FollowFusion: Online MVS for Viewpoint Adaptive Terrain

Zhenyu Xia[1], Shuhui Bu[1*], Lin Chen[1], Kun Li[1], Xuan Jia[1], Xuefeng Cao[2]

***Abstract*— As Unmanned Aerial Vehicle (UAV) technology progresses, aerial photography for geographic information acquisition is increasingly employed across various applications. Consequently, the reconstruction of regions of interest using UAV has garnered significant attention. By meticulously designing flight trajectories and photographic strategies, it is possible to generate precise 3D models, thereby streamlining the reconstruction process for large-scale scenes. However, existing research predominantly relies on an explore-then-exploit strategy for 3D reconstruction, often utilizing pre-existing models. To overcome this constraint, we introduce a novel view-path-planning approach that intelligently selects optimal viewpoints, coupled with an online Multi-View Stereo (MVS) reconstruction algorithm to generate high-quality 3D models. Building upon Complete Coverage Path Planning (CCPP), our method dynamically adjusts the Above Ground Level (AGL) to maintain a consistent Ground Sample Distance (GSD), while concurrently leveraging terrain data to enhance MVS performance. Our experiments demonstrate that the proposed algorithm is capable of producing high-quality 3D models in a single flight, independent of any prior information.**

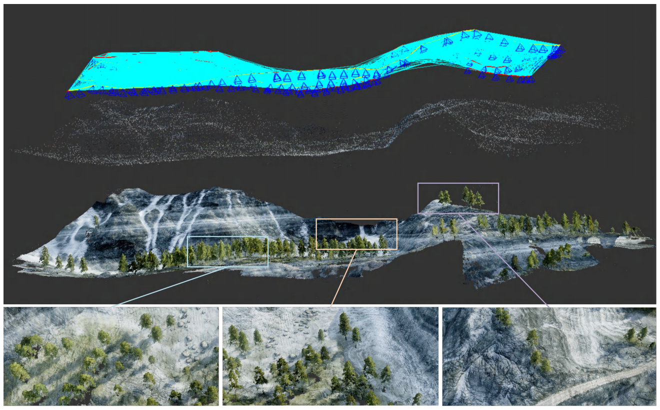

Fig. 1. Illustration of viewplanning and online MVS system. The top layer presents an optimal trajectory, crafted from the terrain to ensure the best viewpoints. The middle layer comprises a sparse point cloud map, serving as a foundational representation of the terrain. The bottom layer showcases a dense map, dynamically generated in real time by our online MVS system.

I. INTRODUCTION

In recent years, the application of 3D reconstruction technology has expanded significantly across various domains, including digital mapping, the Unreal Engine, and the realms of Augmented Reality (AR) and Virtual Reality (VR). The capability to reconstruct real-time maps has become increasingly vital for tasks such as detection, search and rescue operations, and geological surveys. To obtain high-quality maps, Multi-View Stereo (MVS) algorithm [1] [2] has been widely adopted. This algorithm is capable of producing high-quality 3D models from a series of input images.

To better acquire images for MVS reconstruction, a highly flexible and maneuverable UAV equipped with a camera is usually used to photograph the area of interest. However, employing UAV for MVS reconstruction presents challenges. UAV are constrained by battery capacity, necessitating the collection of images within a finite time. Additionally, the reconstruction efficacy of MVS is influenced by the target structure and the image dataset [3] [4]. Therefore, view path planning algorithms are extensively utilized. Numerous prior approaches [5]–[8] have adopted an explore-then-exploit strategy, requiring either two passes over the trajectory or dependence on preliminary models to establish the reconstruction path. This strategy is initiated by establishing a preliminary, coarse model of the scene through a simple, fixed trajectory scan within a secure zone. Subsequently, it devises an inspection route designed to encompass the complete exterior of the coarse model. This strategy has limitations: it requires a prior model, leading to multiple flight plans and reduced task efficiency, and the complex computations involved in processing input images by MVS methods can slow down the overall workflow.

To address the above issues, we propose an online MVS system based on the view path planning method. This algorithm is proficient at constructing large-scale models by prioritizing the most effective local dense mappings. The algorithm uses filtering steps to handle noise and outliers, thereby ensuring both rapid reconstruction and high-quality output. In contrast to offline methods, the proposed method analyzes sparse point cloud maps in real time, offering terrain feedback below the UAV. It utilizes heuristic information from MVS to determine the optimal view for capturing images. With the best view secured, the UAV dynamically adjusts its flight trajectory and viewpoints to more effectively cover complex terrain.

Our proposed method is benchmarked against other techniques in a series of experiments. The results indicate that our method achieves greater efficiency and superior reconstruction quality in the benchmark scenarios. The contribution of this study is summarized as follows:

1) In contrast to the prevalent explore-then-exploit paradigms, we introduce an innovative framework for autonomous 3D modeling that encompasses online MVS reconstruction, and an exploration planning algorithm to ensure enhanced 3D reconstruction efficiency and accuracy.

2) Our novel approach diverges from conventional techniques by leveraging an online MVS system grounded in

*Corresponding author (bushuhui@nwpu.edu.cn).

[1] School of Aeronautics, Northwestern Polytechnical University.

[2] PLA Strategic Support Force Information Engineering University.

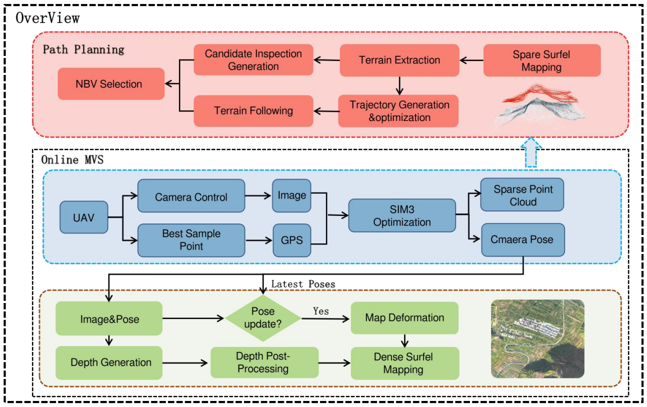

Fig. 2. The overview of the proposed path planning and online MVS algorithm.

monocular algorithms for the construction of 3D models. This innovation effectively mitigates the inefficiency associated with traditional offline algorithms, offering a more time-efficient solution for 3D modeling.

3) The algorithm eliminates the necessity for prior proxy models to generate real-time acquisition trajectories. The meticulously calculated paths ensure comprehensive coverage of high-quality surfaces, optimizing the performance of MVS reconstruction to its fullest potential.

4) We assess the efficacy of our method through our UAV-based experiences in three simulated and one real-world scenario, showcasing an improved performance over existing methods.

II. RELATED WORK

A. 3D Scene Reconstruction

Image-based 3D scene reconstruction has seen widely development over the past few decades, with numerous solutions emerging and being applied in practical scenarios. The representative commercial includes ContextCapture [9], Pix4Dmapper [10], and Metascape [11]. ContextCapture is one of the popular 3D reconstruction software packages that provides a friendly graphics user interface and graphics processing unit (GPU) acceleration to generate high-resolution 3D models. Agisoft Metashape is a reconstruction software with a fully automatic workflow that uses a multi-view matching algorithm to generate accurate 3D models from multiple image data. Pix4Dmapper is primarily used for 3D reconstruction of aerial photography, and is used not only for generating 3D models, but also for generating Digital Surface Models (DSMs) and Digital Orthophotographs (DOMs) from images.

Many researches also provide open-source 3D reconstruction methods such as COLMAP [1] [12] [13], OpenMVG [14] and OpenMVS [15]. These open-source libraries offer a comprehensive set of algorithms for recovering scene surfaces from multiview images, grounded in the traditional geometric approach to 3D reconstruction. Additionally, there are also some deep learning-based 3D reconstruction methods, Cas-MVSNet [16], MS-REDNet [17] employ an inference process that refines the model iteratively from coarse to fine resolutions. VIS-MVSNet [18] and PVS-Net [19] consider pixel visibility information for depth map estimation.

However, the above methods either perform offline 3D reconstruction or necessitate pre-training of the data, which is inconvenient for scenarios demanding immediate processing capabilities. Our proposed method not only meets the real-time requirements but also ensures the precision of the reconstruction result, offering a practical solution for complex environments and real-time applications.

B. Path Planning for Aerial Reconstruction

The problem of photographing in 3D reconstruction by constantly searching for the best viewpoint is known as view planning or active vision [20]. Certain commercial software, such as those referenced in [21] [22], perform aerial photography by merely scanning the area of interest from a favorable position above. However, these solutions often neglect the intricacies of the terrain's structure. To recovery fine details, [23] adopts an explore-then-exploit approach. It starts by designing simple trajectories to obtain a rough model of the scene within a safe range, then designing finer trajectories with sample points based on the geometric configuration of the model, rescanning and reconstructing the final 3D model of the target structure.

The [24]–[26] predict unknown scenes and adjust UAV sampling points by online methods but still do not escape the use of offline reconstruction. [27] [28] real-time generative paths for continuous exploration of the scene and real-time reconstruction of the scene based on the methods of REMODE [29] and Cas-MVSNet, respectively.

In the paper, we consider the factors that affect MVS performance during the UAV's flight photography process and use the results of analyzing the terrain to design the MVS heuristic to determine the camera's position, which allows for real-time 3D reconstruction of scenes with changing terrain.

III. SYSTEM OVERVIEW

The purpose of this study is to achieve efficient reconstruction of complex terrain scenes without any prior information. The Fig.2 illustrates the overall framework of the system, which consists of online MVS (Sect.IV), and heuristic planning (Sect.V). The online MVS employs Simultaneous Localization and Mapping (SLAM) to calculate the poses of the images and subsequently stores these poses along with the corresponding point clouds (Sect.IV-A). Following this, the depth estimation phase (Sect.IV-B) generates depth maps leveraging the estimated image poses, while the noise filtering step (Sect.IV-C) refines these maps by eliminating noise. In instances where a frame is optimized multiple times by SLAM, the associated point clouds are updated in accordance with the latest pose data (Section IV-D). The online planning module then executes real-time terrain analysis, utilizing the stored sparse point cloud data to identify the most effective viewpoints. This process is designed to optimize global coverage efficiency and enhance

MVS performance. The UAV's mission ends after completing the scanning of the region of interest.

IV. Online Multiview Stereo

The online MVS system is based on a monocular dense mapping algorithm. This module relies on SLAM [30] to obtain camera poses. Additionally, it integrates GNSS information to determine the absolute geographic position of the UAV. For our proposed method, each frame is the keyframe [31]. The computed camera poses and sparse point cloud for each keyframe are stored in a database. he keyframe and its pose are utilized for depth estimation, while the point cloud is employed for estimating the topography of the region of interest. The algorithm leverages subsequent sequence images as source images for stereo matching, depth estimation, and filtering in an online manner, thereby enhancing depth accuracy.

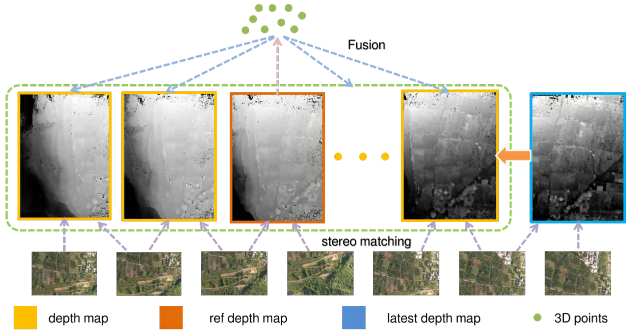

Fig. 3. Illustration of noise filtering for obtaining depth maps.(Sect.IV-C)

A. GNSS Fusion

To achieve a more precise alignment of the generated dense maps with the absolute geographic position, we integrate visual information from the camera with its the GNSS data. This integration not only enhances the robustness of the visual data but also compensates for the lack of scale inherent in monocular SLAM systems. For the convenience and extendibility, we implemented a SLAM based on GSLAM framework [30].

For each frame captured by the camera, we employ SiftGPU [32] to extract feature and descriptor that exhibit superior scale invariance and robustness. Upon securing a set of effective descriptor matches, the relative pose between the images is determined by applying singular value decomposition (SVD) to the essential matrix. Then the position data is fused with the GNSS information using a SIM3 [33] transformation, aligning it with the Earth-Centered, Earth-Fixed (ECEF) coordinate system. For subsequent image sequences captured by the UAV's cameras, we employ a continuous Perspective-n-Point (PnP) algorithm to obtain poses with scale. We then iteratively refine the camera pose by optimizing the reprojection error and GNSS error, ensuring high accuracy and reliability in the final 3D reconstruction.

$$\begin{aligned}\boldsymbol{e}_r = \arg\min_{\boldsymbol{\xi}} \frac{1}{2} \sum_{i=1}^{n} \left\| \boldsymbol{p}_i^{'} - \boldsymbol{K} exp(\xi^{\wedge}) \boldsymbol{P}_i \right\|_2^2 \\ + \boldsymbol{\alpha} \| \boldsymbol{t}_g^{ij} - \boldsymbol{t}_s^{ij} \|_2 \end{aligned} \tag{1}$$

where $\boldsymbol{e}_r$ are designated to represent the cumulative optimization error. The Lie algebra representation of the transformation matrix between two frames is denoted as $\boldsymbol{\xi}$. The $\boldsymbol{\alpha}$ refers to using a certain weight for fusion, which is empirically set to 150. The matrix $\boldsymbol{K}$ denotes the intrinsic parameter matrix. And variable $\boldsymbol{P}_i$ signifies a three-dimensional point that is associated with the pixel on the current frame. Lastly, $\boldsymbol{t}_g^{ij}$ and $\boldsymbol{t}_s^{ij}$ correspond to the translation vectors derived from GPS data and the translation estimated from SLAM computations, respectively.

B. Depth Estimation

UAV surveying typically operates at altitudes of several hundred meters, rendering traditional stereo cameras ineffective in calculating depth due to baseline limitations. Therefore, we first use SLAM to obtain a pose with scale information from captured images, and then use Bouguet's correction method to perform stereo correction on the two images to maximize the common area of left and right perspectives. For the corrected image, the optical axes of the two cameras are parallel, and the poles are located at infinity. For adjacent images, the height of the corresponding pixels has been corrected to the same height, greatly accelerating the speed of binocular matching. In the case of corrected images, parallel acceleration [1] is used to perform stereo matching on high-resolution aerial image data. It forms triangulation on a set of support points that can be robustly matched to generate prior differences. These triangle matching pairs greatly reduce the disparity search space, enabling precise dense reconstruction without the need for global optimization. Further, process the generated disparity map through Depth Filtering to make the disparity more accurate.

C. Depth Filtering

Due to the initial disparity map's insufficient accuracy, we further enhanced its precision by leveraging the consistency constraint across multiple images of the same observation area, as depicted in Fig.3.

We design a First Input First Output (FIFO) queue, where each obtained depth map is placed. When the number of depth maps in the queue reaches N, the consistency check begins. The middle frame in the queue is selected as the reference frame. Its depth map is converted to 3D points and projected onto the other depth maps in the queue. The obtained 2D coordinates p_i and depth values $d_i^{'}$ are then checked to determine if the depth value d_i on the reference frame satisfies the condition $(d_i - d_i^{'})/d_i \ < \ f_{ed}$ and is observed by at least c frames. If this requirement is met, the depth value is retained; otherwise, it is removed. For all accepted points, the depth is assigned as the average of the consistent depths across the various views, thereby

[1] https://github.com/fixstars/libSGM

effectively mitigating the impact of noise. In this paper N is set to 5, c is 3 and f_{ed} to 0.01.

To reduce computational load by eliminating redundant point clouds in the queue, the depth value of the current frame projected to the corresponding point on other frames is set to 0. Subsequently, the depth map of the first frame in the queue is published.

D. Map Deformation

As SLAM iteratively update the pose within the environment, the corresponding point cloud data must be perpetually updated to maintain alignment with the evolving pose information. We refer to [28] for the transformation operation applied to the point cloud corresponding to the updated its pose. If the number of optimizations of SLAM for the bitpose of a frame reaches n, then the point cloud associated with this frame performs the transformation operation directly. For a point cloud $\boldsymbol{\rho}$ associated with a keyframe F, the position of each point in $\boldsymbol{\rho}$ is updated using $\mathbf{T}_{\boldsymbol{w},\boldsymbol{F}}{'}\mathbf{T}^{-1}_{\boldsymbol{w},\boldsymbol{F}}$ for the transformation operation, where $\mathbf{T}_{\boldsymbol{w},\boldsymbol{F}}{'}$,$\mathbf{T}^{-1}_{\boldsymbol{w},\boldsymbol{F}}$ represent the pose of the keyframes before and after optimization, respectively. After the transformation, $\mathbf{T}^{-1}_{\boldsymbol{w},\boldsymbol{F}}$ will be replaced by the optimized pose for the next deformation.

V. PATH PLANNER

The purpose of this section is to effectively improve the accuracy and completeness of reconstruction. Our methodology is structured into four phases: global coverage path planning, target surface extraction, local viewpoint creation, and trajectory optimization. Initially, the algorithm devises a global path designed to encompass the reconstruction area (Sect.V-A). Subsequently, it performs real-time analysis of the terrain's underlying structure, utilizing the point cloud data generated by the SLAM (Sect.V-B). This analysis yields a set of candidate perspectives (Sect.V-C). Finally, the trajectory information is optimized in real time with the goal of maximizing MVS performance (Sect.V-D).

Algorithm 1 Proposed Path Planning Algorithm.

```
Input: Area of interest $S_{area}$ , Sparse point cloud $\#\mathcal{N}(P_i)$,
       and Current position $P_{uav}$
$Path_{target} \leftarrow GlobalPathGen(S_{area})$
while $P_{uav} \neq Path_{target}$ do
    if $\#\mathcal{N}(P_i) \neq empty$ then
        $\{P_1, ..., P_i\}_{xy} \leftarrow UniformSampling(\#\mathcal{N}(P_i))$
        $\{P_1, ..., P_i\} \leftarrow GetSurface(\#\mathcal{N}(P_i), P_{uav})$
        $\{f^1_{pre}, ..., f^i_{pre}\} \leftarrow GetView(\#\mathcal{N}(P_i), \{P_1, ...P_i\})$
        $\chi \leftarrow GenPath(\{P_1, ...P_i\}, \{f^1_{pre}, ..., f^i_{pre}\}, P_{uav})$
        $\chi^* \leftarrow OptimizePath(\chi, f_{ref})$
    end if
    $TarckPath(\chi^*)$
    $Update(P_{uav}, \#\mathcal{N}(P_i))$
end while
```

A. Global Coverage Path Planning

The first step of all aerial photography is to design a flight path that can cover the region of interest. In this study, by using the classical Complete Coverage Path Planning (CCPP) algorithm back-and-forth to generate the path, and then the desired flight altitude of the UAV is first calculated using the GSD:

$$H_g = GSD \cdot f_{camera} \cdot w_{pix} \cdot {L_s}^{-1}, \tag{2}$$

where f_{camera} is the focal length of the camera, w_{pix} is the physical pixel value of the image, and L_s is the physical size of the sensor.

B. Target Surface Extraction

Due to the absence of an ideal flat terrain, the distance of the UAV from the ground varies with the terrain, leading to changes in the GSD. Therefore, it is necessary to adjust the UAV's flight altitude in real time to maintain a constant GSD. To obtain detailed surface features, we analyze the sparse point cloud generated by visual SLAM. For each frame of point cloud $\boldsymbol{M_p}$, we calculate the farthest distance $\boldsymbol{P_{far}}$ from the current position $\boldsymbol{P_{uav}}$, and then uniformly sample this distance along the flight path into N points, as illustrated in Fig.4. For each sampled point $\boldsymbol{P_i}(x_i, y_i, z_i)$, its 3D coordinates are:

$$\begin{aligned}
x_i &= x_{uav} + Euc_{xy}(\boldsymbol{P_{uav}}, \boldsymbol{P_{far}}) \cdot \frac{i}{N}, \\
y_i &= y_{uav} + Euc_{xy}(\boldsymbol{P_{uav}}, \boldsymbol{P_{far}}) \cdot \frac{i}{N}, \\
z_i &= \sum_{\boldsymbol{O_i} \in \#\mathcal{N}(P_i)} \frac{1}{Euc_{xy}(\boldsymbol{P_i}, \boldsymbol{O_i})} \cdot \frac{1}{w_t} \cdot D_i, \\
w_t &= \sum_{\boldsymbol{O_i} \in \#\mathcal{N}(P_i)} \frac{1}{Euc_{xy}(\boldsymbol{P_i}, \boldsymbol{O_i})}
\end{aligned}$$

where $Euc_{xy}(\cdot)$ represents the Euclidean distance between two points within the xy-plane. $\#\mathcal{N}(P_i)$ is the number of point clouds adjacent to P_i within the xy-plane. And D_i is the z-coordinate value of O_i while $O_i \in \#\mathcal{N}(P_i)$. w_t represents the weighting factor.

Since Eq.3 may select repeated point clouds to calculate different path points, even if these points are directly used to generate the trajectory, this trajectory exhibits a certain level of smoothness. However, further consideration is still necessary to maximize the performance of MVS for path optimization.

C. Local Viewpoint Generation

This section describes a method to compute an inspections that provides coverage of the target surfaces. For each sampled point $\boldsymbol{P_i}(x_i, y_i, z_i)$, this method creates a corresponding candidate viewpoints f^i_{pre}. First we calculate the normal vector of the point cloud below each sampling point:

$$\min \sum_{i=1}^{m} ((\boldsymbol{o}_i - \boldsymbol{c})^T \boldsymbol{n})^2, s.t. \|\boldsymbol{n}\|_2 = 1 \tag{3}$$

where $\boldsymbol{o}_i$ is the number of point clouds within a certain range below the sampling point, $\boldsymbol{c}$ is the average of these point clouds, and $\boldsymbol{n}$ is the normal vector.

Then we set the pose of the candidate viewpoint to the direction of the normal vector.

D. Trajectory Optimization

After determining the flight path $\chi = \{P_1, ...P_i\}$, we use heuristic information as the best viewpoint selection to maximize the MVS performance and refine the path. The following subsections describe the algorithms used to predict the quality of the MVS reconstruction and refine path.

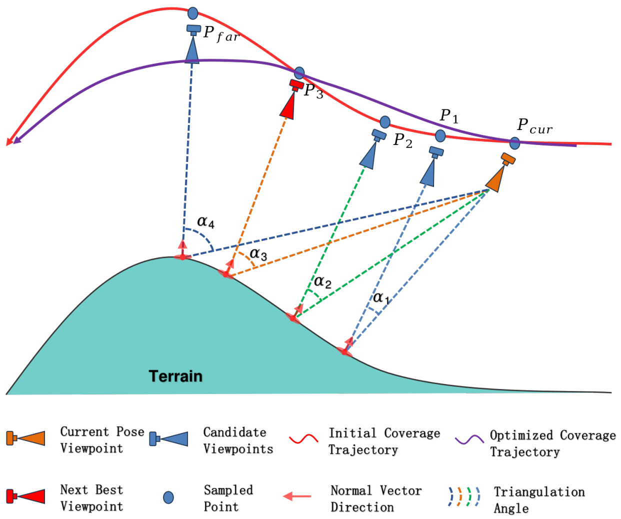

Fig. 4. Illustration of trajectory optimization. For clarity, the process is shown in 2D. The direction of each candidate viewpoints parallel to the terrain normal vector below. The α_i is the angle between the current camera pose and the candidate pose.

Triangulation: For the same point observed by both cameras, an angle that is too large or too small introduces additional error. For the same point observed by both the reference frame and the source viewpoint, the angle α between its two rays is expected to be the parallax angle. We design the score function of α as:

$$s_{angel}(\alpha) = \sum_{i=0}^{n} exp(-\frac{(\alpha_i - \alpha_0)^2}{2\sigma_{prx}^2}), \tag{4}$$

where α_0 is the desired parallax angle, heuristically determined to be $15°$ and σ_{prx} is a constant value.

Overlap: Since the pose of online MVS is determined by SLAM, it is crucial to maintain a certain overlap rate between the images. This ensures that there are as many observations as possible to maintain the accuracy of the pose. For the overlap rate, we guarantee the common view between two frames by determining whether the 3D point cloud below the pair can be projected into the camera plane, and the overlap rate score function is designed as:

$$\mathcal{NP}_{total} = \sum_{O_i \in \#\mathcal{N}(f_{ref})} Pro(f_{ref}, O_i),$$
$$s_{overlap}(f_{ref}, f_{pre}) = \sum_{K_i \in \mathcal{NP}_{total}} \frac{Pro(f_{pre}, K_i)}{\mathcal{NP}_{total}},$$

where $Pro(\cdot)$ represents a binary indicator that determines the feasibility of projecting point cloud $O_i(K_i)$ into either the reference frame (denoted as f_{ref}) or an alternative predicted frame (labeled as f_{pre}). $\mathcal{NP}_{total}$ represents the number of point clouds related to the three adjacent frames of the reference frame f_{ref}.

GSD: Reference image and source image with similar GSD can achieve more accurate stereo matching, we assume that the GSD of the pictures at the same distance from the surface is the same. The score function about the relative distance is defined as:

$$s_{gsd}(dis_{ref}, dis_{pre}) = \frac{min(dis_{ref}, dis_{pre})}{max(dis_{ref}, dis_{pre})} \tag{5}$$

Among them, dis_{ref} and dis_{pre} are the distances between the same 3D point from the reference frame and the candidate frame respectively.

We integrate these metrics into a optimization function that was used to predict the reconstruction quality of the MVS during the acquisition of images:

$$\mathcal{S}_{MVS} = \eta s_{angel} + \kappa s_{gsd} + \mu s_{overlop}, \tag{6}$$

where η, κ, μ are the weighting parameters, κ is set smaller because the trajectory generation during the flight process takes into account the terrain changes. In this paper, we set it to 10, 1, and 54, respectively.

Trajectory optimization is performed based on the above metrics with the present UAV position to obtain the optimized discrete points $\{P_1^*, ..., P_{best}, ..., P_i^*\}$.

$$\begin{aligned} \mathcal{C}_p &= \frac{EG(P_{uav}, P_i)}{Time(P_{uav}, P_{far})}, \\ \hat{\chi} &= argmax \sum_{x \in \chi} \mathcal{C}_p^{-1} \cdot \mathcal{S}_{MVS} \end{aligned} \tag{7}$$

where $Time(\cdot)$ is the time budget. We define $Time(\cdot)$ as $EG(P_{uav}, P_i) \cdot v^{-1}$, where v is the current speed of the drone and $EG(\cdot)$ is the distance between two points.

Lastly, through leveraging B-spline to obtain a final smooth trajectory χ^*.

VI. EXPERIMENTS

A. Experimental Environment

To substantiate the efficacy of our proposed algorithm, we undertook both simulation and real-world experiments. The simulation experiments were executed using Unreal Engine 4 (UE4) in tandem with the AirSim platform. The simulated environment was a vast mountainous terrain[2], within which

[2] https://www.unrealengine.com/marketplace/product/landscape-mountains

we delineated three distinct test segments, designated as Scenario 1, Scenario 2, and Scenario 3, respectively. For the real-world validation, we selected a flight location in Bailuyuan, Xi'an, which was designated as Scenario 4. This comprehensive experimental approach ensures a robust assessment of our algorithm's performance across diverse conditions.

In this study, we benchmark our proposed algorithm against two UAV aerial photography techniques: zigzag and explore-then-exploit (ETE) [8]. Due to the absence of publicly available open-source implementations for the ETE [8], so we use our implementation. For the simulation experiments, the UAV equips a gimbal-mounted camera, capturing imagery at a resolution of 1920 × 1080 pixels. Scenarios 1 and 2 encompass a 600 × 300 m^2 area, whereas Scenario 3 covered a 600 × 200 m^2 area. The real-world experiment, Scenario 4, spann a more considerable area of 1200 × 1000 m^2, with imagery captured at a higher resolution of 5472 × 3648 pixels. Across all scenarios, we establish a target GSD of 0.125 m and limit the UAV's velocity at 3 m/s to facilitate accurate pose estimation and ensure the reliability of our experimental results.

TABLE I

Time Statistics for Each Scenario

	Secnario 1 times (s)	Secnario 2 times (s)	Secnario 3 times (s)	Secnario 4 times (s)
Ours	**69**	**87**	**78**	**205**
COLMAP	8173	8392	8183	17647
Pix4D	436	603	668	7234

The data collected by each method is processed by both COLMAP and our proposed reconstruction algorithm to yield the reconstructed 3D model, respectively. For each scene, we collect a large number of images and use COLMAP to process them as true values. For the reconstruction process we all use the same camera intrinsics. All computations were performed on a PC equips with an NVIDIA RTX 3080 Ti GPU and an Intel Core i9-12900KF CPU with 32 GB RAM.

To evaluate the processing efficiency of the proposed reconstruction algorithms, we conducted a comparative analysis by isolating the algorithm, as detailed in Table.I. And we compare the performance of the algorithms by two metrics, the GSD and the F-score. Fig.6 and Table.II reflect the results of the comparison. In comparison to other methods, our method stands out by offering not only a faster reconstruction process but also by delivering superior model quality. When it comes to the fidelity of details, our method is on par with COLMAP. Regarding the assessment of reconstruction quality, we refer to the evaluation procedures and metrics in [34]. First, the reconstructed point cloud is registered with the ground truth point. Post-registration, two point clouds are resampled onto a voxel grid with a voxel dimension of 0.5 m. Recall is an indicator of the completeness of the reconstructed model, while precision reflects how close the model to the real-world environment.

TABLE II

Results Obtained Using Our 3D Reconstruction Algorithm Under Various Image Capture Methods

	Methods	Precision	Recall	F-score	Comp. 0.3m	Comp. 0.5m
Secnario 1	Zigzag	0.8693	0.8682	0.8687	0.2166	0.6434
	ETE	0.8823	0.9331	0.9069	0.2099	0.6755
	Ours	**0.9061**	**0.9743**	**0.9389**	**0.2844**	**0.7235**
Secnario 2	Zigzag	0.9233	0.8324	0.8755	0.3012	0.7166
	ETE	0.9533	0.9213	0.9370	**0.3348**	0.7356
	Ours	**0.9681**	**0.9423**	**0.9550**	0.3152	**0.7568**
Secnario 3	Zigzag	0.9233	0.8324	0.8755	0.4304	0.8446
	ETE	0.9588	**0.9678**	**0.9632**	0.4506	**0.8561**
	Ours	**0.9681**	0.9423	0.9550	**0.4653**	0.8343
Secnario 4	Zigzag	0.8564	**0.9163**	0.8853	0.2013	0.6813
	ETE	0.8664	0.8963	0.8810	**0.2963**	0.6931
	Ours	**0.8981**	0.8815	**0.8897**	0.2306	**0.7065**

B. Evaluation of 3D Reconstruction

Table.I list a comparative analysis of the runtime statistics of our reconstruction algorithm against COLMAP and Pix4D. Additionally, Fig.6 provides partial reconstruction details of the experiment and a visual representation of the flight trajectory, where ETE is a trajectory generated based on prior information. Our algorithm significantly outperforms COLMAP in terms of processing time, mainly because COLMAP relies on random initialization for image matching, while our approach can obtain matched images through SLAM. Compared with zigzag and ETE, our algorithm only requires a single flight. Table.II demonstrates the efficacy of our reconstruction algorithm when applied to three distinct image acquisition methods. The online MVS algorithm balances the trade-off between accuracy and speed, guaranteeing both accurate and swift generation of 3D models. The error range of most reconstruction points is about 0.3 m, as shown in Fig.5, especially when flying at altitudes exceeding 120 m.

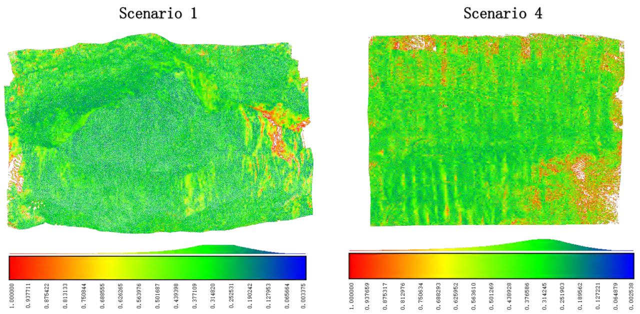

Fig. 5. Error distribution and statistics between our method and COLMAP. The error distribution is concentrated around 0.3m.

Our acquisition technique excels by dynamically adjusting the AGL and the viewpoint in real-time, adeptly responding

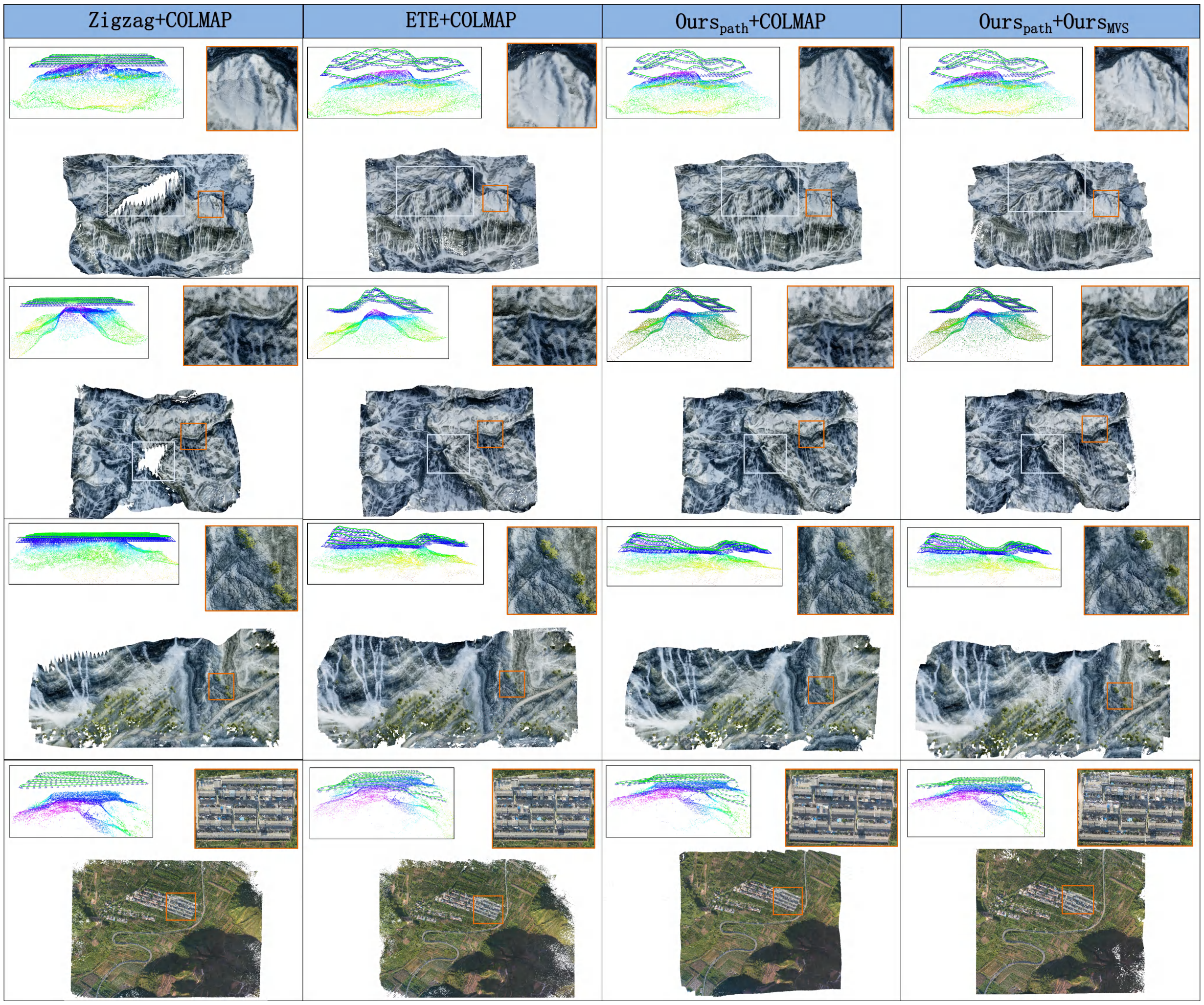

Fig. 6. Reconstructed 3-D models and details with trajectories taken by the UAV. The subscripts path and MVS in the figure represent our path planning method and the online MVS algorithm respectively. The upper left corner shows the flight trajectory, where ETE is the trajectory generated by prior information.

to terrain variations throughout the flight. This approach significantly enhances the quality of the reconstruction, regardless of whether the images are processed with COLMAP or our online MVS system. This real-time adaptability ensures a higher fidelity in the 3D models, making our method a robust solution for complex and variable terrain environments.

C. Evaluation of GSD

In aerial surveying, GSD serves as a crucial indicator, ensuring the consistency of image data. As shown in Fig.7, the GSD of the acquired images by our method is always distributed around the expected value. In scenarios characterized by significant altitude fluctuations, such as Scenario 1 and Scenario 2, our approach delivers a superior GSD distribution when compared to the zigzag and ETE method. This advantage is attributed to our real-time integration of terrain elevation changes into the viewpoint adjustments. Furthermore, in environments such as Scenarios 3 and 4, which have small elevation variations, our algorithm has good GSD distributions compared to ETE, but our algorithm

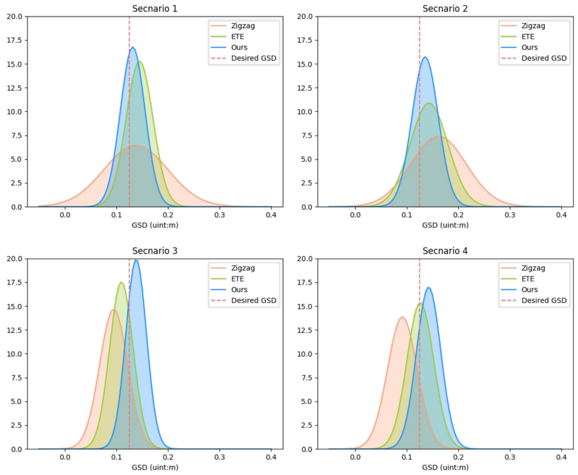

Fig. 7. GSD distribution map, calculated from flight altitude and camera focal length. The GSD data obtained by our method is more consistent.

does not require a priori information.

The consistent maintenance of GSD across diverse terrains allows the MVS algorithm to not only expedite the processing time but also to minimize the loss of detail in the final reconstructions. This attribute is pivotal for applications requiring high-precision 3D models, as it ensures that the reconstructed data remains both accurate and rich in detail.

VII. CONCLUSIONS

In this paper, we introduce an online Multi-View Stereo (MVS) method for optimizing view path planning in large-scale scenes. Our algorithm dynamically adjusts viewpoints in real-time based on terrain feedback to enhance 3D reconstruction effectiveness. By integrating a specialized heuristic function and SLAM-derived data, the algorithm efficiently identifies optimal view paths, enabling smooth trajectory creation to guide UAVs in capturing and reconstructing regions of interest. Experimental results demonstrate that the algorithm can be effectively executed, maintaining superior model completeness without compromising accuracy. This balance of speed and quality establishes our approach as a robust solution for aerial photography and 3D modeling applications.

The limitation of our method is that the entire system relies too much on the SLAM system, which may fail in weak texture areas. In the future, we plan to further optimize the architecture to achieve better results.

REFERENCES

[1] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 501–518.

[2] R. M. Stereopsis, "Accurate, dense, and robust multiview stereopsis," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 32, no. 8, 2010.

[3] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3260–3269.

[4] R. Huang, D. Zou, R. Vaughan, and P. Tan, "Active image-based modeling with a toy drone," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6124–6131.

[5] H. Zhang, Y. Yao, K. Xie, C.-W. Fu, H. Zhang, and H. Huang, "Continuous aerial path planning for 3d urban scene reconstruction." *ACM Trans. Graph.*, vol. 40, no. 6, pp. 225–1, 2021.

[6] B. Hepp, M. Nießner, and O. Hilliges, "Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 1, pp. 1–17, 2018.

[7] Q. Kuang, J. Wu, J. Pan, and B. Zhou, "Real-time uav path planning for autonomous urban scene reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1156–1162.

[8] H. Zhao, B. Zhang, W. Hu, J. Liu, D. Li, Y. Liu, H. Yang, J. Pan, and L. Xu, "Adaptable flight line planning for airborne photogrammetry using dem," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6206–6218, 2021.

[9] Bentley. (2018) Contextcapture. [Online]. Available: https://www.bentley.com/ software/contextcapture

[10] Pix4d. (2022) pix4dmapper. [Online]. Available: https://www.pix4d.com.cn/pix4dmapper

[11] Agisoft. (2022) Agisoft metashape:. [Online]. Available: https://www.agisoft.com

[12] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[14] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[15] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," 2020. [Online]. Available: https://cdcseacave.github.io/openMVS

[16] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2495–2504.

[17] D. Yu, S. Ji, J. Liu, and S. Wei, "Automatic 3d building reconstruction from multi-view aerial images with deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 155–170, 2021.

[18] J. Zhang, Y. Yao, S. Li, Z. Luo, and T. Fang, "Visibility-aware multi-view stereo network," *arXiv preprint arXiv:2008.07928*, 2020.

[19] Q. Xu, W. Su, Y. Qi, W. Tao, and M. Pollefeys, "Learning inverse depth regression for pixelwise visibility-aware multi-view stereo networks," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 2040–2059, 2022.

[20] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller, and Y. Aloimonos, "Gapflyt: Active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2799–2806, 2018.

[21] Pix4d. (2017) pix4dcapture. [Online]. Available: http://pix4d.com/product/pix4dcapture

[22] 3dr Robotics. (2017) 3dr site scan. [Online]. Available: https://3dr.com/

[23] C. Peng and V. Isler, "Adaptive view planning for aerial 3d reconstruction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2981–2987.

[24] C. Feng, H. Li, F. Gao, B. Zhou, and S. Shen, "Predrecon: A prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction," *arXiv preprint arXiv:2302.04488*, 2023.

[25] Y. Liu, R. Cui, K. Xie, M. Gong, and H. Huang, "Aerial path planning for online real-time exploration and offline high-quality reconstruction of large-scale urban scenes," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–16, 2021.

[26] X. Zhou, K. Xie, K. Huang, Y. Liu, Y. Zhou, M. Gong, and H. Huang, "Offsite aerial path planning for efficient urban scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–16, 2020.

[27] S. Song, D. Kim, and S. Jo, "Active 3d modeling via online multi-view stereo," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5284–5291.

[28] S. Song, D. Kim, and S. Choi, "View path planning via online multiview stereo for 3-d modeling of large-scale structures," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 372–390, 2021.

[29] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2609–2616.

[30] Y. Zhao, S. Xu, S. Bu, H. Jiang, and P. Han, "Gslam: A general slam framework and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1110–1120.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[32] C. Wu, "Siftgpu: A gpu implementation of scale invariant feature transform (sift)(2007)," *URL http://cs. unc. edu/˜ ccwu/siftgpu*, 2011.

[33] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[34] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.