forests

MDPI

Article

A Novel Approach for Simultaneous Localization and Dense Mapping Based on Binocular Vision in Forest Ecological Environment

Lina Liu [1,2], Yaqiu Liu [1,2,*], Yunlei Lv [1,3] and Xiang Li [1,3]

[1] College of Computer and Control Engineering, Northeast Forestry University, Harbin 150040, China; lln@nefu.edu.cn (L.L.); yunleilv@nefu.edu.cn (Y.L.); nefu_lx@nefu.edu.cn (X.L.)
[2] Key Laboratory of Sustainable Management of Forest Ecosystems, Ministry of Education, Northeast Forestry University, Harbin 150040, China
[3] National and Local Joint Engineering Laboratory for Ecological Utilization of Biological Resources, Northeast Forestry University, Harbin 150040, China
* Correspondence: yaqiuLiu@nefu.edu.cn

Abstract: The three-dimensional reconstruction of forest ecological environment by low-altitude remote sensing photography from Unmanned Aerial Vehicles (UAVs) provides a powerful basis for the fine surveying of forest resources and forest management. A stereo vision system, D-SLAM, is proposed to realize simultaneous localization and dense mapping for UAVs in complex forest ecological environments. The system takes binocular images as input and 3D dense maps as target outputs, while the 3D sparse maps and the camera poses can be obtained. The tracking thread utilizes temporal clue to match sparse map points for zero-drift localization. The relative motion amount and data association between frames are used as constraints for new keyframes selection, and the binocular image spatial clue compensation strategy is proposed to increase the robustness of the algorithm tracking. The dense mapping thread uses Linear Attention Network (LANet) to predict reliable disparity maps in ill-posed regions, which are transformed to depth maps for constructing dense point cloud maps. Evaluations of three datasets, EuRoC, KITTI and Forest, show that the proposed system can run at 30 ordinary frames and 3 keyframes per second with Forest, with a high localization accuracy of several centimeters for Root Mean Squared Absolute Trajectory Error (RMS ATE) on EuRoC and a Relative Root Mean Squared Error (RMSE) with two average values of 0.64 and 0.2 for t_{rel} and R_{rel} with KITTI, outperforming most mainstream models in terms of tracking accuracy and robustness. Moreover, the advantage of dense mapping compensates for the shortcomings of sparse mapping in most Smultaneous Localization and Mapping (SLAM) systems and the proposed system meets the requirements of real-time localization and dense mapping in the complex ecological environment of forests.

Keywords: binocular vision SLAM; pose estimation; dense mapping; keyframe selection; spatial clue compensation; forest 3D reconstruction

Citation: Liu, L.; Liu, Y.; Lv, Y.; Li, X. A Novel Approach for Simultaneous Localization and Dense Mapping Based on Binocular Vision in Forest Ecological Environment. *Forests* **2024**, *15*, 147. https://doi.org/10.3390/f15010147

Academic Editor: Giorgos Mallinis

Received: 22 November 2023
Revised: 2 January 2024
Accepted: 6 January 2024
Published: 10 January 2024

1. Introduction

With the advantage of quickly and accurately obtaining three-dimensional spatial information, Light Detection and Ranging (LiDAR) is widely used in forest resource surveys. Solares-Canal et al. [1] proposed a methodology based on Machine Learning (ML) techniques to automatically detect the positions of and dasometric information about individual Eucalyptus trees from a point cloud acquired with a portable LiDAR system. Gharineiat et al. [2] summarized the methods of feature extraction and classification using ML techniques for laser point cloud data, which have good applications in scene segmentation, vegetation detection, and tree species classification.

In recent years, with the rapid development of UAVs technology, UAVs carrying LiDAR or visual sensors have greatly assisted in forest ecological exploration and forest management [3], and the forest information they captured can provide an essential basis for the three-dimensional reconstruction of forest ecological models. UAVs do not have a priori information about the relevant environment and their own position before executing the task, and they know nothing about the environment they are in, so there is no way to talk about UAV path planning and autonomous navigation, and how to solve the navigation problem of UAVs in the unknown environment of the forest is a difficult problem for forestry ecological exploration. Simultaneous Localization and Mapping (SLAM) [4] is one of the key algorithms for realizing fully autonomous navigation and real intelligence of mobile robots by means of sensor-equipped motion carriers that can move in unknown environments, senses and build environment maps, and estimate their own positions in the constructed maps at the same time, which empowers the robots to autonomously localize themselves and build real-time maps in unknown environments.

2. Related Work

Visual SLAM [5–7] has the advantages of being low cost, easy to use, and rich in information compared with LiDAR SLAM [8–10], so there is a huge potential for the development of visual SLAM. The mainstream methods for visual SLAM are the direct method and the feature point method. Dense Tracking and Mapping (DTAM) [11] based on the direct method uses the image pixels to construct a cost function and describes the depth using the inverse depth, constructing a 3D dense map in a global optimization. Large-scale direct monocular SLAM (LSD-SLAM) [12] directly matches image luminosity, uses a probabilistic model to represent semi-dense depth maps, and generates maps with global consistency. Direct sparse odometry (DSO) [13] is an improved version of LSD-SLAM that combines the direct method with sparse synchronization optimization, which can be applied in the case where RAM and CPU resources are lacking. Large-scale direct sparse visual odometry with stereo cameras (Stereo DSO) [14] integrates the constraints of the fixed binoculars into the Bundle Adjustment (BA) of the multi-view binoculars, which solves the scale drift problem while mitigating the optical flow sensitivity and the roll-up shutter effect of the conventional direct method.

The direct methods track directly on the image grayscale information, which have the advantages of fast speed and good real-time performance, however, they are based on the assumption of grayscale invariance and are limited to the narrow baseline motion. Feature-based methods use an indirect representation of the image, usually in the form of point features tracked along consecutive frames, recover poses of the camera by minimizing the projection error, which are more robust, and currently dominate the field of vision SLAM. Real-time single camera SLAM (MonoSLAM) [15] is the first monocular SLAM system, which achieves real-time drift-free performance from the structure to the motion model, but the feature stability is greatly affected by motion. Parallel Tracking and Mapping for small AR workspaces (PTAM) [16] is the first visual SLAM to be solved by an optimization method, which pioneers a keyframe mechanism and a dual-threaded parallel processing task to simultaneously handle tracking and mapping. A series of Oriented FAST and Rotated BRIEF (ORB) SLAM algorithms built by the SLAM group at the University of Zaragoza, Spain, is currently the most popular feature point method solution. A versatile and Accurate Monocular SLAM System (ORB-SLAM) [17] was first proposed in 2015, which is based on PTAM and uses ORB descriptors, and it only supports monocular cameras, thus it suffers from the scale uncertainty problem. An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras (ORB-SLAM2) [18] improves the efficiency and robustness of ORB feature extraction and descriptor matching, and it adds functions such as closed-loop detection and pose map optimization to enable it to cope with more complex environments and faster pose change. However, the keyframe selection conditions are more lenient leading to the high redundancy between frames in uniform linear motion, which will bring a higher cost of maintenance and deletion of keyframes in

the later stage, thus affecting the performance of the system. Cameras with excessive cornering, fast speed changes, severe scene occlusion, and large changes in lighting can cause tracking to be lost if keyframes are not inserted in time. In addition, ORB-SLAM2 can only construct sparse maps, which does not enable tasks such as autonomous navigation and obstacle avoidance for robots.

Monocular SLAM is based on spatial geometric relationships and suffer from the disadvantage of scale drift. SLAM based on RGB Depth Camera (RGBD SLAM) [19,20] is susceptible to interference from varying light intensities, making it unsuitable for outdoor scenes. Additionally, the high cost of these cameras hinders their widespread adoption in the industry. Conventional vision SLAM based on geometric transformations has poor robustness when lighting changes, fast carrier motion, and low texture grayscales are not obvious, and are poorly applied in the scenes, as well as having drawbacks such as large amounts of calculations and large cumulative error. Therefore, it is generally used in indoor small target scenes and its application in outdoor complex scenes is limited. Deep learning methods [21–24] are not constrained by the above environmental conditions, and are able to quickly estimate the more accurate disparity in outdoor complex environments, enhancing the robustness of pose estimation and 3D scenes reconstruction. In this work, combine with deep learning, a stereo vision system, D-SLAM, is proposed to realize the simultaneous localization and dense mapping for UAVs in complex forest ecological environment. The main work is as follows.

(1) Using the temporal clue of binocular images as the main clue, the six Degrees of Freedom (6-DoF) rigid body pose of the UAV is estimated by utilizing the minimized visual feature reprojection error.
(2) Using the spatial clue of binocular images as an auxiliary clue, a binocular spatial compensation strategy is proposed to increase the robustness of the algorithm's tracking when the camera corner is too large.
(3) Taking the relative motion amount and data association between frames as the important conditions for filtering keyframes, the keyframe filtering strategy is improved to enhance the system's localization and mapping accuracy as well as running speed.
(4) By increasing the 3D dense map construction thread, using the LANet network to predict the disparity map of the keyframes, and combining the poses of the keyframes to generate a dense point cloud, a dense map of the complex ecological environment of the forest is constructed by utilizing techniques such as point clouds registration, point clouds fusion and point clouds filtering.

3. Study Area and Data

The experimental forest farm of Northeast Forestry University was selected as the study area, using a ZED2 binary camera (Stereolabs, San Francisco, CA, USA) to collect image data and video bag data to create a Forest dataset for the training and testing of generating disparity maps and dense mapping, respectively.

3.1. Study Area

The study area is located at 126°37′ E, 45°43′ N in the Harbin Experimental Forestry Farm of Northeast Forestry Universit, at an altitude of 136–140 m, with a mesothermal continental monsoon climate. The forest is covered with 18 species of plantation forests, including *Larix gmelinii* Rupr., *Quercus mongolica* Fisch. ex Ledeb., *Betula platyphylla* Suk., and *Fraxinus mandshurica* Rupr. The structural types in the sample plots include the tree layer and herb layer, and the canopy density exceeds 0.70. The experimental environment is characterized by the easy lock–lose of Global Navigation Satellite System (GNSS satellites), large sample plot coverage, and obvious topographic relief, which is representative of the field forest exploration tasks.

3.2. Forest Dataset

Five types of forest vegetation, *Larix gmelinii*, *Pinus sylvestris* var. mongolica Litv., *Pinus tabulaeformis* Carr., *Fraxinus mandschurica*, and *Betula platyphylla*, were selected for the experiments, and a ZED2 binocular camera was used with a baseline length of 20 cm, while RGB binocular image pairs and their corresponding disparity maps were collected with an original pixel resolution of 1280 × 720, which was cropped to a resolution of 1240 × 426, totaling 5000 pairs to produce the Forest dataset used for the Neural network model training and testing. Out of this, 80% are used as training data, 10% as the validation set and 10% as the test set. The details are shown in Table 1.

Table 1. Forest dataset details.

Variety	Training Set	Validation Set	Test Set	Total	Resolution	Sparse /Dense	Synthetic /Real
Larix gmelinii	960	120	120	1200	1240 × 426	dense	real
Pinus sylvestris	1200	150	150	1500	1240 × 426	dense	real
Pinus tabulaeformis	800	50	50	500	1240 × 426	dense	real
Fraxinus mandschurica	640	80	80	800	1240 × 426	dense	real
Betula platyphylla	800	100	100	1000	1240 × 426	dense	real

The Forest dataset includes not only the disparity map dataset (Figure 1) used for training and testing LANet networks, but also the bag video dataset (Figure 2) and binocular image dataset (Figures 3 and 4) used for testing the D-SLAM system, including three resolutions: High Definition (HD) 1080:1920 × 1080, HD720:1280 × 720, and Video Graphic Array (VGA): 672 × 376, as shown below.

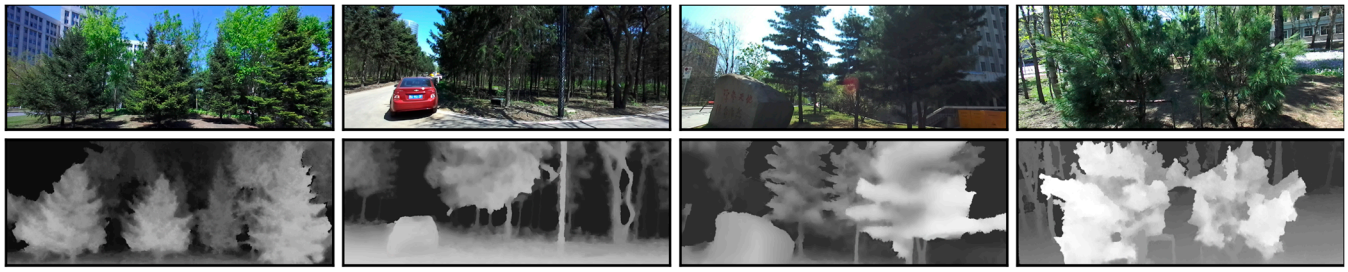

Figure 1. Disparity maps in Forest dataset.

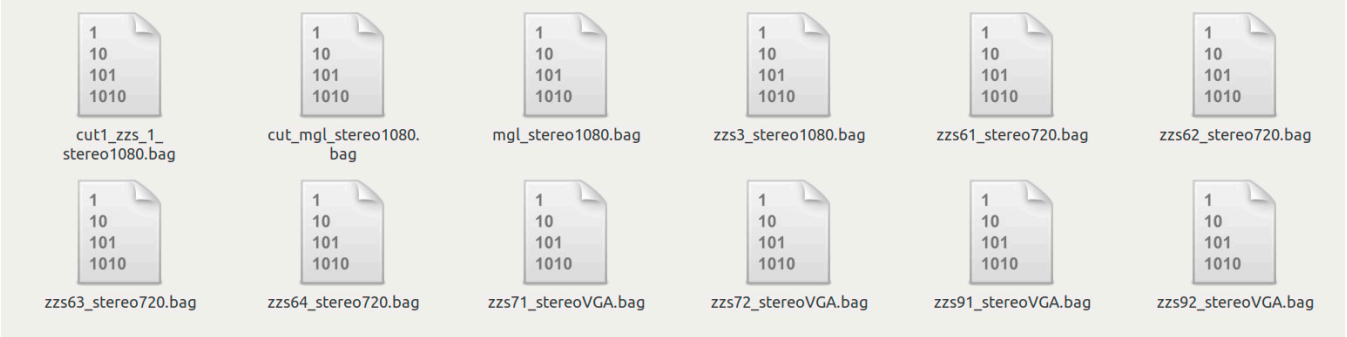

Figure 2. Bag in Forest dataset.

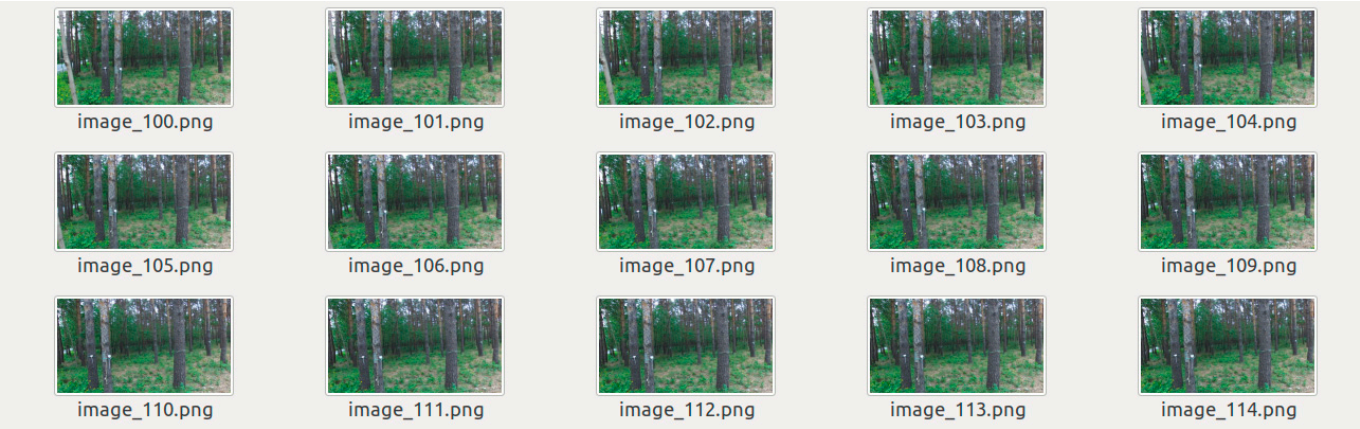

Figure 3. Left images in bag.

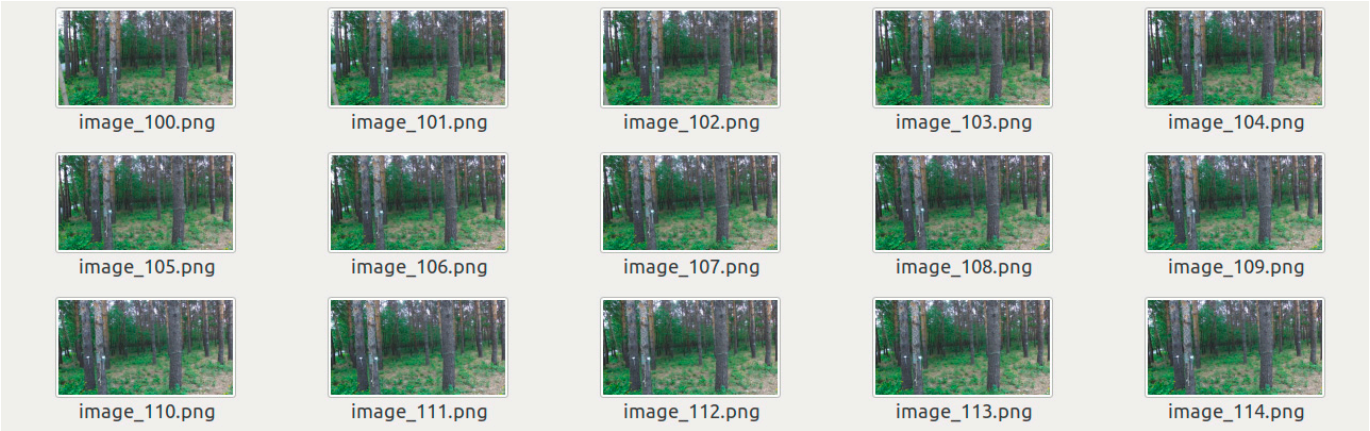

Figure 4. Right images in bag.

4. Methods

The structure of D-SLAM system is shown in Figure 5. The tracking thread searches for feature points to match with the local map in each frame and uses motion-only BA to minimize the reprojection error to optimize the pose of the current frame to realize the camera's location and tracking in each frame, and at the same time determines whether the current frame is a keyframe or not according to the conditions. The local mapping thread receives the keyframes from the tracking thread, eliminates redundant map points, generates new map points, optimizes the local map points and the poses of keyframes, and deletes redundant keyframes. The dense mapping thread receives the disparity map generated by the LANet, combines it with the pose of the keyframe to obtain a 3D point cloud, and then generates the dense point cloud map through point clouds registration, point clouds fusion and point clouds filtering. The loop closing thread corrects the cumulative drift through pose-graph optimization and starts the full BA thread for the BA optimization of all map points and keyframes.

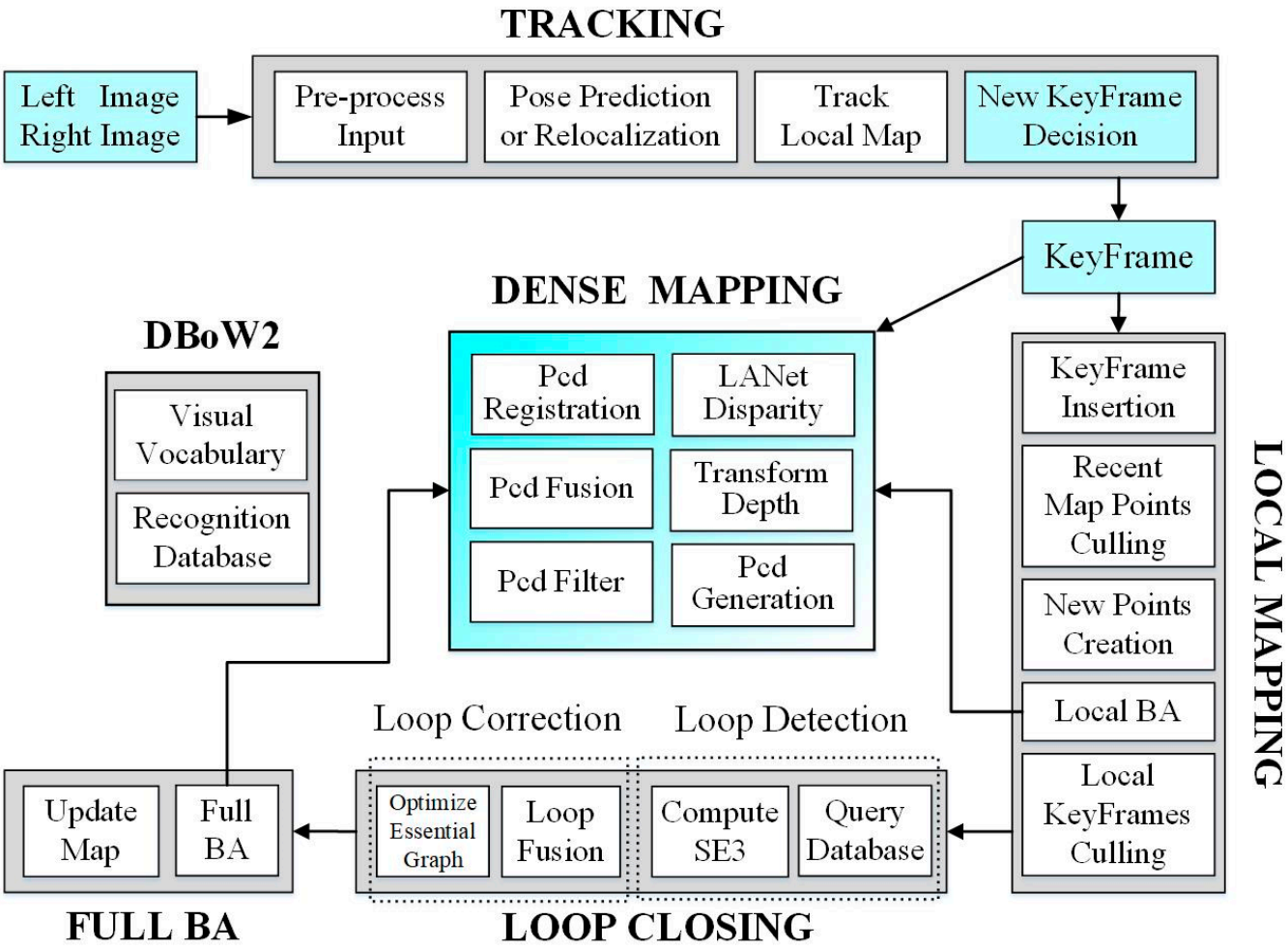

Figure 5. The D-SLAM system consists of four main parallel threads, tracking, local mapping, dense mapping, and loop closing, where the acronyms are defined as follows: Preprocessing (Pre-process), Llocal Bundle Adjustment (Local BA), Full Bundle Adjustment (Full BA), Special Euclidean group (SE3), Point cloud (Pcd), Linear Attention Network (LANet), and bags of binary Words for fast place recognition in image sequences (DboW2).

4.1. Tracking

(1) Binocular initialization

Monocular initialization requires two or more image frames with both rotation and translation necessary for successful initialization. Binocular initialization is conducted in the first frame and absolute scale information is obtained. The binocular camera performs stereo matching through the left and right images of the first frame, using the principle of triangulation to obtain the depth information of the feature points, according to the current frame of the pose to obtain the world coordinates, so the binocular camera can generate 3D map points and create an initial map in the first frame, and then tracking is conducted directly in the next frame.

(2) Pose estimation

In binocular mode, two consecutive frames of the left image in the temporal dimension perform feature matching to find the corresponding data-associated feature points, whereby the left image frames are consecutive in the temporal clue which serves as the main clue for the pose estimation. The continuity of the right image frames in the spatial dimension is used as an auxiliary clue to search for feature matching points on the right spatial clue frame corresponding to the left image frame, and if a match can be made and

the effective depth value of the feature point can be obtained, the 3D-3D Iterative Closest Point (ICP) method can be used to estimate the pose of the current frame.

The two consecutive frames of the left image in the temporal dimension find the corresponding data-associated feature points through the feature matching, assuming that the 3D map points corresponding to these two sets of feature points are $p = \{p_1, p_2, \ldots p_n\}$ and $p' = \{{p'}_1, {p'}_2, \ldots {p'}_n\}$, where $p, p' \in \mathbb{R}^3$. Because each point in p and p' has already been associated with the data through subscripts one by one, ${p'}_i = R \cdot p_i + t$ is satisfied between p and p' in the ideal case, where $R \in SE(3)$ and $t \in \mathbb{R}^3$, but in fact ${p'}_i \neq R \cdot p_i + t$ due to the presence of noise. At this time, (R, t) can be solved by constructing the least squares problem [25] expressed as follows.

$$\begin{aligned} \underset{R,t}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| {p'}_i - (R \cdot p_i + t) \right\|^2 &= \underset{R,t}{\operatorname{argmin}} \sum_{i=1}^{n} \left\{ \left\| ({p'}_i - \overline{p}') - R \cdot (p_i - \overline{p}) \right\|^2 + \left\| \overline{p}' - R \cdot \overline{p} - t \right\|^2 \right\} \\ &= \underset{R,t}{\operatorname{argmin}} \sum_{i=1}^{n} \left\{ \left\| {q'}_i - R \cdot q_i \right\|^2 + \left\| \overline{p}' - R \cdot \overline{p} - t \right\|^2 \right\} \end{aligned} \tag{1}$$

where the point clouds p and p' are moved towards the center, let $q_i = p_i - \overline{p}$, ${q'}_i = {p'}_i - \overline{p}'$, $\overline{p} = \frac{1}{n} \sum_{i=1}^{n} p_i$, and $\overline{p}' = \frac{1}{n} \sum_{i=1}^{n} {p'}_i$.

In Equation (1), the first additive term is expanded and simplified as follows.

$$\begin{aligned} &\underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| {q'}_i - R \cdot q_i \right\|^2 = \underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} \left({q'}_i^T {q'}_i - 2 {q'}_i^T R q_i + q_i^T R^T R q_i \right) \\ &\qquad\qquad \Updownarrow \\ &\underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} -{q'}_i^T R q_i = \underset{R}{\operatorname{argmin}} \left[-tr(R \sum_{i=1}^{n} {q'}_i^T q_i) \right] \end{aligned} \tag{2}$$

Let $M = \sum_{i=1}^{n} {q'}_i^T q_i$, Singular Value Decomposition (SVD) decomposition is utilized to obtain $SVD(M) = USV^T$ and then$R = UV^T$, and by substituting the resulting R into the second additive term $\underset{t}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \overline{p}' - R \cdot \overline{p} - t \right\|^2$, it will be easy to obtain $t = \overline{p}' - R \cdot \overline{p}$.

If there is no right feature point on the spatial cue that can match on the left feature point, or the effective depth value of the left feature point can not be obtained, at this time, it can only be triangulated by the multi-frame view. Based on the known 3D positions of the feature points in the local sliding window, and their 2D observations in the image, the pose of the current frame can be solved by using the 3D-2D Perspective-n-Point (PnP) method.

The coordinates of n 3D spatial points and their 2D point observations are known, n known map points p_i^w (i = 1, 2, ..., n) are selected as reference points from the world coordinate system, and 4 known map points c_j^w (j = 1, 2, 3, 4) are selected as control points, which are associated with the reference points by means of a weighted sum expressed as follows.

$$p_i^w = \sum_{j=1}^{4} \alpha_{ij} c_j^w \tag{3}$$

where $\sum_{j=1}^{4} \alpha_{ij} = 1$.

p_i^c and c_j^c are map points and control points under the camera coordinate system, and because only the values of the coordinates are taken differently, the relative spatial positions between the points have not changed, so the relationship of the weighted sum also holds, then there is

$$p_i^c = \sum_{j=1}^{4} \alpha_{ij} c_j^c \tag{4}$$

In the camera coordinate system, the reference point p_i^c with its corresponding pixel point $p_i^u(u_i, v_i)$ can be described by the projection equation [26] as follows

$$w_i \begin{bmatrix} p_i^u \\ 1 \end{bmatrix} = K \cdot p_i^c = K \cdot \sum_{j=1}^{4} \alpha_{ij} c_j^c \tag{5}$$

there is

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} \Leftrightarrow \begin{cases} \sum\limits_{j=1}^{4} \alpha_{ij} f_u x_j^c + \alpha_{ij}(u_c - u_i) z_j^c = 0 \\ \sum\limits_{j=1}^{4} \alpha_{ij} f_v y_j^c + \alpha_{ij}(v_c - v_i) z_j^c = 0 \end{cases} \Leftrightarrow A_{2\times 12} \cdot h_{12\times 1} = 0 \tag{6}$$

The coefficient matrix $A_{2\times 12}$ is constructed from the weighting coefficients of the reference points, the pixel coordinates and the camera internal parameters, and the vector $h_{12\times 1}$ is constructed from the 12 coordinate values x_1^c、y_1^c、z_1^c, x_2^c、y_2^c、z_2^c, x_3^c、y_3^c、z_3^c, and x_3^c、y_3^c、z_3^c of the 4 control points in the camera coordinate system. From this, a linear equation $A_{2\times 12} \cdot h_{12\times 1} = 0$ can be constructed by projecting a reference point p_i^c to a camera pixel point u_i. Then n reference points p_i^c projected to pixel points u_i can construct the linear equations $A_{2n\times 12} \cdot h_{12\times 1} = 0$, followed by solving the equation for the vector h. The solution process is based on least squares and SVD methods, solving the equations to obtain the coordinate values of the four control points in the camera coordinate system, combined with the known coordinate values of the four control points in the world coordinate system, and then utilizing the ICP algorithm to find the transformation relationship (R, t) between the four control points in the two coordinate systems.

The camera poses solved by the above methods are subject to errors due to noise, computation and other factors, and BA optimization can be used to further optimize the poses and improve the accuracy. Construct a nonlinear least squares problem on reprojection error: 3D points are projected to 2D points which are combined with the observed 2D points to construct the reprojection error equation, and the optimal solution is obtained by iterations using the Gaussian Newton method. In the world coordinate system, the map point p_i^w is converted to camera coordinates $p_i^c = T \cdot p_i^w$ by the transfer matrix $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$, and the pixel coordinates are obtained by projecting the camera coordinates using the camera model

$$w_i \begin{bmatrix} \hat{p}_i^u \\ 1 \end{bmatrix} = K \cdot p_i^c \tag{7}$$

Establish the relationship from world coordinates to pixel coordinates: $w_i \hat{p}_i^u = K \cdot p_i^c = K \cdot T \cdot p_i^w$, i.e., $\hat{p}_i^u = \frac{1}{w_i} K \cdot T \cdot p_i^w$, obtain the error term by subtracting the value $p_i^u(u_i, v_i)$ from the observed coordinates of the 2D point, construct the least squares problem by using the sum of all the error terms, minimize the reprojection error and the camera pose is solved by using Gauss-Newton optimization algorithm [25] as follows.

$$T = \underset{T}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \left\| p_i^u - \frac{1}{w_i} K \cdot T \cdot p_i^w \right\|_2^2 \tag{8}$$

(3) Keyframes selection

ORB-SLAM2 obtains keyframes under more relaxed conditions to ensure that it can "keep up" with the tracking thread in the early stage, however, the quality of the keyframes is not taken into consideration. For instance, the high redundancy between frames when the camera in uniform linear motion will lead to the high cost of maintaining and deleting the keyframes in the later stage, which would affect the performance of the system. The untimely insertion of keyframes can easily lead to loss of tracking when turning or chang-

ing speeds quickly. Furthermore, tracking loss also occurs easily when there is serious occlusion and large changes in lighting.

According to the actual complex ecological environment of the forest, the keyframe selection strategy is designed to avoid the introduction of too much information redundancy due to the excessive image overlap. At the same time, the image overlap should not be too small to ensure that there are some covisibility feature points to avoid tracking loss. Under the constraints of the covisibility graph, the quality of keyframe tracking is guaranteed while achieving the goal of having both constraints and less information redundancy between keyframes and other keyframes in the local map. Based on the above requirements, the amount of relative motion between frames (rotation angles and translation changes) and data association (number of matched feature points) are considered as the important basis for the selection of keyframes.

The amount of relative motion $Tran\ (R,t)$ between the current frame and the previous keyframe is a function of the pose (R,t), and it is defined as follows.

$$Tran\ (R,t) = (1-\alpha)\ ||t|| + \alpha min(2\pi - ||R||\ ,\ ||R||) \tag{9}$$

where $\alpha = (\frac{\tan\omega}{\tan\omega+1})^{\frac{1}{4}}$, and $\omega = min(2\pi - ||R||\ ,\ ||R||)$.

Because $R \in SE(3)\ ,\ t\ \in \mathbb{R}^3$, $Tran$ represent the relative motion between frames, their Euclidean spatial distances are taken as the amount of translation and rotation changes between frames, respectively. $\alpha \in [0,1]$ is a motion transformation factor between frames, the size of which increases exponentially with the increase in the angle of camera rotation, and it has a corner of the amplification function and at the same time has a translation suppression function. When α is large, the suppression $1-\alpha$ of the translation can be brought close to 0, where the amount of relative motion between the frames depends mainly on α. The value of α and the specific expression of $Tran$ are determined by the range of $\omega = min(2\pi - ||R||\ ,\ ||R||)$. $Tran\ (R,t)$ can be expressed as follows,

$$Tran\ (R,t) = \begin{cases} (1-\alpha)||t|| + \alpha min(2\pi - ||R||, ||R||)\ ,\ \omega \in [0, \frac{\pi}{2})\ ,\ \alpha = (\frac{\tan\omega}{\tan\omega+1})^{\frac{1}{4}} \\ \alpha min(\ 2\pi - ||R||, ||R||\ ) \qquad\qquad\quad ,\ \omega \in [\frac{\pi}{2}, \pi]\ ,\ \alpha = (\frac{|\tan\omega|}{|\tan\omega|+1})^{\frac{1}{4}} \end{cases} \tag{10}$$

When the angle of camera rotation $\omega \in [\frac{\pi}{2}, \pi]$, the camera almost loses the perspective, at which time the feature points cannot be matched on the temporal clue causing the camera tracking to fail. In order to increase the robustness of the system in tracking, the spatial clue compensation strategy is adopted: the previous frame of the right image of the spatial clue is used as the clue connection, and it is inserted to compensate for the lost field of view on the temporal clue, continuing the tracking, as shown in Figure 6.

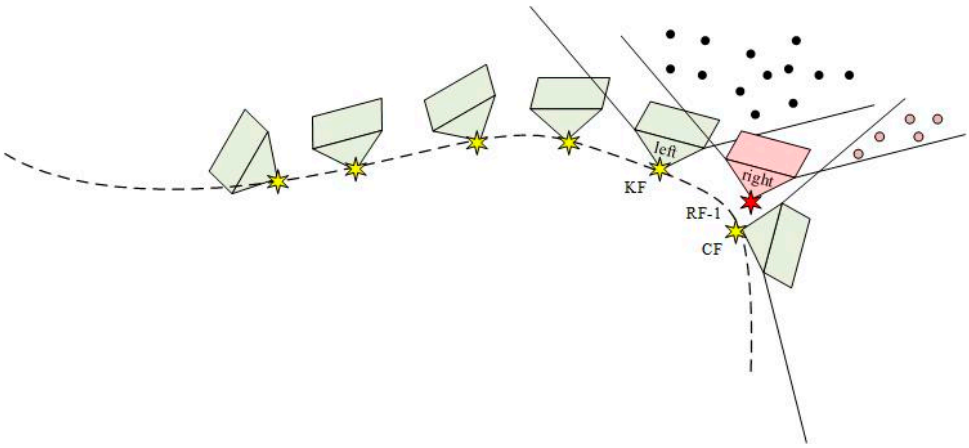

Figure 6. Compensation of spatial clue in the right image frame, where KF represents the key frame, CF represents the current frame, and RF-1 represents the previous frame of the right image.

The system is preset with a maximal threshold of η and a minimal threshold of ξ, comparing the $Tran$ with the threshold are follows:

when $Tran < \xi$, $Frame_{ckey} \neq Frame_{cur}$;
when $\xi \leq Tran \leq \eta$, $Frame_{ckey} = Frame_{cur}$;
when $Tran > \eta$, if $\omega < \frac{\pi}{2}$, $Frame_{key} = Frame_{cur}$, else $\omega \geq \frac{\pi}{2}$, $Frame_{key} = Frame_{rcur} - 1$.

$Frame_{cur}$ is the current frame of the left image, $Frame_{rcur} - 1$ is the previous frame of the right image, $Frame_{ckey}$ is the candidate keyframe, and $Frame_{key}$ is the keyframe.

In the above analysis, when the amount of relative motion between frames $Tran > \eta$, it indicates a significant change in the camera's view, and keyframes should be inserted in time, otherwise the tracking will be lost. When $\xi \leq Tran \leq \eta$, it is the normal motion range of the camera, in which redundant keyframes should be avoided. Taking the amount of relative motion between frames and data association as constraints, combining the candidate keyframes obtained from the above calculation, the keyframes selection for the system needs to satisfy the following two conditions:

(1) The number of tracked covisibility feature points between $Frame_{ckey}$ and the previous keyframe satisfies the following condition: $Track(Frame_{key} - 1, Frame_{ckey}) > \tau_f$.
(2) The number of near points tracked by $Frame_{ckey}$ is less than the threshold τ_t and more than τ_c new near points can be created.

In outdoor environments, such as forest scenes, where most areas are far away from the sensors, the introduction of near and far points for binocular vision as conditions for filtering keyframes is particularly important for improving the localization accuracy of the system. The near point is the feature point in binocular mode where the depth value is less than 40 times the binocular baseline distance, otherwise it is called the far point.

The 3D coordinates obtained by triangulation for the near point are more accurate which can provide information about orientation, translation, and scale. In contrast the far point carries less information which provides only relatively accurate information about orientation. It is very challenging in a large forest scene and at a distance from the camera, the system needs enough near points to accurately estimate the camera's translation, so the system has certain requirements for the number of tracked near points and the number of generated new near points. It works better by setting $\tau_t = 90$, $\tau_c = 50$ in the experiments.

The steps of keyframes selection for the system are as follows.

Step 1: Determine whether the prerequisites for inserting keyframes are met: the system is not currently in localization mode and the local mapping is free, while it is far from the last relocation, and the number of internal points must be greater than the minimum threshold of 15, i.e., mnMatchesInliers > 15.

Step 2: Calculate the relative motion $Tran(R, t)$ between frames and determine the candidate keyframe $Frame_{ckey}$ or keyframe $Frame_{ckey}$ based on the comparison between $Tran(R, t)$ and the thresholds.

Step 3: If it is a candidate keyframe $Frame_{ckey}$, calculate the number of tracked feature points between $Frame_{ckey}$ and the previous keyframe $Frame_{key} - 1$ and perform the judgment of condition 1: $Track(Frame_{key} - 1, Frame_{ckey}) > \tau_f$.

Step 4: Calculate the number of near points tracked by $Frame_{ckey}$ and perform the judgment of condition 2: The number of near points tracked by $Frame_{ckey}$ is less than the threshold τ_t and more than τ_c new near points can be created.

Step 5: If both conditions 1 and 2 are satisfied, which indicates high matching and correlation between frames and the high quality of feature points, the candidate keyframe $Frame_{ckey}$ is set as the keyframe $Frame_{key}$, i.e., $Frame_{key} = Frame_{ckey}$.

Correspondingly, the algorithm of keyframes selection for the system is as follows (Algorithm 1).

Algorithm 1 Keyframe Selection

Input: the binocular image frames $Frame_{cur}$ and $Frame_{rcur}$
Parameter: threshold τ_t, τ_c, τ_f, ξ, η
Output: Keyframe $Frame_{key}$

for each vailable new $Frame_{cur}$ **do**
  calculate $\omega = min\ (2\pi - ||R||\ ,\ ||R||)$
  if $\omega \in [0, \frac{\pi}{2})$ **then**
    $\alpha = (\frac{\tan\omega}{\tan\omega+1})^{\frac{1}{4}}$
    $Tran\ (R,t) = (1-\alpha)\ ||\ t|| + \alpha min\ (2\pi - ||R||\ ,\ ||R||)$
  else
    $\alpha = (\frac{|\tan\omega|}{|\tan\omega|+1})^{\frac{1}{4}}$
    $Tran\ (R,t) = \alpha min(2\pi - ||R||\ ,\ ||R||)$
  end if
  if $Tran\ < \xi$ **then**
    $Frame_{ckey} \neq Frame_{cur}$
  else
    if $\xi \leq Tran\ \leq \eta$ **then**
      $Frame_{ckey} = Frame_{cur}$
    else
      if $\omega < \frac{\pi}{2}$ **then**
        $Frame_{key} = Frame_{cur}$
      else
        $Frame_{key} = Frame_{rcur} - 1$
      end if
    end if
  end if
  calculate the number of covisibility feature points *Track* between $Frame_{ckey}$ and the last keyframe
  if $Track(Frame_{key} - 1, Frame_{ckey}) > \tau_f$ **then**
    calculate the number of near points tracked and the number of new near points created in $Frame_{ckey}$
    if the number of near points tracked in $Frame_{ckey}$ is less than the threshold τ_t and more than τ_c new near points are created **then**
      $Frame_{key} = Frame_{ckey}$
    end if
  end if
end for

4.2. Local Mapping

The local mapping thread implements mid-term data association, it receives keyframes imported from the tracking thread, eliminates substandard map points, generates new map points, performs local map optimization, removes redundant keyframes, and sends optimized keyframes to the loop closing thread. Only the information of adjacent common frames or keyframes is used in the tracking thread; moreover, only the pose of the current frame is optimized, and there is no joint optimization of multiple poses and no optimization of the map points. The Local BA optimizes both multiple keyframes that satisfy a certain covisibility relationship and the corresponding map points, so as to make the keyframes more accurate in terms of poses and map points. More new map points are obtained by re-matching between covisibility keyframes, increasing the number of map points while improving the tracking stability. Removing redundant keyframes helps to reduce the scale and number of Local BAs and improve the real-time performance of the system.

4.3. Loop Closing

Loop closing is divided into two steps: loop closing detection and loop closing correction. Loop closing detection uses Bag of Words (BoW) to accelerate matching, queries

the dataset to detect whether the loop is closed or not, and then computes the Special Euclidean Group (SE3) poses between the current keyframe and the loop closing candidate keyframe. Monocular vision suffers from scale drift while binocular vision easily obtains depth information making the scale observable, so there is no need to deal with scale drift in geometric validation and pose-graph optimization. The loop closing correction focuses on loop closing fusion and essential graph optimization to correct cumulative drift, and starts the Full BA thread for the BA optimization of all map points and keyframes, which is more costly and therefore a separate thread is needed.

4.4. Dense Mapping

The tracking thread calculates the pose for each frame, if the dense map is constructed using each frame in the tracking thread, it not only increases running time and storage space overhead for the system, but it also affects the localization accuracy due to the heavy computation which slows down the system's running speed; hence, the keyframes are used to construct the dense map. Firstly, the disparity map is calculated for each keyframe; secondly, the point cloud is generated by combining the more accurate keyframe poses optimized by Local BA, and then the initial dense map is formed through point clouds registration, point clouds fusion, and point clouds filtering; and thirdly, the dense map is updated by global BA optimization.

Obtaining disparity maps is a key step in dense mapping, while the traditional stereo matching methods have a poor matching effect in the regions of weak texture, occlusion and other features that are not obvious, and the generated disparity map is insufficiently robust. In this work, LANet [27], a linear attention stereo matching network is embedded into D-SLAM as one of the modules to generate dense disparity maps, which are transformed to generate depth maps and point clouds to realize the construction of dense maps of the forest ecological environment. LANet networks are capable of optimizing depth estimation by efficiently utilizing environmental global and local information to improve stereo matching accuracy in ill-posed regions, such as those with weak texture, poor lighting, and occlusion and achieve efficient disparity inference prediction. Because LANet is one of the research findings of the authors of this paper, it has been published in a public paper "LANet: Stereo matching network based on linear-attention mechanism for depth estimation optimization in 3D reconstruction of inter-forest scene" https://www.frontiersin.org/articles/10.3389/fpls.2022.978564/full (accessed on 2 September 2022), and the overview of the LANet as shown in Figure 7.

(1) Feature extraction

ResNet [28] is adopted as the backbone network for feature extraction; all layers use a 3×3 convolutional kernel, and the first stage uses three convolutional layers conv0_1, conv0_2, and conv0_3, to extract the primary features of the image. The second stage uses four sets of basic residual blocks, conv1_x, conv2_x, conv3_x, and conv4_x, to extract the deep semantic features of the image. Downsampling with stride 2 was used in conv0_1 and conv2_1, and the input image size is reduced to 1/4 of the original size after two downsamplings. The dilated convolution is applied to enlarge the receptive field in conv3_x and conv4_x, and the dilation rates of these two layers are 2 and 4, respectively.

(2) Attention Module (AM)

AM can better integrate local and global information to obtain richer feature representations at the pixel level, and the AM consists of two parts: the Spatial Attention Module (SAM) and Channel Attention Module (CAM). SAM captures long-range dependencies between global contexts, seeks correlations between pixels at different locations, and models semantic correlations in the spatial dimension.

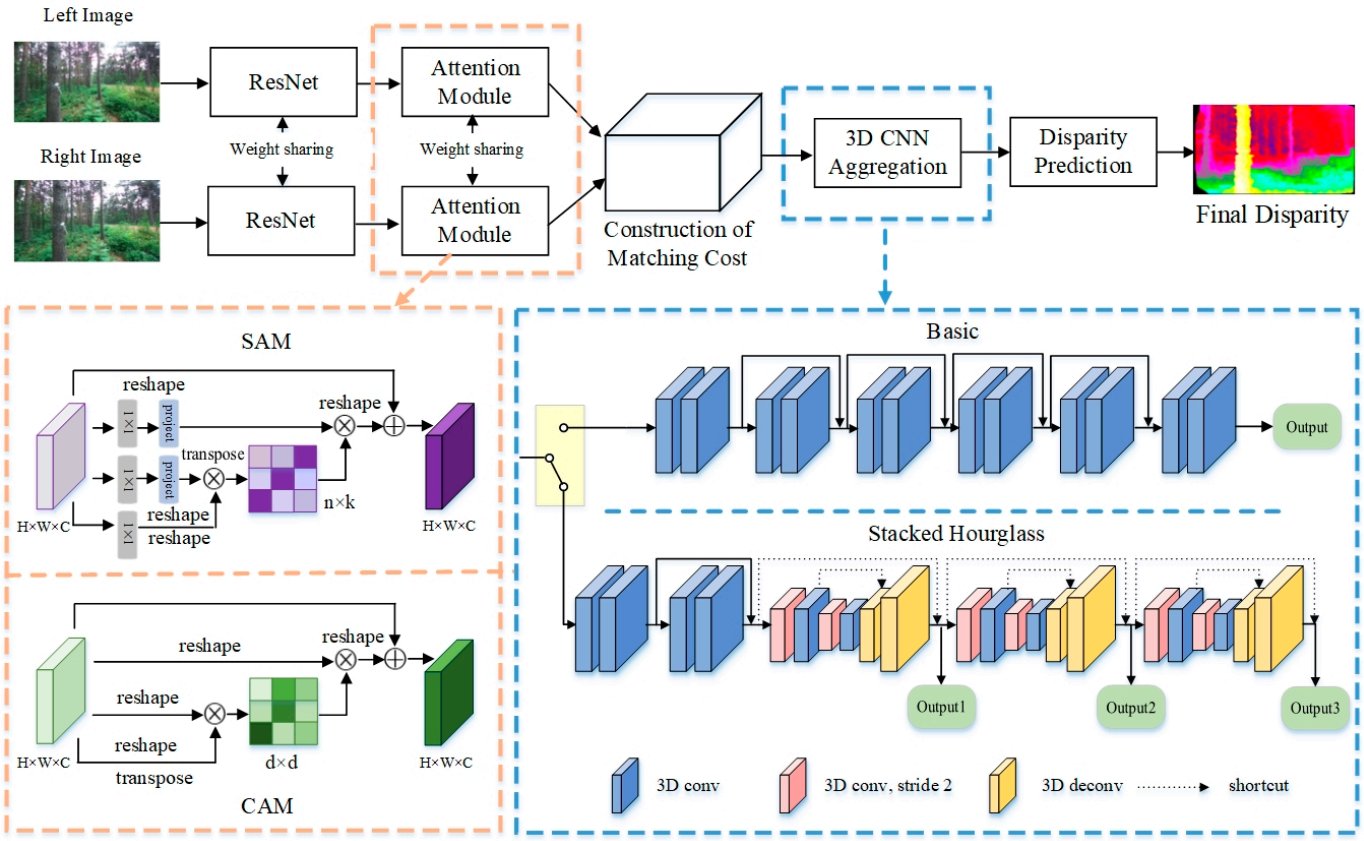

Figure 7. Network structure of LANet. LANet consists of five main parts: feature extraction ResNet, Attention Module (AM), Construction of Matching cost, Three Dimensional Convolutional Neural Network aggregation (3D CNN aggregation), and disparity prediction. The AM consists of two parts: Spatial Attention Module (SAM) and Channel Attention Module (CAM); the 3D CNN aggregation consists of two structures: the basic structure is used for ablation experiments to test the performance of various parts of the network and the stacked hourglass structure is used to optimize the network.

Because the time and space complexity of self-attention [29] is $O(n^2)$, the cost of training and deploying the model is very high when it is used on large-size images. Linear-attention is proposed to be able to reduce the overall complexity of self-attention from $O(n^2)$ to $O(n)$ while retaining high accuracy. The correlation matrix $P \in \mathbb{R}^{n\times n}$ in self-attention is low rank, where most of the information is concentrated in a small number of maximum singular values; hence, a low rank matrix $\overline{P}$ is used to approximate P to reduce the complexity of self-attention by changing its structure. The details are as follows:

Let $X \in \mathbb{R}^{n\times d_m}$ be the input sequence, $W^Q, W^K \in \mathbb{R}^{d_m\times d_k}, W^V \in \mathbb{R}^{d_m\times d_v}$ are three learnable matrices, and $Q = XW^Q, K = XW^K, V = XW^V$, the query matrix, the key matrix and the value matrix $Q, K, V \in \mathbb{R}^{n\times d_m}$ embedded in the input sequence are obtained respectively. where n is the length of the sequence and d_m , d_k , d_v are the dimensions of the hidden layers of the projection space. Two low dimensional linear projection matrices $E \in \mathbb{R}^{n\times k}$ and $F \in \mathbb{R}^{n\times k}$ are constructed, which are fused with K and V to reduce their dimensionality. E or F performs matrix multiplication with K or V to reduce K and V from their original $n \times d$-dimension to the $k \times d$-dimension, as shown in Figure 8.

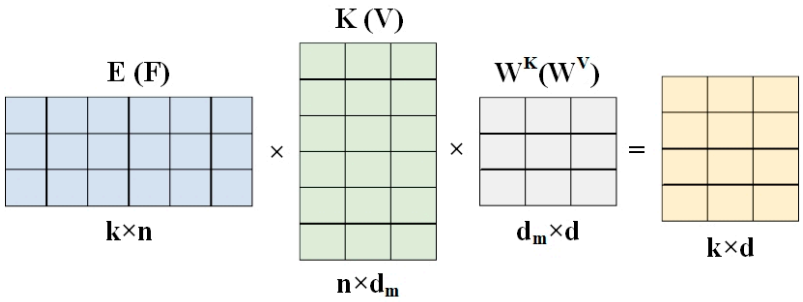

Figure 8. Linear mapping layers.

The correlation matrix $\overline{P} \in \mathbb{R}^{n \times k}$ is computed by the scaled dot product method, and the value of linear-attention is

$$\overline{P} \cdot (FVW^V) \tag{11}$$

where

$$\overline{P} = softmax\left[\frac{QW^Q(EKW^K)^T}{\sqrt{d}}\right] \tag{12}$$

and the complete form of linear-attention is

$$Linear - Attention\left(QW^Q, KW^K, VW^V\right) = softmax\left[\frac{QW^Q(EKW^K)^T}{\sqrt{d}}\right] \cdot (FVW^V) \tag{13}$$

The complexity of linear-attention is mainly determined by $\overline{P}$, $O(\overline{P}) = O(nk)$, and if a very small mapping dimension k is choosen and set to $k << n$, the overall complexity of $\overline{P}$ will decrease to linear $O(n)$. It can be proven that when $k = O(nd/\varepsilon^2)$, the value of $\overline{P} \cdot (FVW^V)$ approaches $P \cdot (VW^V)$, and the value of linear-attention can be approximately equivalent to that of self-attention, with an error of no more than ε.

The feature values obtained through linear-attention are multiplied by the scale factor α and then summed bit-wise with the original features $X \in \mathbb{R}^{H \times W \times C}$ to obtain the spatial attention feature map $Y \in \mathbb{R}^{H \times W \times C}$ as follows.

$$Y_j = \alpha \sum_{i=1}^{n} (\overline{P}_{ij} F_i V_i W_i^V) + X_j \tag{14}$$

where feature Y_j is the weighted sum of the features at all locations and the original location feature X_j. Therefore, it has global contextual information, and spatial attention can fuse similar features in the global spatial range, which is conducive to the consistent expression of feature semantics, and likewise it enhances the robustness of feature extraction in ill-posed regions.

Each channel corresponds to a feature map of a specific category of semantics. CAM models semantic relevance in the channel dimension, capturing long-range semantic dependencies between channel features, enabling global correlations between each channel, which is beneficial for obtaining stronger semantic feature responses and improving feature recognition. CAM is calculated on the original feature map based on the self-attention mechanism, without involving the complexity of $O(n^2)$.

The input feature $X \in \mathbb{R}^{H \times W \times C}$ is reshaped into $Q', K', V' \in \mathbb{R}^{n \times d}$ and there exists $Q' = K' = V'$, where $n = \frac{1}{4}H \times \frac{1}{4}W$, $d = C$, and the channel correlation matrix $P' \in \mathbb{R}^{d \times d}$ is obtained by multiplying the matrices between Q'^T and K' as follows.

$$P'_{ji} = softmax\left[\frac{Q'^T K'}{\sqrt{d}}\right] = \frac{\exp\left[\frac{Q'^T{}_i K'_j}{\sqrt{d}}\right]}{\sum_{i=1}^{C} \exp\left[\frac{Q'^T{}_i K'_j}{\sqrt{d}}\right]} \tag{15}$$

In the Equation (15), P'_{ji} denotes the correlation between the i th channel and the j th channel, and the higher the correlation between the two channel features, the greater the value of P'_{ji}. The final feature for each channel is a weighted sum of the features of all channels and the original feature as follows.

$$Z_j = \beta \sum_{i=1}^{C} (V'_i P'_{ji}) + X_j \tag{16}$$

where Z is the final feature. The self-attention feature map of $\mathbb{R}^{n \times d}$ is obtained by multiplying the matrices between V' and p', which is reshaped into the form of $\mathbb{R}^{H \times W \times C}$ multiplied by a scale factor β, and it is then summed bit-wise with the original feature map $X \in \mathbb{R}^{H \times W \times C}$ to finally obtain the channel attention feature map $Z \in \mathbb{R}^{H \times W \times C}$.

(3) Construction of Matching cost

The feature information from the four parts of conv2_16, conv4_3, SAM and CAM is cascaded to form a 2D 1/4H × 1/4w × 320 feature map which is fused by two convolutional layers of 3 × 3 and 1 × 1 while the channels are compressed to 32, connecting the left 2D feature map with the right feature map corresponding to each disparity to construct a 4D matching cost-volume of 1/4D × 1/4H × 1/4W × 64.

(4) The 3D CNN aggregation

The 3D CNN aggregation module is used for cost-volume regularization, aggregating semantic and structural feature information in disparity and spatial dimensions to predict accurate cost-volume. It consists of two structures: the basic structure is used for ablation experiments to test the performance of various parts of the network, and it consists of twelve convolutional layers with a convolution kernel size of 3 × 3 × 3 performing BN and ReLU. The stacked hourglass structure is used to optimize the network and increase the robustness of disparity prediction in low-texture regions and occluded regions to obtain more accurate disparity values. The first four 3D convolutional layers contain BN and ReLU, and the 3D stacked hourglass network utilizes an "encoder-decoder" structure to reduce the parameters and computation of the network. The encoder downsamples twice by using a 3D convolution with a convolution kernel of 3 × 3 × 3 and a step size of 2. Correspondingly the decoder upsamples twice to recover the size by using an inverse convolution with a step size of 2, while the number of channels is halved. To compensate for the information loss caused by the "encoder–decoder" structure, a 1 × 1 × 1 3D convolution is used inside each hourglass module to connect features of the same size directly, which uses fewer parameters than a 3 × 3 × 3 convolution, and reduces the computational power to 1/27 of the original one, with negligible runtime; thereby, the running speed of the network is improved without increasing the computational cost.

(5) Disparity prediction

Each hourglass corresponds to one output, the total loss is a weighted sum of the losses corresponding to each output, and the last output is the final disparity map. A differentiable Soft Argmin function was utilized to obtain disparity estimation $\hat{d}$ through the regression method as follows. Equations (17)–(19) are from reference [30].

$$\hat{d} = \sum_{k=0}^{D_{\max}-1} k \cdot p_k \tag{17}$$

where $D_{\max}$ denotes the maximum disparity. With the L1 loss function, the total loss is calculated as follows.

$$L = \sum_{i=1}^{3} \lambda_i \cdot Smooth_{L_1}(\hat{d}_i - d_i) \tag{18}$$

where λ_i denotes the coefficient of the i th disparity prediction, d_i denotes the true value of the i th disparity map, $\hat{d}_i$ denotes the i th predicted disparity map, and the $Smooth_{L_1}(x)$ function is expressed as follows.

$$Smooth_{L_1}(x) = \begin{cases} 0.5x^2 & , if|x| < 1 \\ |x| - 0.5 & , otherwise \end{cases} \quad (19)$$

5. Results

The experiment and evaluation of the whole system is split in four parts:

- LANet performs the prediction training and evaluation of disparity maps on the Scene Flow and Forest datasets, and compares them with several mainstream methods.
- D-SLAM tests the accuracy of projection trajectories on two datasets, EuRoC and KITTI, and compares them with mainstream SLAM systems.
- D-SLAM tests the partial and overall performance of the system on three datasets, EuRoC, KITTI and Forest.
- D-SLAM performs real-time dense mapping on two datasets, KITTI and Forest, as well as analyzing and discussing the mapping results.

5.1. Experiment on Disparity Map Generation by LANet

LANet is pre-trained on the clean pass dataset of Scene Flow [31], and fine-tuned training is conducted on the Forest target dataset. Network training was based on Python 3.9.7, the PyTorch 1.11.0 framework, one Nvidia TITAN Xp GPU 3090 for the server, and Adam [32] for the optimizer, with β1 = 0.9, β2= 0.999, and batch size set to eight.

(1) Ablation experiments on Scene Flow

Ablation experiments are carried out on the Scene Flow dataset to test the performance of each key module and parameter in the network. In Table 2, Res is the ResNet module, SA denotes the Spatial Attention Module using a self-attention mechanism, SAM denotes the Spatial Attention Module using a linear-attention mechanism, CAM denotes the Channel Attention Module, k is the dimensionality of E and F in the model, E and F share the same parameter, i.e., E = F, Basic denotes the basic structure, and Hourglass is stacked hourglass network. Experiments were conducted to evaluate the performance of each key module with evaluation metrics which are >1, >2, and >3pixel error, End Point Error EPE, and runtime.

Table 2. Ablation experiments of attention mechanism on Scene Flow.

Module	>1 px (%)	>2 px (%)	>3 px (%)	EPE (px)	Runtime (s)
Res_Base	12.78	8.11	6.41	1.65	0.12
Res_CAM_Base	11.12	7.02	5.36	1.21	0.14
Res_SA_Base	10.24	6.48	4.91	1.03	0.24
Res_SAM_k128_Base	10.47	6.65	5.04	1.10	0.16
Res_SAM_k256_Base	10.38	6.58	4.98	1.07	0.17
Res_SAM_k512_Base	10.29	6.52	4.93	1.05	0.18
Res_CAM_SAM_k512_Base	9.26	5.56	3.95	0.95	0.19
Res_CAM_SAM_k512_Hourglass	7.22	3.71	2.31	0.82	0.25

As shown in Table 2, the EPE of Res_Base is 1.65, and with the addition of CAM and SAM, the EPE becomes 1.21 and 1.1, respectively, resulting in a significant reduction in error rates. The error rate of disparity values shows that adding AM can significantly improve the accuracy of disparity prediction, thereby achieving the goal of improving the accuracy of dense mapping in D-SLAM systems. When the value of k becomes larger, the EPE of Res_SAM_kx_Base gradually approaches that of Res_SA_Base, and when k = 512, the EPEs of both are almost equal, while the inference time of the former changes little which is significantly faster than that of Res_SA_Base. Thereby, it is proved that the inference speed

of linear-attention is significantly faster than that of self-attention when their error rates are close. Compared to Res_CAM_SAM_k512_Base, Res_CAM_SAM_k512_Hourglass has a significant advantage which reduces the error rate for the whole network > 3 px from 3.95 to 2.31 and EPE from 0.95 to 0.82.

(2) Comparative experiments on Forest

Several mainstream methods are compared in the Forest dataset, and the performance of each method was evaluated by three evaluation metrics, the proportion of pixels with prediction errors in all regions of the first frame image (D1-all), EPE and time. The test server was 3090GPU, and the image resolution was 1240 × 426.

The results in Table 3 indicate that after fine-tuning on the Forest dateset, LANet exhibits better performance than on the SceneFlow dataset, with an EPE reduction from 0.82 to 0.68 and an accuracy improvement of 20.6%. The D1 all and EPE of LANet are 2.15 and 0.68, respectively, which are better than those of the comparative model. The running speed is 0.35 s, and although it is not the fastest, it is also relatively competitive.

Table 3. Comparative experiments of disparity detection with Forest. The network models for comparison are Matching Cost with a Convolutional Neural Network (MC-CNN), Geometry and Context network (GCNet), Learning deep correspondence through prior and posterior feature constancy (iResNet), Disparity Network (DispNet), a two-stage convolutional neural network (CRL), Exploiting Semantic Information for Disparity Estimation (SegStereo), Edge Stereo network (EdgeStereo), and Pyramid Stereo Matching Network (PSMNet).

Method	Runtime (s)	D1-All (%)	EPE (px)
MC-CNN [33]	67.09	4.08	3.96
GCNet [34]	1.01	3.65	2.79
iResNet [35]	0.20	3.58	2.73
DispNet [31]	0.14	3.08	1.96
CRL [36]	0.55	2.75	1.54
SegStereo [37]	0.68	3.12	2.01
EdgeStereo [38]	0.40	2.81	1.68
PSMNet [39]	0.48	2.61	1.25
LANet	0.35	2.15	0.68

Figure 9 shows the visualization of disparity maps generated by LANet, PSMNet and GCNet with Forest, with the colors representing different disparity values, the farther the distance the smaller the disparity value, and the black color indicating the distant points, whose disparity values are so small that they can be ignored.

The rectangular box regions where the matching error of each method is large are usually found in locations containing fine structures such as branches, trunks, and leaf edges, as well as weakly textured regions and occluded regions. In column A, there are significant differences in the predictions of each model at the border between the pink trees and the crimson sky, and PSMNet and GCNet can preserve the main contours of the edges while the predictions are inaccurate at the fine structures; however, the LANet can better preserve the fine features of the edges while the predictions are closer to the true value. In column B, for the prediction of the red trunk, PSMNet and GCNet show missing trunk pixels, and for the prediction of the pink car's rear glass, LANet shows few color deviations, PSMNet shows more color deviations, and GCNet shows more errors in color. In column C, for the red trunk prediction, LANet shows pixel discontinuity and few missing pixels, PSMNet and GCNet show larger missing pixels or even missing trunks, and for the prediction of the purple leaf, LANet is able to retain the edge features better, PSMNet misses some fine edge structure features, and GCNet has too many edge predictions and a mismatched pieces.

The attention mechanism integrates local and global information, seeking correlations between pixels at different locations to obtain richer feature representations at the pixel

level, which are beneficial for obtaining stronger semantic feature responses and improving feature recognition. Therefore, it can predict more reliable disparity maps in ill-posed regions such as those with weak texture, poor lighting, and occlusion. After testing, LANet has shown better performance than the comparative model in terms of accuracy and visualization, and it is also more competitive in terms of runtime.

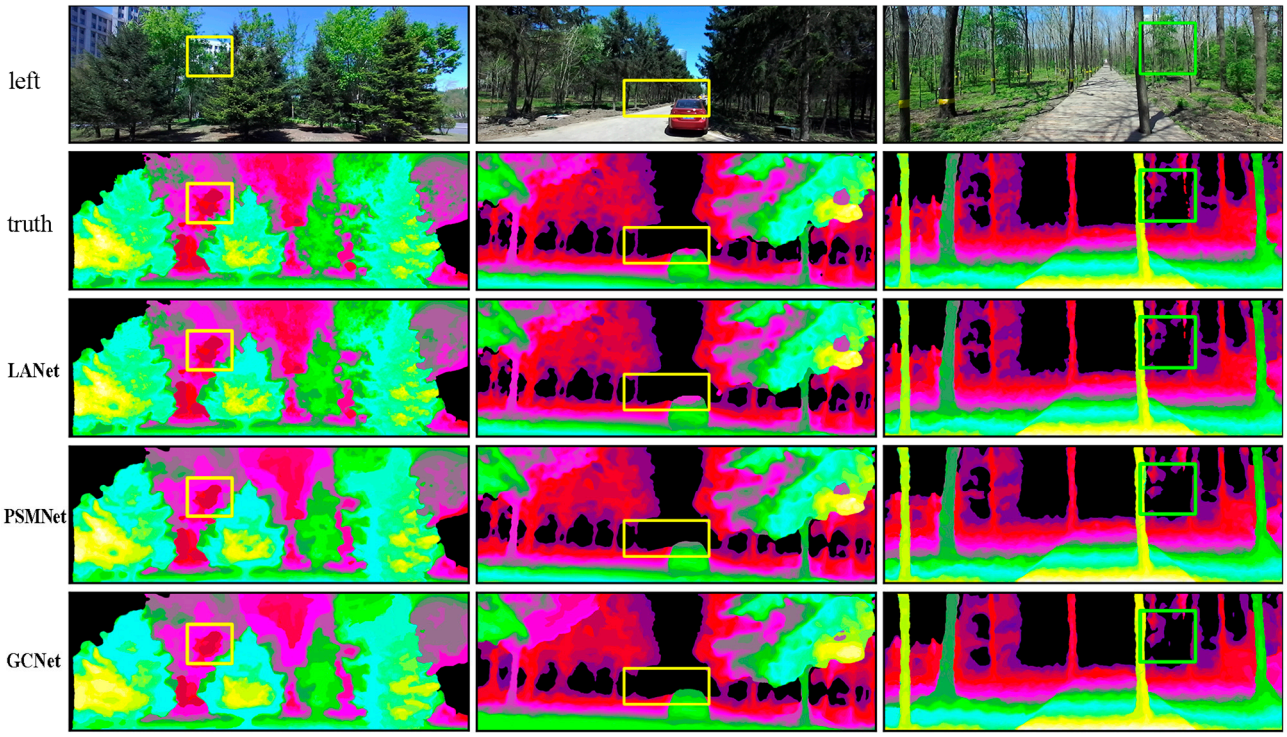

Figure 9. The visualization of disparity maps with Forest. The yellow or green boxes are the regions with significant disparity contrast generated by various methods.

LANet is embedded into the D-SLAM system, and is lightweighted in order to ensure the real-time performance of the system, and the stacked hourglass structure used to optimize the network is cut off to improve the running speed of the system. For the purpose of making a balance between accuracy and speed, the Res_CAM_SAM_k512_Base combination modules are selected, with an EPE of 0.95 and a running time of 0.19 s, which fully meets the performance needs of the dense mapping thread of D-SLAM.

5.2. Experiment on the Location Accuracy of Visual Odometry

In this section, the performance of D-SLAM will be evaluated for several sequences on two popular datasets. In order to demonstrate the robustness of the proposed system, the estimation of camera generated trajectories and maps was compared with the Ground Truth (GT). In addition, the results are also compared with some advanced SLAM systems by using the results published by the original author and standard evaluation metrics in the literature. D-SLAM experiments are all conducted on a Dell G3 3590 portable computer, with an Intel Core i7-9750H CPU, 2.6GHz, 16GB memory, and only the CPU was used.

5.2.1. EuRoC Dataset

The EuRoC dataset [40] contains 11 stereo sequences recorded from a micro aerial vehicle (MAV) flying around two different rooms and a large industrial environment. The baseline of the binocular sensor is 11 cm, providing images at 20 Hz. The sequences are classified into three levels: easy, medium, and difficult based on the speed, illumination, and scene texture of the MAV.

(1) Estimated trajectory maps

Figure 10 is the estimated trajectory for nine sequences from EuRoC; by comparison with the GT, the D-SLAM system shows better trajectory accuracy on these sequences.

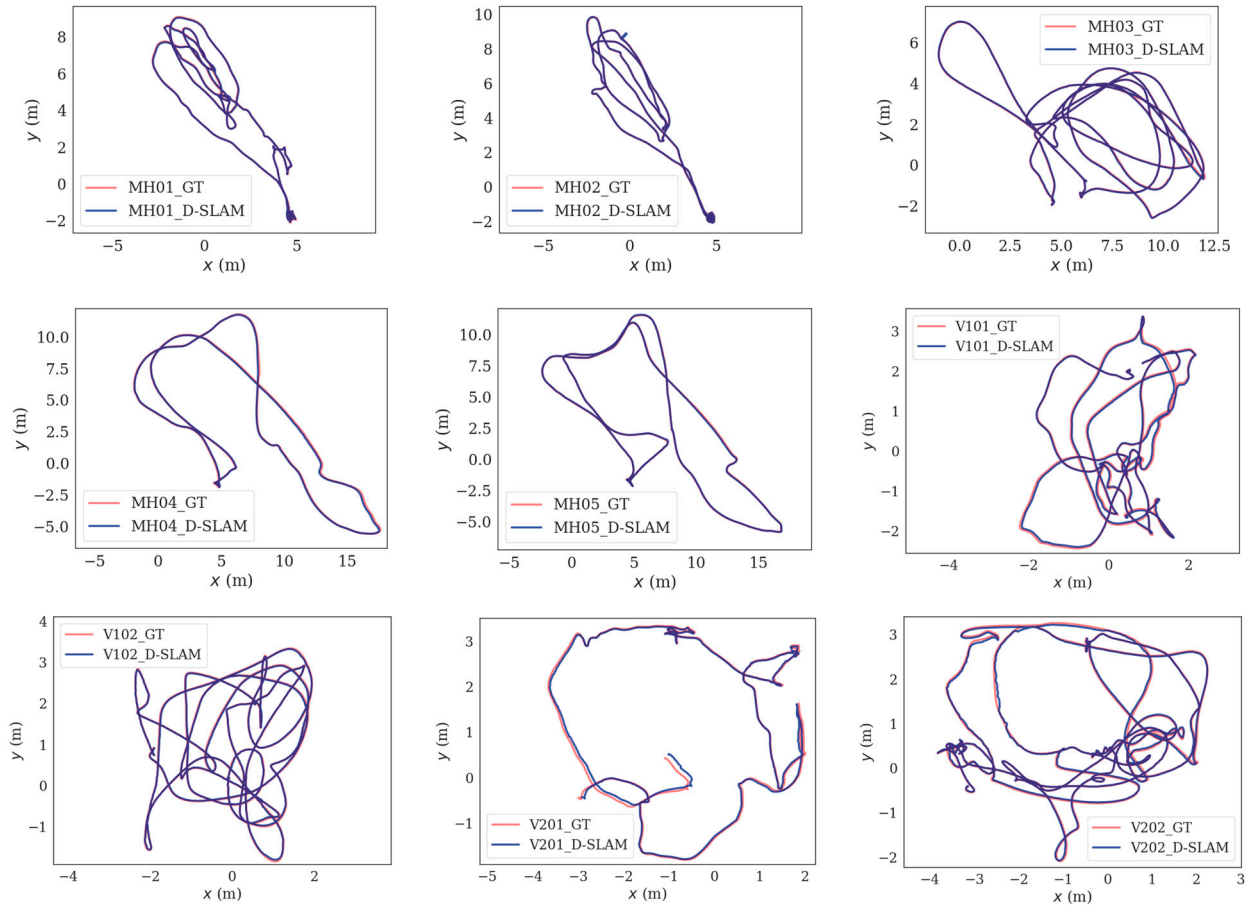

Figure 10. Estimated trajectory (dark blue) and GT (red) for 9 sequences on EuRoC.

(2) RMS ATE comparison

As is usual in the field, the accuracy is measured by RMS ATE [41]. The SE (3) transform is used to align the estimated trajectory with the GT. The result of D-SLAM is the average of five executions, while the other results are reported by the authors of each system and compared with GT for all frames in the trajectory.

As shown in Table 4 ORB-SLAM2 and VINS-Fusion all get lost in some parts of V2_03_difficult sequence due to severe motion blur. Even BASALT, a stereo vision-inertial odometry system, was not able to complete tracking on this sequence due to the loss of some frames by one of the cameras. However, due to the use of the binocular image spatial clue compensation strategy, D-SLAM could utilize the right image to compensate for the lost field of view of the camera to a certain extent when the above situations occured, and it successfully tracked and achieved an error of 0.468 on the V2_03_ difficult sequence. SVO is a semi direct visual odometry that can run in weak texture and high frequency texture environments. However, the pose estimation has significant cumulative error due to its lack of loop detection and relocation. The system is very dependent on the accuracy of pose estimation, making it difficult to relocate once tracking fails. It performs well on easy sequences, while the tracking accuracy decreases rapidly on medium and difficult sequences, with a larger RMS ATE.

Table 4. RMS ATE comparison in the EuRoC dataset (RMS ATE in m, scale error in %). The network models for comparison are ORB-SLAM2, a general optimization-based framework for local odometry estimation with multiple sensors (VINS-Fusion), Semidirect visual odometry for monocular and multicamera systems (SVO2), Visual-Inertial Mapping with Non-Linear Factor Recovery (BASALT), and D-SLAM.

Sequence	ORB-SLAM2 [18]	VINS-Fusion [42]	SVO2 [43]	BASALT [44]	D-SLAM
MH_01_easy	0.035	0.540	0.040	0.070	0.032
MH_02_easy	0.018	0.460	0.070	0.060	0.018
MH_03_medium	0.028	0.330	0.270	0.070	0.026
MH_04_ difficult	0.119	0.780	0.170	0.130	0.095
MH_05_ difficult	0.060	0.500	0.120	0.110	0.055
V1_01_easy	0.035	0.550	0.040	0.040	0.032
V1_02_medium	0.020	0.230	0.040	0.050	0.022
V1_03_difficult	0.048	–	0.070	0.100	0.041
V2_01_easy	0.037	0.230	0.050	0.040	0.035
V2_02_medium	0.035	0.200	0.090	0.050	0.030
V2_03_difficult	–	–	0.790	–	0.468

5.2.2. KITTI Dataset

The KITTI [45] dataset has become the standard for evaluating visual SLAM, which contains stereo images recorded from a car in urban and highway environment. The baseline of the binocular sensor is 54 cm and works at 10Hz with a rectified resolution of 1240 × 376 pixels. The D-SLAM system was tested on 11 sequences of the KITTI dataset, and the results are as follows.

(1) Error graph for the 05 sequence of KITTI

The trajectory error is calculated by Evaluation Visual Odometry (EVO) tool. Absolute Pose Error (APE) calculates the difference between the estimated value of the SLAM system and the ground truth of the camera's pose, which is suitable for evaluating the accuracy of the algorithm and the global consistency of the camera trajectory. Relative Pose Error (RPE) calculates the difference between the estimated pose change and the true pose change at the same two timestamps, which is suitable for evaluating the drift of the system and the local accuracy of the camera trajectory.

Because the KITTI dataset contains large outdoor urban and highway scenes, its overall APE error is much higher than that of the indoor dataset EuRoC, and the larger errors occur near the turns or where no loop closing occurs at the edge of the trajectory, as shown in Figure 11. In the translation direction of this sequence, the mean of APE is 1.310438 m, the median is 1.184211 m, the rmse is 1.469650 m, and the std is 0.665299. Compared with APE, RPE is smaller, with a mean of 0.015108 m, a mean of 0.013262 m, a rmse of 0.018214 m, and a std of 0.010172 m.

(2) Comparison of projection trajectories for the 08 sequence on KITTI

Figure 12 shows the comparison between the projection trajectories of D-SLAM, Large-scale direct SLAM with stereo cameras (LSD-SLAM) [46], and a stereo SLAM system through the combination of Points and Line segments (PL-SLAM) [47] on the KITTI08 sequence and the GT, from which it can be intuitively observed that the trajectory of D-SLAM is closer to the GT compared to the other two methods, while the trajectory drift of PL-SLAM is relatively large. Unlike D-SLAM, the inferior performance of PL-SLAM is mainly explained by the fact that it does not perform LBA in every frame, so the drift along the trajectory is not corrected, especially in sequences like 08 without a loop closing, resulting in a relatively large final drift of the trajectory. In addition, the translation and rotation deviations on the y-axis are relatively large for the various methods.

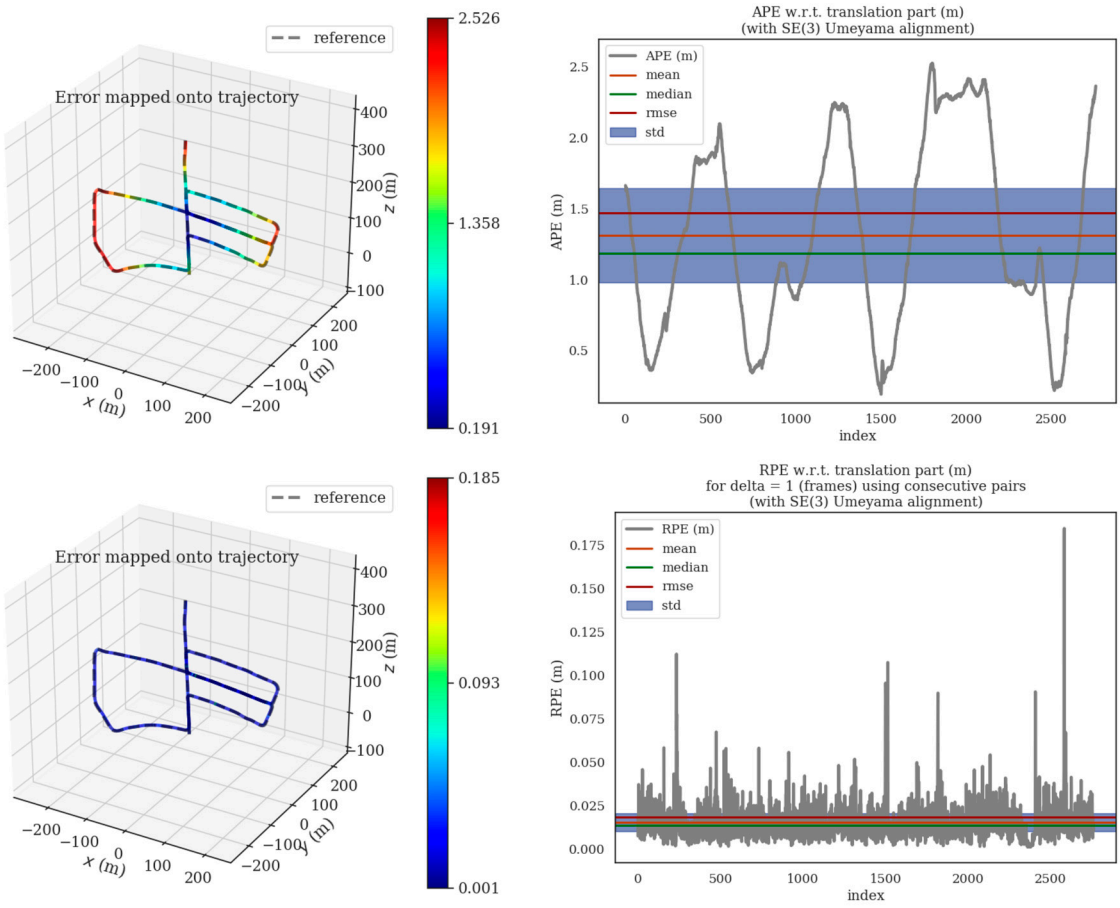

Figure 11. Error graph for the 05 sequence of KITTI.

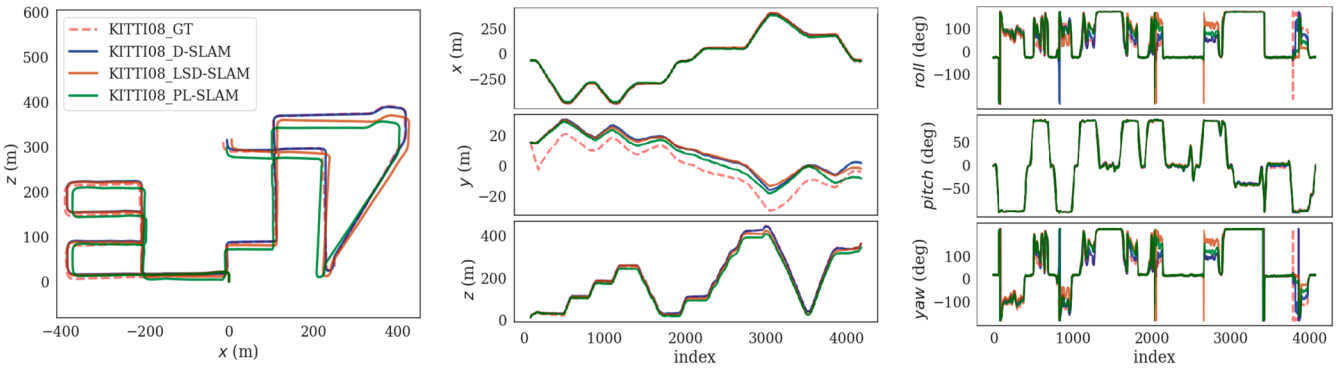

Figure 12. Comparison of projection trajectories for the 08 sequence of KITTI.

(3) Comparison of relative RMSE of KITTI

The metrics of the average relative translation error (t_{rel}) and rotation error (R_{rel}) are used to estimated the relative RMSE [41]. The translation error t_{rel} is expressed in %, the rotation error R_{rel} is also expressed deg/100 m relative to the translation, and the dash indicates that the experiment failed. The comparison of relative RMSE of KITTI is shown in Table 5.

Table 5. Relative RMSE of KITTI.

Sequence	ORB-SLAM2		LSD-SLAM		PL-SLAM		D-SLAM	
	t_{rel}	R_{rel}	t_{rel}	R_{rel}	t_{rel}	R_{rel}	t_{rel}	R_{rel}
00	0.70	0.25	0.63	0.26	2.36	0.89	0.67	0.24
01	1.39	0.21	2.36	0.36	5.80	2.32	1.05	0.19
02	0.76	0.23	0.79	0.23	2.35	0.91	0.68	0.21
03	0.71	0.18	1.01	0.28	3.74	1.54	0.65	0.16
04	0.48	0.13	0.38	0.31	2.21	0.30	0.45	0.13
05	0.40	0.16	0.64	0.18	1.74	0.88	0.38	0.15
06	0.51	0.15	0.71	0.18	3.51	2.72	0.45	0.14
07	0.50	0.28	0.56	0.29	1.83	1.03	0.44	0.25
08	1.05	0.32	1.11	0.31	2.18	1.15	0.98	0.30
09	0.87	0.27	1.14	0.25	1.68	0.92	0.75	0.24
10	0.60	0.27	0.72	0.33	1.21	0.99	0.55	0.23
Avg.	0.72	0.22	0.91	0.27	2.60	1.24	0.64	0.20

The two sequences with large errors in Table 5 are 01 and 08, neither of which show a loop closing. The 01 sequence is the only highway sequence in the KITTI dataset, in which few near points can be tracked due to the high speed and low frame rate, so it is difficult to estimate the translation, and the t_{rel} of the various methods are large. However, there are many distant points that can be tracked for long periods of time, and therefore, the rotation can be accurately estimated. ORB-SLAM2 is able to achieve a better error with an R_{rel} value of 0.21 deg/100 m, while D-SLAM is even smaller with a value of 0.19 deg/100 m. Without any loop closing in the 08 sequence, PL-SLAM is unable to correct the drift of the trajectory in time for the absence of Local BA, whereas ORB-SLAM2 and Stereo LSD-SLAM, although they perform Local BA for each frame, cannot perform Full BA due to the absence of loop closing in this sequence, which also results in the global error not being corrected, leading to a large cumulative error. The D-SLAM system is relatively accurate in the pose estimation of each previous frame, and even without loop closing correction, the drift will not be too severe. The D-SLAM system achieved an average t_{rel} of 0.64m and an average R_{rel} of 0.20, which is more accurate compared to some mainstream stereo systems and has significant advantages in most cases.

5.3. System Real-Time Evaluation

In order to evaluate the real-time performance of the proposed system, the runtimes of different resolutions of the three datasets are presented in Table 6. Because each of these sequences contains only one loop closing, the BA and Loop shown in the table are measurements where the associated task is executed only once.

Because the loop closing of the Psv_02 sequence of Forest contains more keyframes, the covisibility graph is constructed more densely, resulting in higher costs for loop fusion, as well as the higher cost of pose map optimization and Full BA tasks. In addition, the higher the density of covisibility graph, the more keyframes and points the local map contains, resulting in higher costs for local map tracking and LocalBA.

The two threads of loop closing and Full BA in Table 6 consume more time, especially Full BA, for which the D-SLAM system takes 1.42 s. However, these two operations are executed in separate threads, so they do not affect the real-time performance of the other components of the system. The real-time performance of the SLAM system is mainly determined by the speed at which the tracking thread processes each frame of the RGB image, while local mapping, dense mapping, and loop closing threads only process key frames without the need for real-time operation.

The running time of the system on the three sequences is 139.87 s, 124.9 s and 162.97 s, respectively. According to the frame rate and time of tracking threads, the D-SLAM system is able to run at 30 ordinary frames and 3 keyframes per second, which fully meets the

real-time requirements of the SLAM system for forest environment location and dense map construction.

Table 6. Running time of each thread in miliseconds (ms). Where FPS represents Frames Per Second, Essential Graph Opt. represents Essential Graph Optimization, KFs represents KeyFrames, and MPs represents Map Points.

Part	Detail	EuRoC	KITTI	Forest
Settings	Sequence	V2_02	07	Psv_02
	Sensor	Stereo	Stereo	Stereo
	Resolution	752 × 480	1226 × 370	672 × 376
	Camera FPS	20Hz	10Hz	30Hz
	ORB Features	1200	2000	1200
Tracking	Stereo Rectification	2.95	–	–
	ORB Extraction	11.52	21.85	9.69
	Stereo Matching	10.54	13.64	8.81
	Pose Prediction	2.15	2.25	2.05
	Local Map Tracking	9.25	4.21	8.86
	Keframe Selection	5.65	6.12	5.29
	Total	42.06	48.07	34.70
Local Mapping	Keyframe Insertion	8.56	10.03	8.24
	Map Point Culling	0.24	0.38	0.22
	Map Point Creation	35.36	42.26	33.05
	Local BA	135.02	66.35	180.14
	Keyframe Culling	3.61	0.89	2.14
	Total	182.79	119.91	223.79
Dense Mapping	Pcd generation	123.85	138.54	95.37
	Pcd registration	18.87	22.36	15.29
	Pcd fusion	26.59	33.84	23.68
	Pcd filter	32.01	43.35	28.87
	Total	201.32	238.09	163.21
Loop Closing	Database Query	3.25	3.59	3.07
	SE3 Estimation	0.58	0.87	0.51
	Loop Fusion	20.23	79.86	298.25
	Essential Graph Opt.	71.36	175.97	268.95
	Total	95.42	260.29	570.78
Full BA	Full BA	345.71	1120.51	1420.36
	Map Update	3.09	9.65	6.58
	Total	348.8	1130.16	1426.94
Map Size	KFs	249	241	354
	MPs	14,027	26,074	17,325
	Run time	139.87s	124.9s	162.97s

5.4. Dense Mapping

This section will evaluate the dense mapping performance of D-SLAM on two challenging datasets, KITTI and Forest. Six visualized images generated during the dense mapping process are displayed on each dataset. In order to analyze the dense point cloud more comprehensively and intuitively, local detail point cloud images obtained from different perspectives are displayed also. D-SLAM supports two operating modes: online real-time dense mapping and bag video dense mapping. In order to facilitate parameter adjustment, the bag video mode was used for testing in this study.

5.4.1. Dense Mapping on KITTI Dataset

Figure 13 shows the dense map construction of the 01 sequence of the KITTI dataset. This sequence is a real-time image of a highway. Due to its high speed and low frame rate, the camera in this scene has a large translation, little rotation, and no loop closing,

making it challenging. It can be observed from the Figure 13d that the projection trajectory has better accuracy in the straight section of the highway, while there is some slight drift near the turning at the end, which is due to the fact that there is no loop closing for Full BA, resulting in an increase in the cumulative error of the trajectory and ultimately an increase in drift. In Figure 13e, red points represent the covisibility observation points of the covisibility graph keyframes, i.e., reference map points, while black points represent all map points generated by keyframes. Figure 13f is the overall dense point cloud map generated after the previous steps of processing, and its local details are shown in Figure 14. It shows the local dense point cloud maps from different views of the KITTI01, in which the details of the highway can be clearly seen, including the dotted lines, crosswalks, tree shadows, and green grass along the highway. The dense point cloud maps generated from the KITTI dataset are clearer due to the fact that the KITTI dataset is a high-resolution image dataset, coupled with the long baseline of the binocular sensors, and the corrected stereo images.

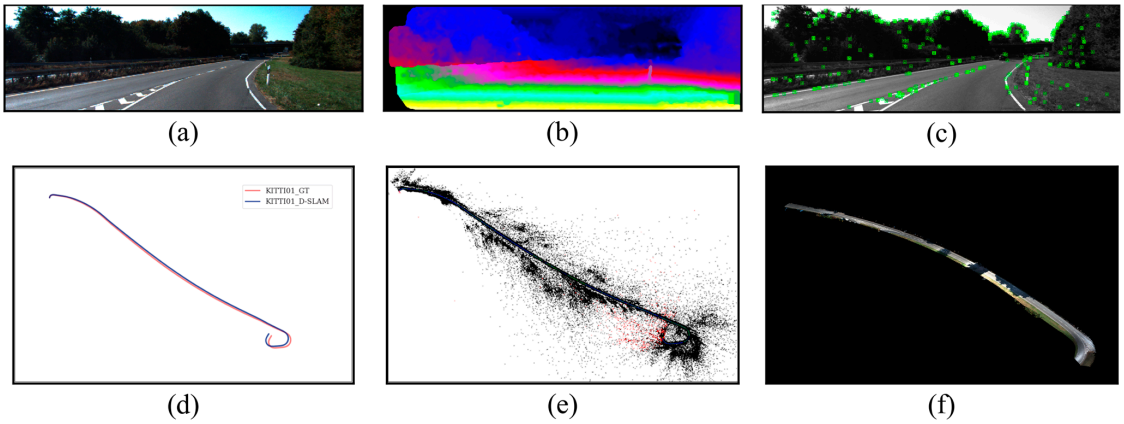

Figure 13. Dense mapping Effect for 01-sequence of KITTI.(**a**) a left RGB image, (**b**) a visual disparity map, (**c**) afeature point tracking map, (**d**) anestimated trajectory map, (**e**) asparse point cloud map, and (**f**) a dense point cloud map.

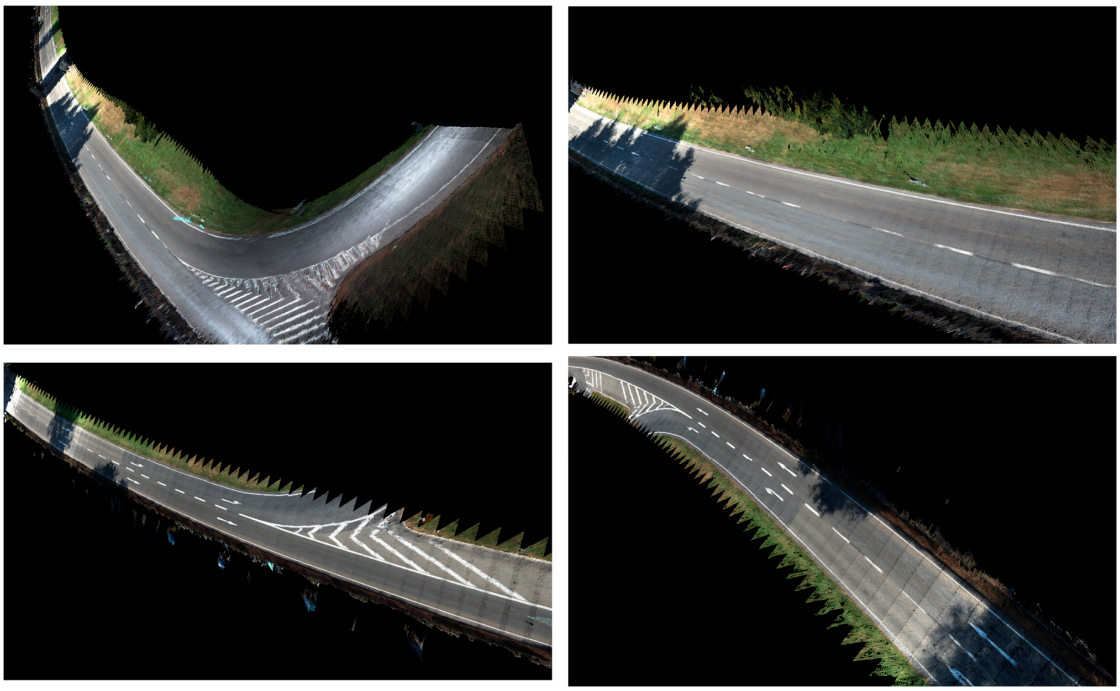

Figure 14. Local dense point cloud maps from different views for the KITTI01 sequence.

5.4.2. Dense Mapping with Forest Dataset

Forest is a large forest scene dataset with low texture images, the trunk features are very inconspicuous, and the number of feature points is not large enough, in order to have enough near point feature points to ensure the tracking accuracy and the effect of dense mapping, it is necessary to insert as many keyframes as possible, and at the same time to avoid redundancy, based on which the keyframe selection strategy has been designed previously for the characteristics of the forest scene. Figure 15 shows the dense mapping process with the Forest dataset.

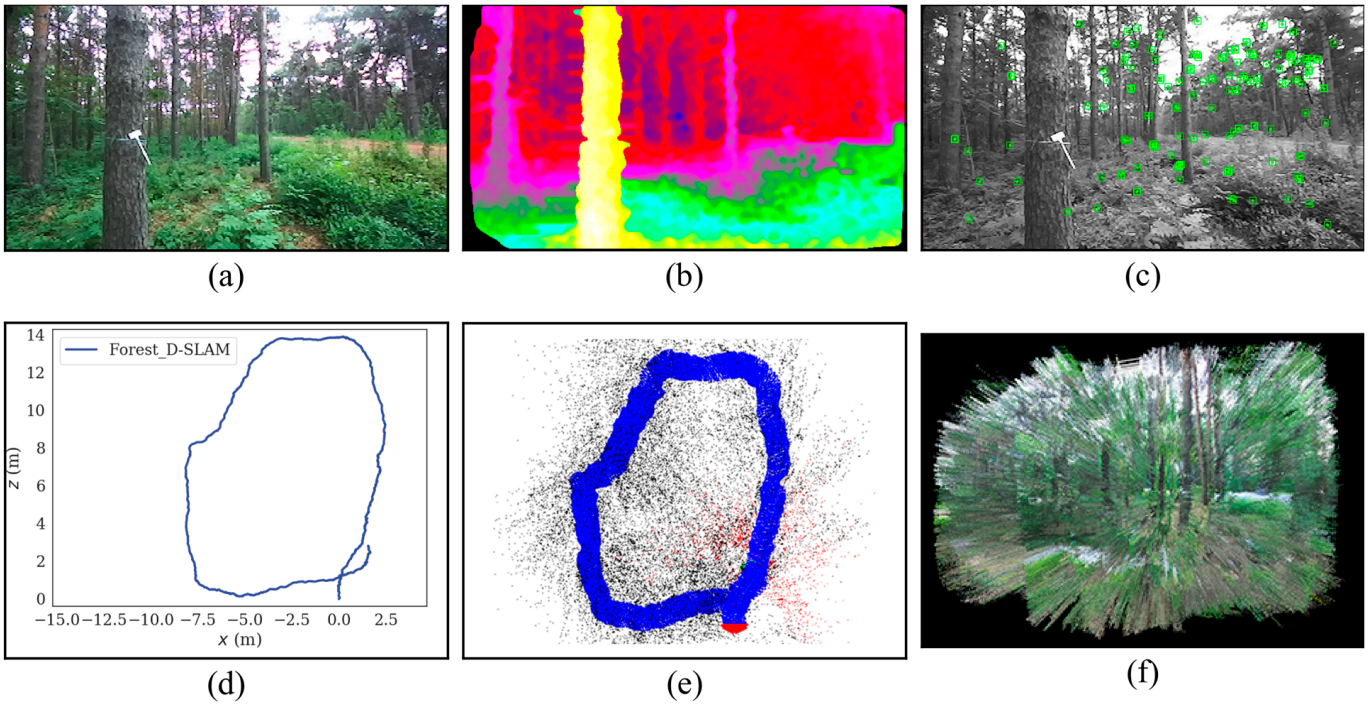

Figure 15. Dense mapping process on Forest, where (**a**) is the left RGB image, (**b**) is the visual disparity map, (**c**) is the feature point tracking map, (**d**) is the estimated trajectory map, which has a loop closing and the localization of the front end and the accuracy of the back end mapping are improved after loop closing correction, (**e**) is the sparse point cloud map, in which the blue boxes represent the keyframes, the green box represents the current frame, the red box represents the start frame, the red points represent the reference map points, and the black points represent all map points generated by keyframes. and (**f**) is the overall effect of the dense point cloud map.

Because Figure 15f shows the overall effect of the 3D point cloud of the forest scene from one perspective, the details from many perspectives are not visible, therefore, Figure 16 shows the details from different perspectives after being rotated, from which the structure of the forest scene can be clearly reproduced, including the density, poses and spatial position of the forest trees; the height, thickness, outline, color and texture of the tree trunks; the color and density of the leaves; the canopy; and the ground surface, which truly reflect the sample structure of the forest scene and provides an important basis for forestry exploration.

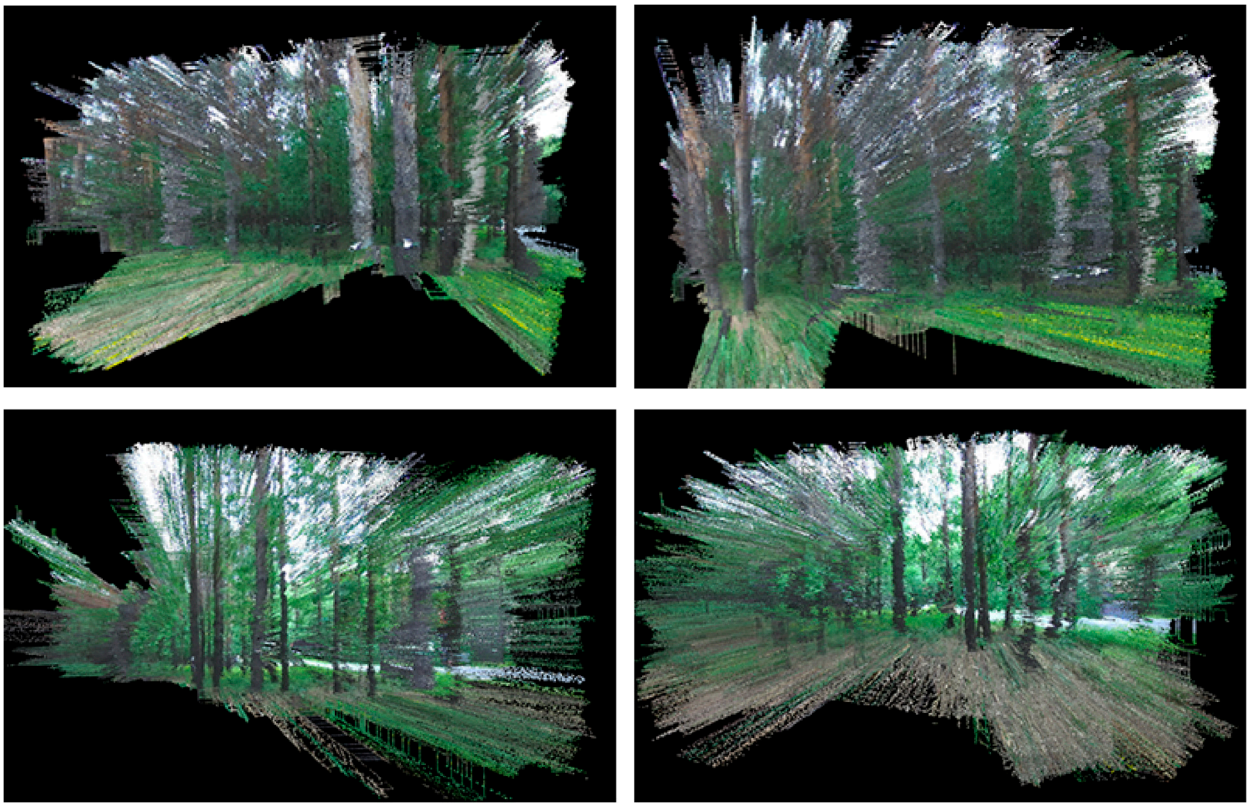

Figure 16. Localized dense point cloud at different angles with Forest.

6. Discussion

All tests of the D-SLAM system were run on a Dell G3 3590 portable computer, which was equipped with a ZED2 binocular camera capable of real-time localization and dense mapping in the forest scene, supportting both image dataset operation and real-time forest scene operation.

In terms of localization accuracy, D-SLAM can estimate the true scale of maps and trajectories without drift and achieve a high accuracy with an RMS ATE of 1.8 cm on EuRoC dataset (Table 4), outperforming international mainstream systems VINS Fusion, SVO2, and BASALT. Especially on the two challenging sequences V1_ 03_ difficult and V2_ 03_ difficult, ORB-SLAM2, VINS Fusion, and BASALT all failed to track, while the D-SLAM system adopted a binocular image spatial clue compensation strategy, which can use the right image to compensate for the lost field of view of the camera to a certain extent when the camera rotation angle is too large, so there was no tracking loss on difficult sequences. In addition, on the 11 sequences of KITTI, the D-SLAM system achieved relative RMSE with two average values of 0.64 and 0.2 for t_{rel} and R_{rel}, respectively (Table 5), which is superior to international mainstream SLAM systems ORB-SLAM2, LSD-SLAM, and PL-SLAM. Accordingly, D-SLAM is robust enough to be of great advantage in most cases. The error difference between various methods on the EuRoC and KITTI datasets is significant, which is directly related to the images in the dataset. EuRoC contains indoor small scene images with a small field of view distance, resulting in smaller errors, while KITTI is an outdoor highway large scene dataset with a larger field of view, fewer near points, and more far points, resulting in larger errors, even reaching tens of centimeters to several meters. In addition, the resolution, acquisition frequency, illumination, texture, and other factors of the image also have a significant impact on the error.

In terms of the real-time performance of the system, the dense mapping thread only processes keyframes, which does not affect the real-time performance of the system. From the frame rate and runtime of the tracking thread, it can be inferred that the D-SLAM system can run at a speed of 30 ordinary frames and 3 keyframes per second, fully meeting

the real-time requirements of SLAM system forest ecological environment localization and dense mapping.

In terms of dense mapping, the construction of real-time dense maps in three-dimensional space generally uses RGBD-SLAM or LiDAR-SLAM, which obtain depth information of the scene through depth sensors or LiDAR sensors. However, they are expensive and not conducive to the popularization and application in the industry. Moreover, depth sensors cannot be used outdoors, and laser sensors can only build sparse maps. Visual sensors can overcome these shortcomings and be applied to outdoor for dense mapping. Visual SLAM is generally applied to more regular outdoor scenes such as buildings, streets, roads, and parks. However, its application in complex forest environments has been rarely reported internationally. Therefore, dense mapping poses significant challenges in forest scenes with low texture, uneven lighting, and severe occlusion. Despite many disadvantages, the experimental results (Figure 16) show that it is possible to observe the structure of the forest ecological scenes, such as the density, pose, and spatial position of the tree, and the height, thickness, outline, color, and texture of the tree trunks, which meets the general needs of forest surveys and provides an important basis for forestry ecological exploration and forestry management. This work has innovation in both technology and application, providing important reference value for related research on forest digital twins.

In terms of image texture, KITTI is a dataset of urban and highway with highly textured image sequences. Its binocular images are high-definition images, with a baseline of 54 cm for binocular sensors, and the binocular images are rectified images with a resolution of 1240 $\times$ 376 pixels, therefore the generated disparity map has high accuracy and the dense point cloud map constructed is relatively clear. While the Forest dataset is collected by the ZED2 binocular camera with a baseline length of only 20 cm, which limits the accuracy of its disparity map. There are three types of image resolutions: HD1080:1920 $\times$ 1080, HD720:1280 $\times$ 720, and VGA: 672 $\times$ 376. High resolution images have more pixels and clearer texture features, which can improve the accuracy of localization and map construction; however, at the same time, it will increase the processing time of the front-end VO and the construction time of dense point cloud maps, which will slow down the overall running speed of the system. Moreover, high-resolution images have high performance requirements for hardware platforms such as computing speed and storage space, which are difficult to meet for general consumer level platform configurations. By balancing speed and accuracy, the VGA is chosen with the smallest resolution, which is equivalent to one-half of the KITTI resolution. Forest is a sequence of forest ecological images with low texture and large scenes, due to the low image resolution with VGA, the trunk features are not obvious, the similarity of the leaves and bushes is larger, the trees are severely obstructed, the light is unstable, the forest ecological environment scene is larger, and there are less nearpoints and more farpoints, resulting in the effect of the dense point cloud maps generated from Forest being not as clear as those of the KITTI dataset. However, it still meets the general needs of forest ecological surveys and forestry management.

This research utilizes visual images from binocular cameras to construct a three-dimensional forest map. However, visual sensors are generally affected by light, and the image quality collected under conditions of high exposure or low light is poor, which affects the effect of dense mapping. The system is almost unable to work at night, rainy days, and on snowy days. In addition, the camera's movement speed should not be too fast to prevent the system's processing speed from falling behind, and the camera's angle should not exceed 180 to avoid system tracking loss. When the angle of camera rotation $\omega \geq \frac{\pi}{2}$, the camera almost loses the perspective, at which time the feature points cannot be matched on the temporal clue causing the camera tracking to fail. The spatial clue compensation strategy is adopted: the previous frame of the right image of the spatial clue is used as the clue connection, and is inserted to compensate for the lost field of view of the temporal clue, continuing the tracking. When the camera rotation angle exceeds 90 degrees, the larger the angle, the greater the challenge. Due to uneven lighting, severe tree occlusion, large field of view, and fewer features in complex forest scenes, the imple-

mentation of the system poses significant challenges. The implementation of the system in real-world scenarios should involve drones equipped with binocular cameras and software and hardware platforms, which require lightweight processing. The performance of the platform also affects the system's running speed and dense mapping accuracy. If a high-performance platform can be configured and GPU acceleration can be used, it will further improve the system's running speed and dense mapping accuracy.

Because this research mainly focuses on forest ecological scenes below the canopy, the UAV flies under the canopy of the trees and collects data mainly on the trunks, branches, leaves, bushes, and forest grasses under the canopy, without including canopy information. In future research, satellite remote sensing technology can be combined to collect canopy information to construct broader and more comprehensive 3D forest ecological models, which provide powerful basis for fine surveying of forest resources, forest management, and forest rescue through visualized digital twins of forest environments.

7. Conclusions

This study explores the use of low-cost binocular cameras for the accurate 6-DoF pose estimation of UAVs in forest ecological spatial environment in a D-SLAM system, with a lightweight localization mode that uses only Tracking threads to track unmodeled areas to achieve zero drift. A dense mapping thread is added to construct dense point cloud maps of the forest ecological spatial environment. The amount of relative motion between frames and data association are used as constraints to filter keyframes, and a binocular image spatial clue compensation strategy is adopted to improve the robustness of tracking in adverse conditions such as large rotation, fast motion, and insufficient texture. Compared with the direct methods, the proposed approach can be used for wide-baseline feature matching, which is more suitable for 3D reconstruction scenes requiring high depth accuracy. The D-SLAM system runs at a speed of 30 ordinary frames and 3 keyframes per second, achieving location accuracy of several centimeters with the EuRoC dataset and a local t_{rel} average of 0.64m and R_{rel} average of 0.20 with the KITTI dataset, which outperform some mainstream sytems in terms of location accuracy and robustness, and have significant advantages in most cases. With a consumer-level computing platform, the system is able to work in real-time on the CPU, and the dense maps constructed can clearly reproduce the structure of the forest ecological interior scenes, meeting the requirements of the UAV's localization and mapping in terms of accuracy and speed. Moreover, the system is more reliable in the case of a signal blockage and can be a powerful complement and alternative solution to the current expensive commercial GNSS/Inertial Navigation System (INS) navigation systems. However, the system is greatly affected by light, and the location and dense mapping results are poor under conditions of high exposure or low light. In addition, the system will lose tracking when the camera moves too fast and the rotation angle is too large. In the future work, various sensors such as Inertial Measurement Unit (IMU) and LiDAR can be integrated to compensate for the limitations and shortcomings of the system. Neural networks can also be used to replace some or all modules of the system, solving the problem of limited system applications to a certain extent. In addition, it is possible to combine high-altitude remote sensing to capture broader forest images and construct a more comprehensive and extensive three-dimensional map of forest ecology.

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

Funding: This research was supported by the Fundamental Research Funds for the Central Universities (Grant No. 2572023CT15-03) and the National Natural Science Foundation of China (Grant No. 32271865).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Solares-Canal, A.; Alonso, L.; Picos, J.; Armesto, J. Automatic tree detection and attribute characterization using portable terrestrial lidar. *Trees* **2023**, *37*, 963–979. [CrossRef]
2. Gharineiat, Z.; Tarsha Kurdi, F.; Campbell, G. Review of automatic processing of topography and surface feature identification LiDAR data using machine learning techniques. *Remote Sens.* **2022**, *14*, 4685. [CrossRef]
3. Rijal, A.; Cristan, R.; Gallagher, T.; Narine, L.L.; Parajuli, M. Evaluating the feasibility and potential of unmanned aerial vehicles to monitor implementation of forestry best management practices in the coastal plain of the southeastern United States. *For. Ecol. Manag.* **2023**, *545*, 121280. [CrossRef]
4. Smith, R.C.; Cheeseman, P. On the Representation and Estimation of Spatial Uncertainty. *Int. J. Robot. Res.* **1986**, *5*, 56–68. [CrossRef]
5. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of SLAM. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
6. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
7. Servières, 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]
8. Zhang, J.; Singh, S. Loam: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems Conference, Berkeley, CA, USA, 14–16 July 2014. [CrossRef]
9. Khan, M.U.; Zaidi, S.A.A.; Ishtiaq, A.; Bukhari, S.U.R.; Farman, A. A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. In Proceedings of the Mohammad Ali Jinnah University Conference on Informatics and Computing, 2021 (MAJICC21), Karachi, Pakistan, 15–17 July 2021. [CrossRef]
10. Xu, M.; Lin, S.; Wang, J.; Chen, Z. A LiDAR SLAM System with Geometry Feature Group-Based Stable Feature Selection and Three-Stage Loop Closure Optimization. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 8504810. [CrossRef]
11. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense Tracking and Mapping in Real-Time. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, 6–13 November 2011. [CrossRef]
12. Engel, J.; Sturm, J.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 834–849. [CrossRef]
13. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [CrossRef]
14. Wang, R.; Schworer, M.; Cremers, D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3903–3911. [CrossRef]
15. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef]
16. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008; 20 September 2008. [CrossRef]
17. 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]
18. 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]
19. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Davidson, P. Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568. [CrossRef]
20. Dai, A.; Ritchie, D.; Bokeloh, M.; Reed, S.E.; Sturm, J.; Nießner, M. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.* **2017**, *36*, 1. [CrossRef]
21. Zhang, J.; Sui, W.; Wang, X.; Meng, W.; Zhu, H.; Zhang, Q. Deep Online Correction for Monocular Visual Odometry. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14396–14402. [CrossRef]
22. Li, S.; Wang, X.; Cao, Y.; Xue, F.; Yan, Z.; Zha, H. Self-supervised deep visual odometry with online adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition ((CVPR), Seattle, WA, USA, 13–19 June; 2020; pp. 6339–6348.
23. Li, S.; Wu, X.; Cao, Y.; Zha, H. Generalizing to the Open World: Deep Visual Odometry with Online Adaptation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13179–13188.

24. Zhang, Y.; Wu, Y.; Tong, K.; Chen, H.; Yuan, Y. Review of Visual Simultaneous Localization and Mapping Based on Deep Learning. *Remote Sens.* **2023**, *15*, 2740. [CrossRef]
25. Gao, X.; Zhang, T. *Visual SLAM Fourteen Lectures-From Theory to Practice*; Publishing House of Electronics Industry: Beijing, China, 2019; pp. 128–129, 184–185.
26. Zhang, H. *Robot SLAM Navigation*; China Machine Press: Beijing, China, 2022; pp. 292–293.
27. Liu, L.; Liu, Y.; Lv, Y.; Xing, J. LANet: Stereo matching network based on linear-attention mechanism for depth estimation optimization in 3D reconstruction of inter-forest scene. *Front. Plant Sci.* **2022**, *13*, 978564. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advance in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008. [CrossRef]
30. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016; pp. 223–225.
31. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4040–4048. [CrossRef]
32. Diederik, P.K.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015. [CrossRef]
33. Zbontar, J.; LeCun, Y. Computing the stereo matching cost with a convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7-12 June 2015. [CrossRef]
34. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 66–75. [CrossRef]
35. Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Qiao, L.; Chen, W.; Zhou, L.; Zhang, J. Learning deep correspondence through prior and posterior feature constancy. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
36. Pang, J.H.; Sun, W.X.; Ren, J.S.; Yang, C.; Yan, Q. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 887–895. [CrossRef]
37. Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; Jia, J. SegStereo: Exploiting Semantic Information for Disparity Estimation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018. [CrossRef]
38. Song, X.; Zhao, X.; Fang, L.; Hu, H. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *Int. J. Comput. Vis.* **2020**, *128*, 910–930. [CrossRef]
39. Chang, J.R.; Chen, Y.S. Pyramid stereo matching network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
40. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
41. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580. [CrossRef]
42. 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.
43. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multi-camera systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [CrossRef]
44. Cremers, D.; Schubert, D.; Stückler, J.; Demmel, N.; Usenko, V. Visual-Inertial Mapping with Non-Linear Factor Recovery. *IEEE Robot. Autom. Lett.* **2019**, *5*, 422–429. [CrossRef]
45. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
46. Engel, J.; Stueckler, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015. [CrossRef]
47. Gomez-Ojeda, R.; Moreno, F.A.; Zuñiga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. A Stereo SLAM System Through the Combination of Points and Line Segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [CrossRef]