depth threshold T_d , satisfying $T_d \in d_c$, to segment the closed area occupied by the foreground obstacle Ω_{obs} and the flyable background area without obstacles Ω_{bg} . The boundary contour of the obstacle is represented by $\partial\Omega_{obs}$.

$$\begin{cases} \Omega_{obs} \cap \Omega_{bg} = \partial\Omega_{obs} \\ \Omega_{obs} \cup \Omega_{bg} = I_t \end{cases} \quad (25)$$

Define $d(p_{uv})$ as the minimum distance from a point to the obstacle boundary. The safety radius R_{safe} is the maximum distance of the safety zone Ω_{safe} . Then area $B_{R_{safe}}(p_{uv})$ represents a closed neighborhood with p_{uv} as the center and R_{safe} as the radius.

$$d(p_{uv}) = \text{mind}(p_{uv}, \partial\Omega_{obs}) \quad (26)$$

$$R_{safe} = \max_{p_{uv} \in \Omega_{safe}} d(p_{uv}) \quad (27)$$

$$B_{R_{safe}}(p_{uv}) = \{x \in \Omega_{img} \mid \|x - p_{uv}\| \leq R_{safe}\} \quad (28)$$

During flight, it is necessary to judge whether there are obstacles in the direction of the target point. If there is no obstacle interference between the own position and the target point, it means that obstacle avoidance is currently not required.

$$u_t^2 = \begin{cases} 1 & , \forall x \in B_{R_{safe}}(p_{uv}^{target}) , I_t(x) < T_d \\ 0 & , \exists x \in B_{R_{safe}}(p_{uv}^{target}) , I_t(x) \geq T_d \end{cases} \quad (29)$$

3.2.3. Forward-Direction Obstacle Judgment

The obstacle judgment in the forward direction is similar to the obstacle judgment in the target-point direction. In addition to the target-point direction, the direction of the quadrotor is also a special direction. No matter what direction it is, it is at the center point in the pixel coordinate system. Currently, it is necessary to judge whether there is an obstacle

in the direction of the center point. If there is no obstacle in front of the quadrotor, it means that there is no need to avoid obstacles without changing direction, which directly emulates the pigeon's strategy of minimizing posture changes during flight to avoid unnecessary attitude adjustments, which conserves energy.

$$u_t^3 = \begin{cases} 1 & , \forall \mathbf{x} \in B_{R_{safe}}(\mathbf{p}_{uv}^{center}) \quad , I_t(\mathbf{x}) < T_d \\ 0 & , \exists \mathbf{x} \in B_{R_{safe}}(\mathbf{p}_{uv}^{center}) \quad , I_t(\mathbf{x}) \geq T_d \end{cases} \quad (30)$$

where $\mathbf{p}_{uv}^{center} = \left[COL/2 \quad ROW/2 \right]^T$ is the center position of the image.

3.2.4. Gap Direction Judgment

During flight operations, the quadrotor may encounter not only common columnar obstacles but also wall-type obstacles. This can lead to a degradation in visual information quality, meaning the visual data obtained cannot match the richness available under ideal conditions. Under such circumstances, it becomes crucial to determine the direction of potential gaps and identify a navigable passage within the entire visual information framework that allows the quadrotor to pass through safely. In the absence of such a gap, the quadrotor should execute evasive maneuvers to exit the current field of view and explore alternative flight paths, which parallels the pigeon's instincts when no viable gap is perceived.

$$u_t^4 = \begin{cases} 1 & \exists \mathbf{p}_{uv} \in \Omega_{img} \quad , \forall \mathbf{x} \in B_{R_{safe}}(\mathbf{p}_{uv}) \quad , I_t(\mathbf{x}) \geq T_d \\ 0 & \forall \mathbf{p}_{uv} \in \Omega_{img} \quad , \exists \mathbf{x} \in B_{R_{safe}}(\mathbf{p}_{uv}) \quad , I_t(\mathbf{x}) < T_d \end{cases} \quad (31)$$

In the actual computation process, the quadrotor's safety region can be utilized for jump judgments. As mentioned above, pigeons are able to rapidly identify entire topological gaps as coherent regions through the lateral inhibition mechanism in the retina. Following this biological principle, our method focuses on identifying regions that meet threshold criteria, rather than individual pixels. Consequently, this approach significantly enhances operational speed and circumvents unnecessary repetitive searches. The extracted image information outcomes are systematically organized and documented as S_t to filter unsafe actions through constraint judgment.

$$S_t = \phi(I_t(\Omega_{img})) = \left[u_t^1 \quad u_t^2 \quad u_t^3 \quad u_t^4 \right]^T \quad (32)$$

3.3. Pigeon-Inspired Depth-Reasoning-Driven Action Decision

Drawing inspiration from the pigeon's ability to instantly select the safest and most efficient flight path among multiple visual gaps, our action decision module translates depth-derived safety fields into optimal trajectory directions. Based on the judgment criteria, we can distinguish between safe and unsafe regions. The tendency of pigeons to traverse gaps in a manner that minimizes changes to their posture can be modeled by a potential field that attains an extremum at the optimal location within the gap and gradually decays toward the boundaries. We further incorporate the target direction as an attractor, analogous to the bird's homing instinct, resulting in a composite field that balances gap centering with goal progression. Within the safe region, it is necessary to devise a behavioral decision-making function that enables the quadrotor to select an optimal trajectory within the redundant space, as shown in Figure 6.

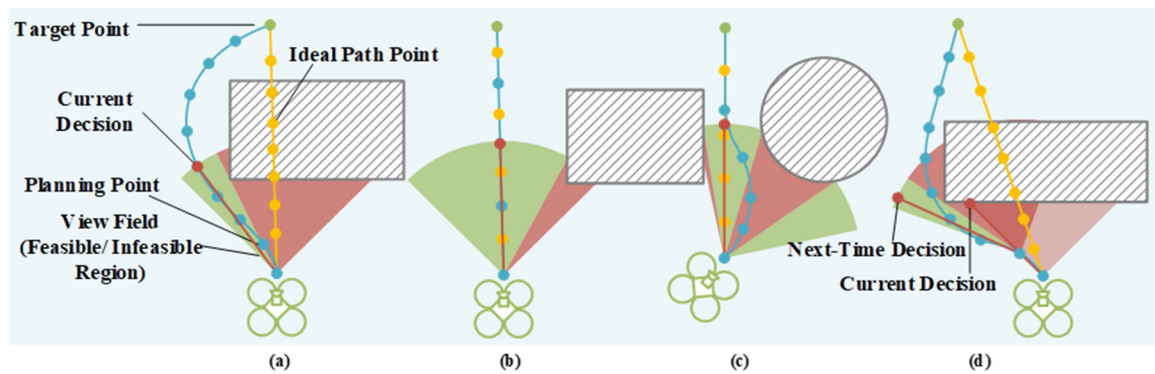


Figure 6. Schematic diagram of four quadrotor trajectory generation results. (a–d) The schematic results under different conditions.

3.3.1. Target-Point Direction Decision

To mathematically formalize this biological instinct, we map the visual saliency of the selected gap into a navigational potential field. Pigeons need to maintain their destination, such as their nest, during flight, which means that a centered gradient potential field function J_{target} must be created. This function will be used to enhance the spatial trajectory path of the quadrotor. This potential field function is based on a covariance matrix $\mathbf{R}_{\text{target}}$.

$$J_{\text{target}}(\mathbf{p}_{uv}) \triangleq (\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{target}})^T \mathbf{R}_{\text{target}} (\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{target}}) = \|\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{target}}\|_{\mathbf{R}_{\text{target}}}^2 \quad (33)$$

$$\mathbf{R}_{\text{target}} = \sigma_{\text{target}} \sigma_{\text{target}}^T = \begin{bmatrix} \sigma_{\text{target},1}^2 & \sigma_{\text{target},1} \sigma_{\text{target},2} \\ \sigma_{\text{target},2} \sigma_{\text{target},1} & \sigma_{\text{target},2}^2 \end{bmatrix} \quad (34)$$

3.3.2. Center-Point Direction Decision

As mentioned earlier, pigeons have a path selection mechanism that minimizes attitude changes during flight, so a mathematical model for this part needs to be added to the optimization terms. Like the target point, define another potential field source. This part can be understood as combining the result with the current posture information of the quadrotor.

$$J_{\text{center}}(\mathbf{p}_{uv}) \triangleq (\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{center}})^T \mathbf{R}_{\text{center}} (\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{center}}) = \|\mathbf{p}_{uv} - \mathbf{p}_{uv}^{\text{center}}\|_{\mathbf{R}_{\text{center}}}^2 \quad (35)$$

$$\mathbf{R}_{\text{center}} = \sigma_{\text{center}} \sigma_{\text{center}}^T = \begin{bmatrix} \sigma_{\text{center},1}^2 & \sigma_{\text{center},1} \sigma_{\text{center},2} \\ \sigma_{\text{center},2} \sigma_{\text{center},1} & \sigma_{\text{center},2}^2 \end{bmatrix} \quad (36)$$

We can also assign different weights σ to the two dimensions of the image, as shown in Figure 5c. This transforms the equipotential lines from circles into ellipses, thereby adjusting the emphasis on horizontal flight and climbing actions during flight.

3.3.3. Gradient Safety Field

In addition to the quadcopter's radius, this safety radius must also consider the distance from obstacles that is required due to airflow disturbances during flight. The obstacle avoidance behavior of pigeons is driven by the looming effect, which correlates with the rate of expansion of an object on the retina, enabling them to achieve smooth, oscillation-free flight trajectories even in confined spaces. Consequently, a gradual safety-

field function J_{safe} has been designed, where the potential field strength can be adjusted using a specific parameter k_{safe} .

$$J_{safe}(p_{uv}) = \begin{cases} f\left(\frac{d(p_{uv})}{R_{safe}}\right) \triangleq \exp\left(-k_{safe}\left(\frac{d(p_{uv})}{R_{safe}}\right)\right) & , p_{uv} \in \Omega_{safe} \\ 0 & , otherwise \end{cases} \quad (37)$$

The overall potential function is delineated as follows. Upon employing the proposed criteria to ascertain the necessity for obstacle avoidance, this potential function is utilized to derive the desired trajectory point, wherein the adjustable weight coefficients ω_i ($i = 1, 2, 3$).

This potential function represents a vector that points from the target direction to the desired direction within the safe space, under the influence of the safety potential field emanating from the obstacle boundary. This approach ensures that the trajectory constraints are not merely geometric boundaries, but represent a dynamic ‘safety envelope’ that reacts to the visual flux, wherein mapping bridges the gap between discrete perception and continuous control.

$$J_{potential} = \omega_1 \cdot J_{target} + \omega_2 \cdot J_{center} + \omega_3 \cdot J_{safe} \quad (38)$$

$$\mathbf{p}_c^* = \mathbf{K}_c^{-1} \mathbf{p}_{uv}^* = \mathbf{K}_c^{-1} \cdot \underset{u,v}{\operatorname{argmin}}(J_{total}) \quad (39)$$

The graph of this objective function is depicted in Figure 5d. While it is non-convex in its entirety, it exhibits local convexity. Provided that the initial value for the optimization is chosen as the target point position, the problem translates into a local convex optimization with an optimal solution.

3.4. Spatiotemporal Traversal Trajectory Generation

Once a pigeon selects a flight direction, it executes a smooth trajectory that respects its dynamic constraints. Our polynomial trajectory generation step corresponds to this motor planning phase, producing dynamically feasible paths that interpolate the waypoints derived from the potential field. The set of path points obtained in the above algorithm is $\mathcal{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_M\}$; the trajectory is divided into M segments, each of which is parameterized by the following L -order polynomial in $t \in [t_k, t_{k+1}]$.

$$\mathcal{J} = \sum_{k=0}^{M-1} \int_{t_k}^{t_{k+1}} \left\| \frac{d^3 \mathbf{p}_k(t)}{dt^3} \right\|^2 dt \quad (40)$$

Equation (40) defines the polynomial representation of each trajectory segment $\mathbf{p}_k(t)$, and order 3 is chosen to ensure the continuity of position, velocity, acceleration and jerk at segment boundaries. Minimizing jerk yields smooth, visually stable trajectories suitable for quadrotor flight. The objective function can be analytically expressed as a quadratic form with coefficients $\mathbf{a} = [\mathbf{a}_0^0, \dots, \mathbf{a}_1^L, \dots, \mathbf{a}_{M-1}^L]^T$.

$$\mathcal{L} = \mathbf{a}^T \mathbf{Q}(L) \mathbf{a} \quad (41)$$

where $\mathbf{Q}(L)$ is the block diagonal Hessian matrix associated with the polynomial order L . The trajectory satisfies the following constraints: path point constraints, derivative continuity constraints, and dynamic constraints.

$$\left\{ \begin{array}{l} \mathbf{p}_k(t_k) = \mathbf{p}_k^{start} \\ \mathbf{p}_k(t_{k+1}) = \mathbf{p}_k^{end} \\ \frac{d^m \mathbf{p}_{k-1}(t_k)}{dt^m} = \frac{d^m \mathbf{p}_k(t_k)}{dt^m}, m = 0, 1, 2 \\ \|\mathbf{v}(t)\| = \left\| \frac{d\mathbf{p}(t)}{dt} \right\| \leq \mathbf{v}_{max} \\ \|\mathbf{a}(t)\| = \left\| \frac{d^2\mathbf{p}(t)}{dt^2} \right\| \leq \mathbf{a}_{max} \end{array} \right. \quad (42)$$

Equation (42) summarizes the constraints that the trajectory must satisfy: waypoint constraints, continuity constraints and dynamic feasibility constraints. Together, these three equations define a standard minimum-snap trajectory optimization problem, which we solve using quadratic programming and time adjustment following established methods. In addition, the obstacle avoidance constraint is implicit in the path point set $\mathcal{P}$. As shown in Figure 4b, during the planning process, the limited space is divided into several safe spaces.

$$\mathcal{P}_{safe} = B_{\mathcal{R}}(\mathcal{P}) \quad \mathcal{R} = \mathbf{v}(t) \cdot \Delta t_c \pm R_{safe} \quad (43)$$

where Δt_c is the sampling interval of the camera. In the final trajectory, we use the decoupled optimization of time and space

$$\min_{\mathbf{a}, \mathbf{T}} \mathcal{J}(\mathbf{a}, L) + \omega_{\mathcal{J}} \sum_{k=0}^{M-1} \Delta t_k \quad (44)$$

where $\omega_{\mathcal{J}}$ is the optimization weight coefficient.

4. Experimental Results and Analysis

To validate the proposed framework, comprehensive experiments are conducted in both simulation environments and real-world scenarios. All tests emphasized autonomous navigation in unmapped 3D spaces without prior information.

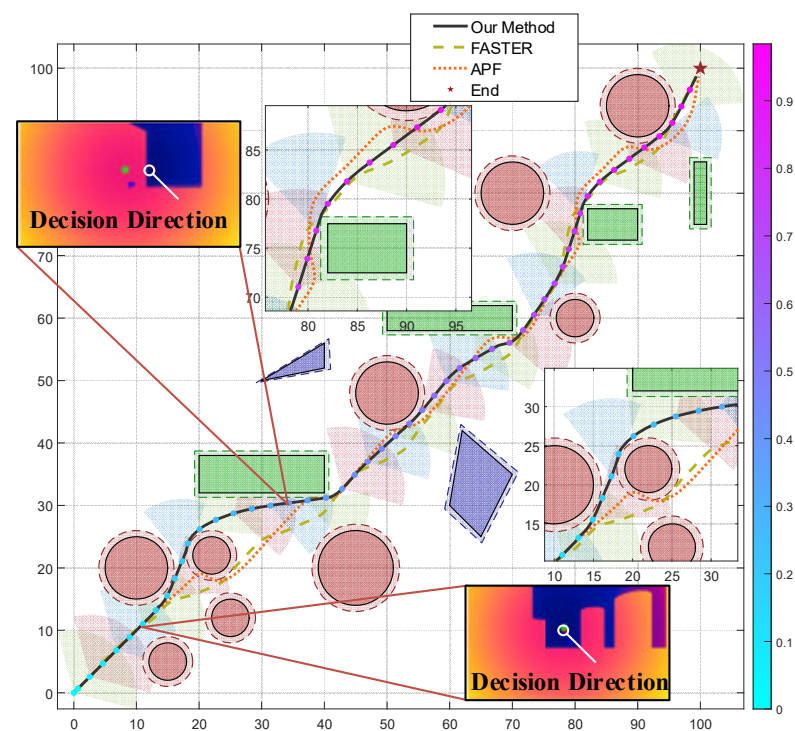
4.1. System Experiment

Simulations are executed in Gazebo, which models complex environments. The quadcopter equipped with a depth camera performs obstacle avoidance flight in a space full of obstacles and design a Visual-Inertial Odometer for real-time positioning during the flight. The algorithm and physical structure parameters are set as listed in Table 1, which includes the basic structural parameters of the camera and the basic physical parameters of the UAV, all of which are set to appropriate values. The parameter weights of the method proposed in this paper are fitness values obtained through experimental tuning to ensure the best trade-off between smoothness and collision avoidance. The safety margin pixel value is calibrated based on the depth sensor resolution and is positively correlated with the safety radius of the gap expansion required to identify the gap at flight speed. To rigorously evaluate the proposed pigeon-inspired depth-reasoning-driven decision framework, we conducted comparative experiments against FASTER [31] and the Artificial Potential Field (APF) method. It is important to note that for the purposes of these comparative baselines, both FASTER and APF were provided with pre-processed local maps or explicit obstacle position information as inputs to ensure their standard operational requirements. In contrast, our proposed method operates directly on raw, unprocessed depth information without any prior knowledge or map construction.

Table 1. The algorithm and physical structure parameters.

Parameters	Values	Parameters	Values
FOV	120°	T_d	100
$d_{\min}$	0 m	$d_{\max}$	10 m
ω_1	0.0008	ω_2	0.001
ω_3	100	R_{safe}	30 px
Δt_c	0.2 s	Δt_k	0.2 s
$\omega_{\mathcal{J}}$	1.5	$\omega_{\max}$	45°/s
$a_{\min}$	−0.5 m/s ²	$a_{\max}$	0.2 m/s ²
$v_{\min}$	0 m/s	$v_{\max}$	1 m/s

Figures 7 and 8 present the flight path of the quadrotor in 2D and 3D, respectively, combining real-time visual information to perform dynamic obstacle avoidance, and mark the location of obstacles in the environment. The color image of the camera is used for identification and positioning, and the depth map is used for obstacle avoidance and positioning. Near the coordinate (10, 10, 5), the depth information detects the existence of an obstacle, as shown in the depth map in Figure 7. This causes a significant deviation in the original path, forming a local avoidance trajectory. Similar avoidance is also reflected in the flat obstacle near the point (35, 30, 5). And the avoidance process is always kept outside the safe distance.

**Figure 7.** The 2D flight trajectory image of the quadrotor. The fan-shaped area is part of the visual range detected by the camera.

Despite the obstacle avoidance maneuver, the overall path is relatively smooth, without violent reentry or oscillation, indicating that the path planning algorithm takes into account flight efficiency while ensuring safety. This figure strongly proves that the fused visual information, especially the accurate distance perception provided by the depth map, is the key to achieving robust, real-time obstacle avoidance and traversal flight.

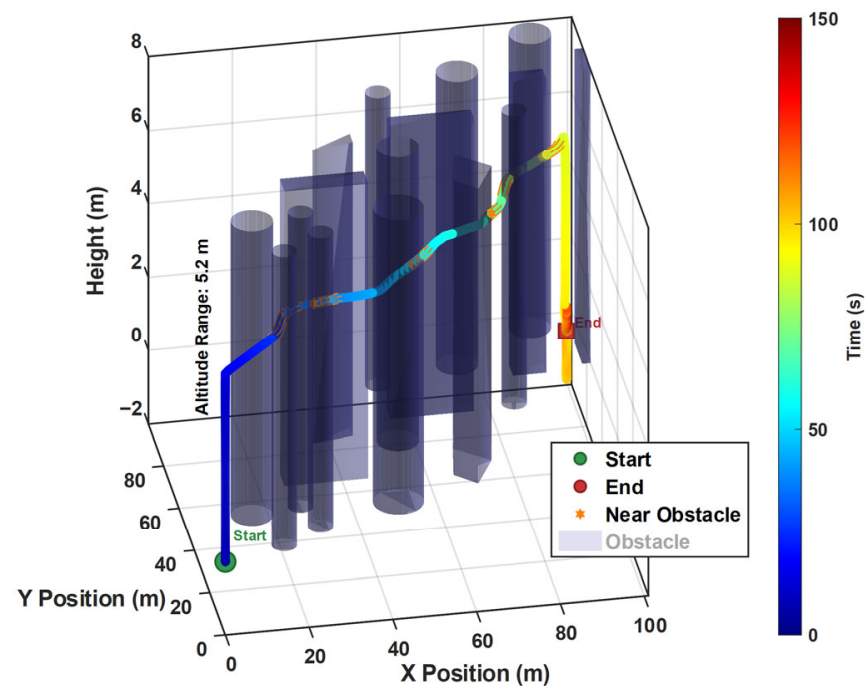


Figure 8. The 3D image of the quadcopter flying through space with obstacle avoidance. The translucent cylinders are obstacles, and the colored curves are the flight paths of the quadrotors over time.

The average values from multiple runs of the experiment are shown in Table 2, where $\text{smoothness} = 1 / (1 + \bar{\kappa}) \in (0, 1]$; $\bar{\kappa}$ is the average curvature along the trajectory weighted by arc length and higher values indicate smoother paths. While FASTER and APF achieve smooth obstacle avoidance using the provided local maps, they incur significant hidden costs in real-world scenarios due to the necessity of map building (e.g., voxel grid or ESDF updates). Importantly, the reported calculation time corresponds exclusively to its trajectory optimization stage; map construction was performed offline prior to planning and is not included at this time. This difference arises from the fundamental algorithmic distinction: FASTER performs iterative gradient-based optimization on a high-resolution ESDF grid, while our method directly reasons on raw depth pixels using a lightweight potential field, requiring no discretization or iterative solver. In contrast, APF achieves the lowest computation time but suffers from oscillations and local minima due to its reactive nature. Our method eliminates this dependency, directly translating pixel-level depth reasoning into control constraints, which is inherently more robust to the latencies typically introduced by mapping threads in unmapped 3D environments. The experimental results in Figure 7 illustrate that, while FASTER produces more optimal paths by exploiting known obstacle geometries, its performance is strictly bound by the accuracy of the map. In contrast, our approach demonstrates comparable safety despite the lack of a local map. By reasoning directly from the raw depth field, our method avoids the mapping lag and potential discretization errors inherent in voxel-based representations. While APF remains a fast reactive baseline, its reliance on a clear potential field gradient often leads to suboptimal oscillations, whereas our depth-reasoning logic generates smoother, more dynamically feasible trajectories by considering the 3D geometric constraints of the FOV. Please refer to Supplementary Material Figure S1 for the overall flowchart of the method.

Table 2. Algorithm comparison results in simulation.

Algorithm	Trajectory Length	Track Time	Smoothness	Calculation Time	Minimum Distance	Proportion Less than 0.85 m
Our Method	147.0 m	149.4 s	0.961	30.836 s	0.806 m	0.40%
FASTER	145.8 m	148.2 s	0.943	143.798 s	1.038 m	0.00%
APF	151.4 m	153.8 s	0.919	0.087 s	0.309 m	2.73%

4.2. Real-World Experiment

In addition to the simulation experiment, we conducted actual flight verification in our existing environment under multiple conditions. The experimental site and equipment are shown in Figure 9. In our real-world flight experiments, the quadrotor platform is equipped with an Intel Jetson Xavier NX as the primary onboard computer for high-level tasks, including VIO processing and the proposed pigeon-inspired depth-reasoning-driven decision algorithm. A CUAV X7+ flight controller is utilized for low-level attitude and position control. To ensure the reliability of depth information—upon which our map-less trajectory planning heavily relies—we selected a relatively moderate and structured experimental environment. This choice was made to maintain the stability of the raw depth data provided by the stereo camera, thereby minimizing the impact of severe sensor noise and allowing for a more focused evaluation of the planning algorithm’s decision-making logic and real-time performance in unmapped 3D spaces. The depth camera we used was Intel’s D435i camera, and its parameter information is shown in Table 3. The depth camera receives information about the surrounding environment and generates decision information to guide the quadrotor to fly quickly through a space full of obstacles.

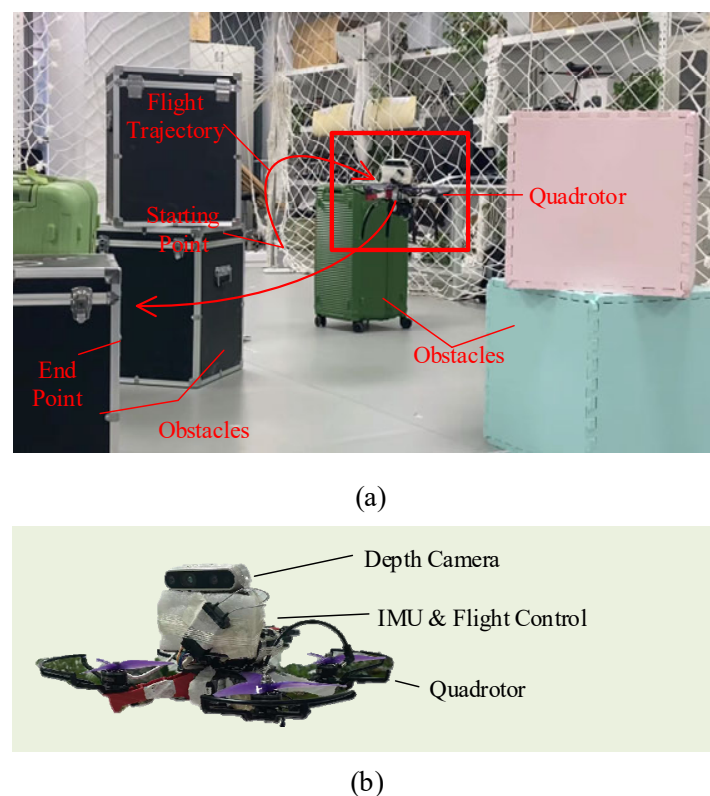
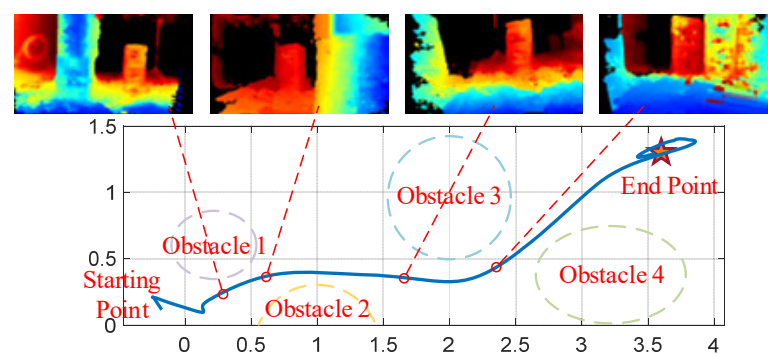
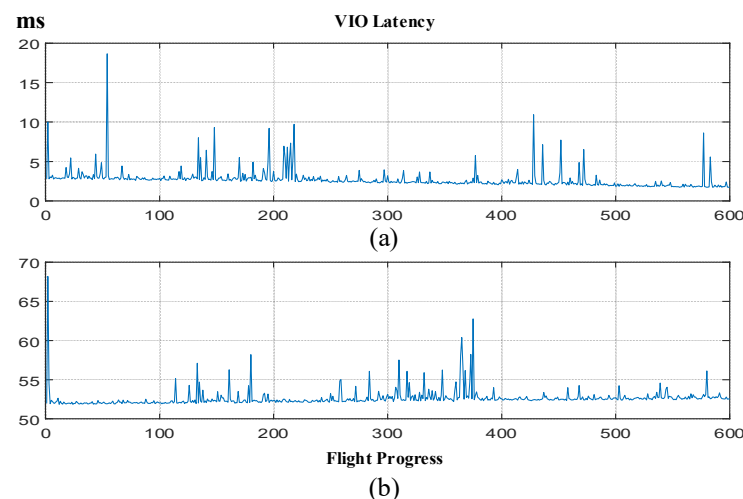


Figure 9. Flight verification test of actual quadrotor. (a) Test site and process. (b) Quadrotor with depth camera we used.

Table 3. Depth camera parameter information.

Parameter	Value
Depth Detection Range	0.2~10 m
Binocular Baseline	50 mm
Depth Stream Frequency	Up to 90 FPS

As shown in Figure 10, the quadrotor successfully navigated through a cluttered environment containing four obstacles with an average spacing of 1 m. Despite the constrained space, the trajectory remained smooth with a total path length of 4.4 m. As shown in Figure 11, the VIO delay characteristics reveal robust performance under real-flight conditions, with an average processing latency of 52.6 ms. And during the entire verification process, the counter is not reset, which proves its robustness.

**Figure 10.** Actual obstacle avoidance flight trajectory and depth information image of quadrotor aircraft. In the depth map, the color spectrum transitions continuously from blue (nearest) to red (farthest).**Figure 11.** Delay of VIO during flight and mission process. (a) ORB-based VIO. (b) SuperGlue-based VIO.

Similar to the results in the simulation, the flight speed and attitude angle of the quadrotor can be maintained within a stable range during the actual flight to ensure a smooth path as shown in Figure 12. During the actual flight, due to the limitation of the site size, in order to ensure the safety of the flight, we limited the speed of the quadrotor. The results showed that the quadrotor can still maintain a flight speed of about 0.5 m/s while performing the obstacle avoidance and crossing flight mission. After multiple actual flight

verifications, the results show that the proposed pigeon-inspired depth-reasoning-driven decision framework is designed for high-speed traversal in unmapped environments where computational efficiency is paramount. However, its performance is subject to certain constraints and environmental factors.

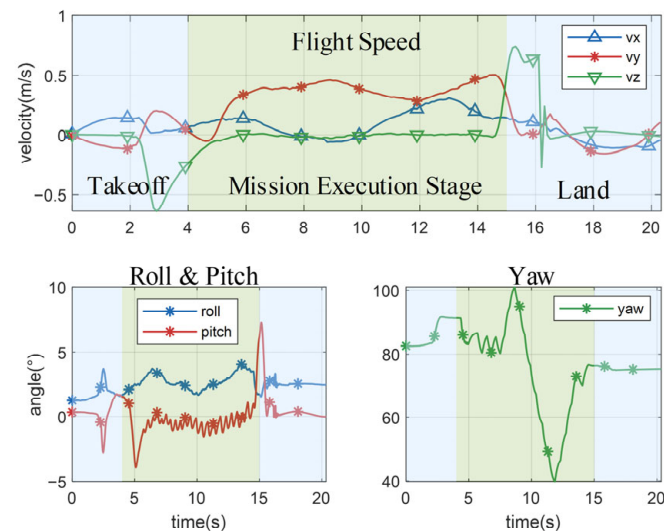


Figure 12. The attitude curve and speed curve of the quadrotor during obstacle avoidance and crossing flight.

As demonstrated in practical validation, our proposed pigeon-inspired depth-reasoning-driven decision fundamentally relies on the quality of raw depth inputs. The entire decision-making pipeline, from gap detection to safety field construction, operates directly on depth measurements without explicit map filtering or temporal accumulation. Consequently, the performance of the framework is inherently sensitive to the accuracy of the depth sensor. In environments with highly reflective surfaces or extreme lighting conditions that cause depth holes or significant noise, the depth-reasoning module may generate suboptimal or overly conservative constraints. Although the filtering and weighting-based coupling methods employed in this approach can mitigate such dependency to a certain extent, it must be acknowledged that the reliability of the depth data acquired by the sensor remains a critical factor. Although we employ SuperGlue to enhance matching robustness, any significant drift in state estimation will directly translate into errors in the world-frame projection of depth constraints, potentially leading to collision risks during high-speed maneuvers.

5. Conclusions

This paper presents a pigeon-inspired depth-reasoning-driven decision framework designed to overcome the computational bottlenecks and mapping latencies inherent in autonomous quadrotor flight in unmapped 3D spaces. The framework directly emulates three key avian navigation strategies: gap selection via instantaneous depth gradients, path selection that minimizes posture changes, and a safety field driven by the looming effect. By bypassing traditional spatial discretization and explicit map construction, the proposed system establishes a direct link between raw sensory depth perception and trajectory generation.

First, to realize the gap selection mechanism, the framework segments raw depth images in real time to identify passable topological gaps without constructing a global map. Second, to replicate the path selection strategy that minimizes posture changes, we construct a gradient potential field that guides the quadrotor toward the optimal position of

the selected gap. This encourages smooth, minimally oscillatory trajectories, analogous to the pigeon's tendency to maintain flight stability during gap traversal. Third, to implement the looming-effect-driven safety field, we define a non-linear repulsive potential whose strength increases inversely with distance to obstacles.

Together, these three biologically inspired components form a constraint-fusion planning strategy that combines hard safety limits with soft optimization to generate dynamically feasible trajectories from sparse waypoints, replicating the instantaneous trade-off between collision avoidance and flight efficiency observed in nature. To enable this framework in practice, we integrate a robust perception backbone that combines depth-guided visual-inertial odometry with deep-learning-based feature matching, providing the necessary state estimation stability under aggressive maneuvers.

Experimental validation confirms the framework's effectiveness in unmapped 3D spaces. Future efforts will focus on exploring more profound biological mechanisms and enhancing the system's capability to handle highly dynamic obstacles, further optimizing computational efficiency for onboard deployment, and extending its application to large-scale, long-duration autonomous exploration missions.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/biomimetics11040283/s1>, Figure S1: Pigeon-Inspired Depth-Reasoning-Driven Decision Framework for Quadrotors Diagram.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; software, R.S.; validation, R.S.; formal analysis, Y.S.; investigation, Y.S. and R.S.; resources, Y.S.; data curation, Y.S. and R.S.; writing—original draft preparation, Y.S. and R.S.; writing—review and editing, Y.S. and R.S.; visualization, Y.S. and R.S.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Key R&D Program of China under Grant 2023YFC3011001; in part by the Beijing Natural Science Foundation under Grant 4262053; and in part by the National Natural Science Foundation of China under Grant 62473025 and Grant 62533006; in part by the State Key Laboratory of Autonomous Intelligent Unmanned Systems Opening Project under Grant ZZKF2025-1-9.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data Availability Statement: The original contributions presented in the study are included in the article/Supplementary Material.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, C.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; Li, H. End-to-end autonomous driving: Challenges and frontiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10164–10183.
2. 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. [\[CrossRef\]](#)
3. Rebecq, H.; Horstschaefter, T.; Gallego, G.; Scaramuzza, D. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time. *IEEE Robot. Autom. Lett.* **2017**, *2*, 593–600. [\[CrossRef\]](#)
4. Ekstrom, D. Why vision is important to how we navigate. *Hippocampus* **2015**, *25*, 731–735. [\[CrossRef\]](#)
5. Servièrès, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking. *J. Sens.* **2021**, *2021*, 2054828. [\[CrossRef\]](#)
6. Zhu, C.; Meurer, M.; Günther, C. Integrity of Visual Navigation—Developments, Challenges, and Prospects. *J. Inst. Navig.* **2022**, *69*, navi.518. [\[CrossRef\]](#)
7. Qin, T.; Li, P.; Shen, S. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1019. [\[CrossRef\]](#)

8. Qin, T.; Pan, J.; Cao, S.; Shen, S. A general optimization-based framework for local odometry estimation with multiple sensors. *arXiv* **2019**, arXiv:1901.03638. [[CrossRef](#)]
9. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
10. Engel, J.; Schöps, T.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [[CrossRef](#)] [[PubMed](#)]
11. Mourikis, A.I.; Roumeliotis, S.I. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy*; IEEE: New York, NY, USA, 2007; pp. 3565–3572.
12. Bloesch, M.; Burri, M.; Omari, S.; Hutter, M.; Siegwart, R. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *Int. J. Robot. Res.* **2017**, *36*, 1053–1072. [[CrossRef](#)]
13. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-Direct Monocular Visual Odometry. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*; IEEE: New York, NY, USA, 2014; pp. 15–22.
14. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
15. 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](#)]
16. Harris, C.; Stephens, M. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference, Manchester, UK*; The Plessey Company: Plymouth, UK, 1988; pp. 147–151.
17. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain*; IEEE: New York, NY, USA, 2011; pp. 2564–2571.
18. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
19. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In *Computer Vision—ECCV 2006*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 430–443.
20. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision—ECCV 2010*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6314, pp. 778–792.
21. Teed, Z.; Deng, J. DeepV2D: Video to depth with differentiable structure from motion. *arXiv* **2018**, arXiv:1812.04605.
22. Teed, Z.; Deng, J. Droid-SLAM: Deep visual slam for monocular, stereo, and RGB-D cameras. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 16558–16569.
23. Wang, Z.; Zhou, X.; Xu, C.; Gao, F. Geometrically constrained trajectory optimization for multicopters. *IEEE Trans. Robot.* **2022**, *38*, 3259–3277. [[CrossRef](#)]
24. Gao, F.; Wang, L.; Zhou, B.; Zhou, X.; Pan, J.; Shen, S. Teach-Repeat-Replan: A Complete and Robust System for Aggressive Flight in Complex Environments. *IEEE Trans. Robot.* **2020**, *36*, 1528–1544. [[CrossRef](#)]
25. Wang, M.; Wang, Q.; Wang, Z.; Gao, Y.; Wang, J.; Cui, C.; Li, Y.; Ding, Z.; Wang, K.; Xu, C. Unlocking aerobatic potential of quadcopters: Autonomous freestyle flight generation and execution. *Sci. Robot.* **2025**, *10*, eadp9905. [[CrossRef](#)] [[PubMed](#)]
26. Dong, L.; Jiang, F.; Peng, Y. Attention-Based UAV Trajectory Optimization for Wireless Power Transfer-Assisted IoT Systems. *IEEE Trans. Ind. Electron.* **2025**, *72*, 8463–8471. [[CrossRef](#)]
27. Xu, Z.; Han, X.; Shen, H.; Jin, H.; Shimada, K. NavRL: Learning safe flight in dynamic environments. *IEEE Robot. Autom. Lett.* **2025**, *10*, 3668–3675. [[CrossRef](#)]
28. Lu, Y.; Xie, F.; Ji, T. Human-in-the-loop reinforcement learning for dynamic soaring: A trajectory planning and control integrated system. *Eng. Appl. Artif. Intell.* **2025**, *157*, 111219. [[CrossRef](#)]
29. Chai, Y.; Wang, Y.; Yang, X.; Im, S.-K.; He, Q. UAVRM-A*: A Complex Network and 3D Radio Map-Based Algorithm for Optimizing Cellular-Connected UAV Path Planning. *Sensors* **2025**, *25*, 4052. [[CrossRef](#)]
30. Ratliff, N.; Zucker, M.; Bagnell, J.A.; Srinivasa, S. CHOMP: Gradient optimization techniques for efficient motion planning. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan*; IEEE: New York, NY, USA, 2009; pp. 489–494.
31. Tordesillas, J.; Lopez, B.T.; Everett, M.; How, J.P. FASTER: Fast and Safe Trajectory Planner for Navigation in Unknown Environments. *IEEE Trans. Robot.* **2022**, *38*, 922–938. [[CrossRef](#)]
32. Huo, M.; Duan, H.; Yang, Q.; Zhang, D.; Qiu, H. Live-fly experimentation for pigeon-inspired obstacle avoidance of quadrotor unmanned aerial vehicles. *Sci. China Inf. Sci.* **2019**, *62*, 52201. [[CrossRef](#)]
33. Brighton, C.H.; Kloepper, L.N.; Harding, C.D.; Larkman, L.; McGowan, K.; Zusi, L.; Taylor, G.K. Raptors avoid the confusion effect by targeting fixed points in dense aerial prey aggregations. *Nat. Commun.* **2022**, *13*, 4778. [[CrossRef](#)]
34. Caraco, T.; Martindale, S.; Pulliam, H. Avian flocking in the presence of a predator. *Nature* **1980**, *285*, 400–401. [[CrossRef](#)]
35. Brighton, C.H.; Taylor, G.K. Hawks steer attacks using a guidance system tuned for close pursuit of erratically manoeuvring targets. *Nat. Commun.* **2019**, *10*, 2462. [[CrossRef](#)]

36. Farine, D.R.; Aplin, L.M.; Garroway, C.J.; Mann, R.P.; Sheldon, B.C. Collective decision making and social interaction rules in mixed-species flocks of songbirds. *Anim. Behav.* **2014**, *95*, 173–182. [[CrossRef](#)]
37. Duan, H.; You, L.; Fan, Y.; Li, M. Phase transition control of UAV swarm based on bird-inspired self-propelled mechanism. *Acta Autom. Sin.* **2025**, *51*, 960–971.
38. KleinHeerenbrink, M.; France, L.A.; Brighton, C.H.; Taylor, G.K. Optimization of avian perching manoeuvres. *Nature* **2022**, *607*, 91–96. [[CrossRef](#)] [[PubMed](#)]
39. Wüest, V.; Jeger, S.; Feroskhan, M.; Ajanic, E.; Bergonti, F.; Floreano, D. Agile perching maneuvers in birds and morphing-wing drones. *Nat. Commun.* **2024**, *15*, 8330. [[CrossRef](#)]
40. Lin, H.; Ros, G.I.; Biewener, A.A. Through the eyes of a bird: Modelling visually guided obstacle flight. *J. R. Soc. Interface* **2014**, *11*, 20140239. [[CrossRef](#)]
41. Williams, D.C.; Biewener, A.A. Pigeons trade efficiency for stability in response to level of challenge during confined flight. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 3392–3396. [[CrossRef](#)] [[PubMed](#)]
42. Wang, Y.; Frost, J.B. Time to collision is signalled by neurons in the nucleus rotundus of pigeons. *Nature* **1992**, *356*, 236–238. [[CrossRef](#)] [[PubMed](#)]
43. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA; IEEE: New York, NY, USA, 2018; pp. 1650–1659.
44. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 4938–4947.
45. Saad, M.; Bonilla, L.D.; Bassma, G.; Ghogho, M. Deep Learning Techniques for Visual SLAM: A Survey. *IEEE Access* **2023**, *11*, 20026–20050. [[CrossRef](#)]
46. Xu, K.; Hao, Y.; Yuan, S.; Wang, C.; Xie, L. AirSLAM: An Efficient and Illumination-Robust Point-Line Visual SLAM System. *IEEE Trans. Robot.* **2025**, *41*, 1673–1692. [[CrossRef](#)]
47. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [[CrossRef](#)]
48. Shen, S.; Michael, N.; Kumar, V. Tightly-Coupled Monocular Visual-Inertial Fusion for Autonomous Flight of Rotorcraft MAVs. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA; IEEE: New York, NY, USA, 2015; pp. 5303–5310.

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.