Computers and Artificial Intelligence；Vol.3, No.2; 2026
E-ISSN: 3006 -3728 P-ISSN：3006-371X
DOI: https://doi.org/10.70267/cai.26v3n2.1327
Published by: Zeus Press

Construction of a Dense Mapping System for Stereo Vision SLAM Based on Point-Line Fusion

Lanqing Zhang and Lili Yuan*

Henan Polytechnic University, Jiaozuo, Henan, China

**Corresponding author: Lili Yuan*

Abstract

To address issues such as insufficient feature extraction, inaccurate localization, and sparse point clouds encountered by visual SLAM systems in low-texture, dynamic environments, this paper proposes improvements to PLCD-SLAM. It introduces dual hardware-software constraints for depth estimation and presents a novel dense mapping architecture for stereo visual SLAM based on point-line fusion. Leveraging the robust Superpoint feature extractor and the strong structural capabilities of end-to-end wireframe parsing, the system compensates for feature information loss and enhances pose estimation accuracy. During depth computation, stereo matching combined with dual soft-hard constraints optimizes depth value precision, enabling high-accuracy dense mapping. To validate the improved algorithm's performance, comparative experiments were conducted on the KITTI, EUROC, and TUM (converted to stereo) public datasets. Results demonstrate that the proposed method achieves significantly lower absolute trajectory error (RMSE) than mainstream SLAM algorithms. It substantially enhances the system's robustness and positioning accuracy in dynamic and complex environments while producing high-quality dense maps.

Keywords

visual SLAM, point-line fusion, wireframe parsing, desnse mapping

1. Introduction

Simultaneous Localization and Mapping (SLAM) has become a core technological foundation for autonomous systems such as mobile robots, autonomous vehicles, and augmented reality devices, enabling these systems to perceive their environment, estimate their position and orientation, and construct spatial models in unseen scenes. Among various sensor configurations, stereo vision systems have garnered significant attention due to their advantages—low cost, passive sensing, and the ability to directly estimate depth through stereo matching—making them the preferred solution for large-scale outdoor and complex indoor applications. As SLAM technology expands its applications, downstream tasks like autonomous navigation, obstacle avoidance, and scene reconstruction demand higher mapping quality. Dense point cloud maps, capable of accurately representing environmental details, are gradually replacing sparse feature maps to meet the need for fine-grained environmental interaction.

However, real-world application scenarios often pose significant challenges to stereo SLAM systems. In low-texture environments (e.g., white walls, deserts) and dynamic scenes, traditional point-feature extraction methods like ORB and SIFT frequently suffer from insufficient feature points, low matching accuracy, or even tracking loss [1]. These issues directly lead to cumulative pose estimation errors, resulting in sparse and incomplete dense maps that severely compromise system robustness and reliability. Although RGB-D cameras can directly capture depth information, limitations such as short effective range and sensitivity to ambient lighting make them difficult to adapt to large-scale outdoor scenes. Simultaneously, pure point-based dense mapping methods heavily rely on texture information, experiencing significant performance degradation when feature information is scarce [2].

To address these limitations, researchers have turned to multi-feature fusion strategies, with point-line fusion emerging as a highly promising solution. As inherent structural elements in artificial environments, line features provide stable geometric constraints even in low-texture regions [3]. Compared to noise-prone point features, line features exhibit greater invariance to lighting variations and viewpoint changes, effectively compensating for missing point feature information. Early point-line fusion SLAM methods (e.g., PL-SLAM) incorporated line segments extracted via LSD or EDLines into traditional point-based frameworks, moderately enhancing pose estimation stability. However, these methods still face limitations: line feature extraction is prone to short lines and broken lines, potentially leading to “over-extraction” or “mis-extraction” under varying lighting conditions; moreover, depth estimation lacks effective constraint mechanisms, resulting in low-accuracy dense maps that fail to fully leverage the complementary advantages of point and line features.

In recent years, advances in deep learning have revitalized stereo SLAM. Feature extractors like SuperPoint enable end-to-end robust feature detection, reliably extracting discriminative points even in complex environments. Concurrently, end-to-end line-based parsing algorithms enhance the structural expressiveness of line features, providing high-quality segment inputs for multi-feature fusion. On the other hand, dense mapping techniques based on stereo matching have been continuously optimized. Multi-scale feature pyramids and adaptive feature selection mechanisms have improved the utilization efficiency of feature information, while the introduction of hard and soft constraints has further enhanced the accuracy and reliability of depth estimation. However, existing approaches rarely integrate these techniques organically: most point-line fusion frameworks still rely on traditional feature extractors, and the depth estimation process lacks specialized, effective constraints tailored for point-line fusion. Consequently, their performance in complex scenes falls short of optimal.

To address critical challenges in stereo dense mapping under low-texture and varying illumination conditions—such as insufficient feature extraction, inaccurate pose estimation, and sparse point clouds—this paper proposes an improved stereo visual SLAM dense mapping system based on dual constraints: point-line fusion and depth estimation. This system inherits and reuses core designs validated in PLCD-SLAM: 1) It fuses the robust feature extraction capability of SuperPoint with the structural representation advantages of end-to-end wireframe parsing, achieving complementary fusion of point and line features; 2) It employs a multi-scale feature pyramid and adaptive feature selection mechanism to mitigate feature information loss in complex environments. Building upon this foundation, the core innovation and improvement focus on depth estimation: 3) The system innovatively introduces hard and soft constraints during stereo matching and depth computation. This dual-constraint mechanism enhances depth estimation accuracy and consistency while ensuring dense map completeness and detail richness. Experimental validation on the KITTI, EUROC, and TUM datasets demonstrates that compared to mainstream SLAM algorithms and PLCD-SLAM, the proposed system achieves further reduction in absolute trajectory error (RMSE), significantly enhanced robustness and accuracy in dynamic complex environments, and generates higher-quality dense point cloud maps.

2. Research Hypotheses

Dense mapping based on stereo vision has emerged as a research hotspot in the SLAM field due to its advantages such as low cost and direct depth estimation. Significant progress has been made in algorithmic frameworks, feature representations, and depth optimization. This section reviews relevant work across three areas: point-line fusion SLAM, monocular dense mapping, and stereo dense mapping.

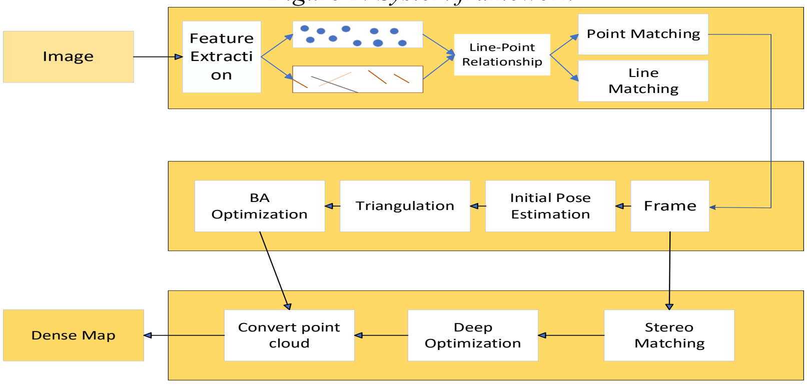
2.1 Point-line Fusion SLAM

To address the issue of insufficient point features in complex environments, point-line fusion has emerged as an effective technical approach. Early on, Gonzalez-Jimenez et al. [5] proposed PL-SLAM, which introduced line segments extracted via LSD or EDLines into traditional point-based frameworks, thereby enhancing the stability of pose estimation. Xu et al. [4] proposed AIRSLAM, a point-line association SLAM system for dynamic environments. This system combines the robust feature extraction capabilities of SuperPoint [6] with the structural representation advantages of end-to-end line frame parsing. It incorporates a multi-scale feature pyramid and an adaptive feature selection mechanism to mitigate feature information loss in complex environments. Building upon this, Hang et al. [14] proposed a point-line stereo SLAM method based on an improved ELSED line detection algorithm. This approach enhances line feature quality through short segment merging and masked feature homogeneity optimization, while implementing a bidirectional consistency detection mechanism to improve matching accuracy. These improvements significantly boost the system's robustness in low-texture and dynamic lighting conditions. Regarding backend optimization, related work typically extends the PnPL formula for line features to construct a joint optimization function, incorporating the reprojection error of both point and line features into the optimization objective to improve pose estimation accuracy.

2.2 Monocular Dense SLAM

Although this paper focuses on stereo dense mapping, monocular dense SLAM-related work provides important references in areas such as dense depth estimation and real-time optimization. Gallagher et al. [21] proposed the first fully dense SLAM system quantitatively evaluated in autonomous driving scenarios, which fuses sparse feature tracking with dense depth prediction to enhance the scale-aware capability of monocular depth estimation. Wimbauer et al. [9] introduced the semi-supervised monocular dense reconstruction framework MonoRec, which employs a MaskModule to handle moving objects in dynamic environments, achieving high-precision depth estimation for both static and dynamic objects. In recent years, implicit representation-based methods have advanced rapidly. Liu et al. [8] proposed SLAM3R, an end-to-end monocular real-time dense reconstruction framework that generates globally consistent 3D point clouds directly from RGB video without explicit camera parameter estimation. Chi et al. [9] introduced GS-SLAM, the first dense SLAM system incorporating 3D Gaussian splatting. By employing an adaptive Gaussian expansion strategy, it achieves high-fidelity real-time rendering and high-precision mapping, offering novel insights for dense map representation and optimization.

Figure 1: System framework



2.3 Dense Stereo Mapping Method

Stereo dense mapping methods are primarily categorized into direct methods and feature-based methods, with ongoing innovations in integrating deep learning to enhance performance.

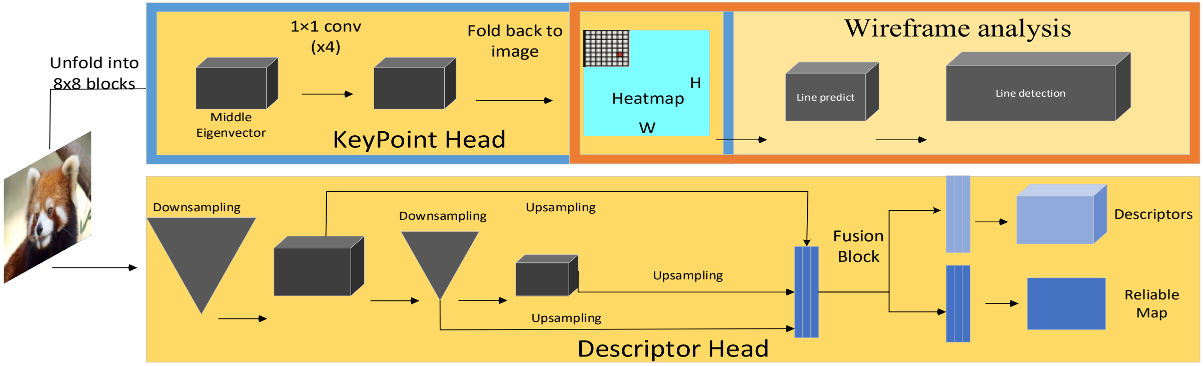
Direct methods demonstrate unique advantages in low-texture environments by optimizing photometric error to avoid reliance on sparse feature matching. Engel et al. [10] proposed Stereo LSD-SLAM, the first stereo direct SLAM system capable of real-time operation on CPUs. This method combines static stereo and temporal multi-view stereo cues to estimate high-gradient pixel depths, outputting a semi-dense depth map that lays the foundation for direct stereo dense mapping. Building upon this, Wang et al. [11] extended DSO to a stereo variant (Stereo DSO), integrating stereo constraints into the direct optimization to reduce monocular scale drift and enhance depth estimation accuracy, making it suitable for high-motion, large-scale scenes. For higher accuracy demands, Teed et al. [14] proposed DROID-SLAM. By organically integrating depth networks with dense BA through a differentiable dense BA layer, it directly optimizes depth at each pixel. This end-to-end optimization method overcomes the depth basis limitations of traditional approaches, enabling high-precision dense mapping across multiple sensors, including stereo cameras.

Feature-based methods are widely adopted in engineering practice due to their robust stability. Mur-Artal et al. [13] proposed the classic feature-based SLAM framework ORB-SLAM2, which natively supports stereo cameras. Although it defaults to sparse mapping, dense mapping can be achieved by extending the stereo depth fusion module, making it a commonly used foundational framework for engineering development. To adapt to complex outdoor environments, Xue et al. [16] improved ORB-SLAM2 by incorporating adaptive threshold feature extraction and a stereo depth fusion module, effectively addressing feature loss and point cloud sparsity in shaded orchard settings. Wang et al. [16] further integrated a cross-attention stereo matching model into the ORB-SLAM3 framework, significantly enhancing stereo matching accuracy and producing high-precision color depth point clouds. Over 90% of dense point cloud errors remained within 0.5m, making it suitable for outdoor autonomous navigation scenarios.

Recently, the integration of deep learning has emerged as a key direction for enhancing the performance of dense stereo mapping. Reference [17] proposes a deep learning-assisted high-resolution stereo depth reconstruction method, utilizing deep learning to generate initial disparity maps that guide the SGBM algorithm. This approach balances the accuracy and efficiency of stereo matching while reducing computational complexity. Koestler et al. [18] introduced TANDEM, a real-time monocular dense SLAM framework that integrates direct visual odometry with depth multi-view stereo (MVS). Although designed for monocular cameras, its dense depth prediction based on CVA-MVSNet and TSDF voxel mesh fusion strategy provide valuable insights for binocular dense mapping.

Existing stereo dense mapping methods have made significant progress in direct optimization, feature matching, and deep learning fusion. Point-line fusion strategies have effectively enhanced the system's robustness in low-texture environments. However, current point-line fusion SLAM still faces bottlenecks in depth estimation: traditional methods lack effective constraint mechanisms during depth computation, while deep learning-based approaches rarely design targeted optimization strategies leveraging the complementary properties of point and line features. Although prior work has achieved effective fusion of point and line features and improved feature utilization efficiency, there remains room for improvement in depth estimation accuracy and dense map completeness. Therefore, this paper focuses on the depth estimation stage of point-line fusion stereo dense mapping, introducing a dual mechanism of hard and soft constraints to further enhance stereo matching accuracy and dense map quality.

Figure 2: Point-line fusion framework



3. Research Design

The proposed point-line fusion stereo dense mapping SLAM system primarily consists of three components: “feature extraction,” “pose estimation,” and “dense mapping.” It ensures pose robustness through complementary constraints between point and line features while achieving high-precision dense map construction by leveraging soft and hard dual-constraint depth computation. The overall framework is illustrated in Figure 1.

3.1 Point-Line Fusion Module

Image data is first input into the feature extraction module: SuperPoint is used to extract robust point features, while an end-to-end wireframe parsing algorithm extracts structured line features. Geometric relationships between points and lines within the same frame are established (e.g., point affiliation on line segments). See Figure 2 for details.

Subsequently, LightGlue achieved “single matching for dual purposes” through “reusing point matching results.” Specifically, after LightGlue completes point matching, the output matching pairs are used both for triangulating 3D points and calculating initial poses. Simultaneously, by applying the point-line association formula 1, these matching pairs are directly leveraged to derive corresponding line matches—eliminating the need for separate feature extraction and matching for lines.

$$\mathrm{d}_{i\mathrm{j}} = \frac{|A_{\mathrm{j}}.x_i + B_j y_i + C_j|}{\sqrt{A_j^2 + B_j^2}} \tag{1}$$

The corresponding points and lines here are $P(x_i, y_i)$ and $A_{\mathrm{j}}x + B_j y + C_i = 0$. Their relationship is determined by the distance from the point to the line segment and the projection range constraint.

3.2 Position Estimation Module

Matching-Based Point-Line Feature Input Pose Estimation Module: First, keyframes are selected based on matching results. Initial pose estimation is then performed using these selected keyframes, generating initial map points through triangulation. Finally, BA optimization is introduced, incorporating point reprojection errors and line endpoint/direction constraints into the objective function. This optimizes camera pose and map point parameters, outputting high-precision camera trajectories and sparse map priors.

3.2.1 Initial Pose Estimation

For initial pose estimation in stereo SLAM, this paper employs stereoscopic matching disparity and essential matrix decomposition. The core approach involves directly solving interframe rotation and translation through joint optimization of point-line features and geometric constraints from stereo cameras. Camera intrinsic parameters K and polar lines B are obtained via left-right polar line correction of images and camera calibration.

The initial pose is obtained through essential matrix decomposition: based on normalized matched point coordinates, the essential matrix constraint equation E is constructed (as shown in Equation 2).

$$x_{R,i}^T E x_{L,i} = 0 \tag{2}$$

The normalized coordinates for the right eye match points are $x_{R,i}$, and for the left eye match points are $x_{L,i}$. Subsequently, the 8-point algorithm combined with SVD decomposition is employed to solve the intrinsic matrix, with singular values corrected via rank-2 constraints to ensure compliance with rigid body motion geometry.

Subsequently, the essential matrix decomposition yields four sets of pose candidate solutions. The optimal solution with positive depth R, t, and $Z_{\mathrm{i,calc}}$ is retained—specifically, the unique valid solution where all 3D points lie in front of the camera. This process yields the initial rotation matrix and unit-scale translation vector.

After extreme line calibration, true depth is computed via horizontal parallax between left and right eye matching points (Formula 3), and a scale factor is calculated using binocular matching points (Formula 4). The unit scale translation vector is calibrated to physical scale (Formula 5), achieving scale alignment of pose with accuracy surpassing motion-recovery scale methods in monocular SLAM.

$$Z_{\mathrm{i}} = \frac{f_x B}{d_i} \tag{3}$$

$$s = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{Z_i}{Z_{i,calc}} \tag{4}$$

$$t_{real} = s \cdot t \tag{5}$$

Where f_x is the focal length, $d_i = u_{L,i} - u_{R,i}$. $p_{L,i} = (u_{L,i}, v_{L,i}, 1)^T$ is the homogeneous pixel coordinate of the left eye, and is the number of binocular matching points.

Finally, the system constructs a joint point-line optimization objective function (Equation 6), incorporating the normalized reprojection error of point features and the endpoint projection error of line features (Equation 7) into the optimization. The pose parameters are iteratively optimized using the Levenberg-Marquardt algorithm. After optimization convergence, the initial results are validated through reprojection error thresholds (point error < 2 pixels, line error < 3 pixels) to ensure validity, providing high-precision pose priors for subsequent local BA optimization and dense mapping.

$$\min_{R,t}\left(\sum_{i=1}^{N}\| x_{R,i} - Rx_{L,i} - t \|^2 + \lambda\sum_{j=1}^{M}\| r_{line,j} \|^2\right) \tag{6}$$

$$r_{line,j} = \begin{bmatrix} d(\hat{p}_{R,j1}, l_{R,j}) \\ d(\hat{p}_{R,j2}, l_{R,j}) \end{bmatrix} \tag{7}$$

where $\hat{p}_{R,j1} = RK^{-1}p_{L,j1} + t, \hat{p}_{R,j2} = RK^{-1}p_{L,j2}$ represents the predicted coordinates of the left eye line segment endpoint after pose transformation, and $l_{R,j}$ denotes the linear equation parameters of the jth line segment for the right eye.

3.2.2 Triangulation

In the triangulation process of point features from a stereo camera, the geometric properties after polar line correction simplify the solution of 3D point coordinates into a direct mapping of disparity and depth. Its core principle leverages the baseline constraints and similarity triangle theory of the stereo system to achieve efficient, true-scale 3D coordinate computation.

With initialized pose estimation, parameters such as R, t, K, and B are known. The disparity obtained via Equation 4 enables triangulation of point features:

$$X = \frac{(\mathrm{u} - c_x)Z}{f_x}, Y = \frac{(v - c_y)Z}{f_y} \tag{8}$$

Among them c_x, c_y, f_x, f_x and are parameters in K, representing the main point coordinates and focal length, respectively.

In line feature processing for stereo SLAM, the representation and triangulation of 3D lines must balance geometric completeness with parametric simplicity. The core approach involves achieving a compact representation of 3D lines through Plücker coordinates and completing line triangulation based on the polar line constraints of stereo-matched line segments. To reduce parameterization, this paper converts Plücker coordinates to an orthogonal representation.

$$(U, W) \in SO(3) \times SO(2) : L = \left[n \mid v\right] = \begin{bmatrix} \frac{n}{\|n\|} & \frac{v}{\|v\|} & \frac{n \times v}{\|n \times v\|} \end{bmatrix}\begin{bmatrix} \|n\| & 0 \\ 0 & \|v\| \\ 0 & 0 \end{bmatrix} \tag{9}$$

$$W = \frac{1}{\sqrt{\|n\|^2 + \|v\|^2}}\begin{bmatrix} \|n\| & -\|v\| \\ \|v\| & \|n\| \end{bmatrix} \in SO(2) \tag{10}$$

Here, v is the unit direction vector of the 3D line, n is the normal vector from the origin to the plane containing the line, U is the rotation matrix for the 3D line, and W is the rotation matrix for the 2D line.

If the above method fails, the feature can be constructed directly using the triangulated points, as shown in Formula 11.

$$L = \begin{bmatrix} n \\ v \end{bmatrix} = \begin{bmatrix} X_1 \times X_2 \\ \dfrac{X_1 - X_2}{\| X_1 - X_2 \|} \end{bmatrix} \tag{11}$$

Among these, X_1, X_2two 3D points belonging to the same 2D line segment satisfy the condition that the distance from the image plane to the line segment is the shortest.

3.2.3 BA Optimization (Re-projection Error)

For the 3D point X_p observed in the i-th frame, its reprojection error is the Euclidean distance between the "observed 2D point" and the "predicted 2D point" projected from the 3D point onto the image.

$$r_{i,X_p} = \hat{x}_{i,p} - \pi(R_{cw}X_p + t_{cw}) \tag{12}$$

Here, $\hat{x}_{i,p}$ represents the observed 2D coordinates of the 3D point in frame i, $\pi(\cdot)$ denotes the projection function, and R_{cw}, t_{cw} are the camera poses to be optimized.

3D-line reprojection calculates 2D line segments from 3D lines onto the image plane and is used to optimize residuals. It primarily consists of two steps: coordinate system transformation and image projection. The 3D line in the world coordinate system $L_w = (n_w, v_w)$ is transformed into the camera coordinate system $L_C = (n_c, v_c)$.

$$L_c = \begin{bmatrix} n_c \\ v_c \end{bmatrix} = \begin{bmatrix} R_{cw} & t_{cw}{}^{\wedge} R_{cw} \\ 0 & R_{cw} \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix} = H_{cw} L_w \tag{13}$$

Among these, R_{cw}, t_{cw}, H_{cw}, $t_{cw}{}^{\wedge}$ correspond respectively to the rotation matrix from the world coordinate system to the camera coordinate system, the translation vector from the world coordinate system to the camera coordinate system, the transformation matrix of the 3D line, and the antisymmetric matrix of the translation vector. After transforming the coordinate system, the 3D line in the camera coordinate system is projected onto a 2D line $l = (A, B, C)$ segment in the image plane (with the equation Ax + By + C = 0).

$$l = \begin{bmatrix} A \\ B \\ C \end{bmatrix} = P_c L_{[:3]} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} n_c \tag{14}$$

Among these, the line projection matrix (3×3) P_c is composed of the camera's intrinsic parameters and $L_{[:3]}$ is defined in the camera coordinate system n_c. The error function is given by Equation 16.

$$\begin{aligned} r_{i,L_q} &= e_l(\hat{l}_{i,q}, P_c(H_{cw}L_q)_{[:3]}) \\ e_l(\hat{l}_{i,q}, l_{i,q}) &= \begin{bmatrix} d(\hat{p}_{i,q1}, l_{i,q}) & d(\hat{p}_{i,q2}, l_{i,q}) \end{bmatrix}^T \end{aligned} \tag{15}$$

$l_{i,q}$ is the predicted 2D line segment of the 3D line projected onto the i-th frame, $\hat{l}_{i,q}$ is the observed 2D line segment of the 3D line in the i-th frame, $\hat{p}_{i,q1}$, $\hat{p}_{i,q2}$ are its endpoints, where $d(\cdot)$ is the distance from point to the line segment.

Finally, this paper constructs a joint optimization objective function that integrates point reprojection error and line projection error, with the specific form as follows:

$$E=\sum\| r_{i,X_p} \|_{\Sigma_P}^2+\sum\| r_{i,L_q} \|_{\Sigma_L}^2 \tag{16}$$

Among these, $\sum_P$ and $\sum_L$ represent the covariance matrices for point and line residuals, respectively. This objective function is iteratively minimized using the Levenberg-Marquardt algorithm, enabling efficient convergence to the optimal solution. It simultaneously optimizes camera pose and 3D point/line coordinates, preserving the local detail constraints of point features while leveraging the global structural stability of line features to compensate for the limitations of point features. This approach ultimately enhances the system's pose estimation accuracy in complex scenarios involving low-resolution textures and varying lighting conditions. line coordinates. In complex scenes with low texture and varying lighting, this approach preserves the local detail constraints of point features while leveraging the global structural stability of line features to compensate for point feature deficiencies. Ultimately, it enhances the robustness of system pose estimation and the accuracy of dense mapping.

3.3 Dense Map Building Module

The innovative breakthrough of this module lies in its multi-layer collaborative mechanism—combining "LightGlue geometric priors + SGM stereo matching for dense disparity + photometric consistency constraints"—to establish a more reliable dense mapping approach: LightGlue's sparse feature matching provides global geometric constraints, anchoring the "global orientation" for subsequent SGM disparity estimation; SGM then achieves pixel-level initial disparity coverage; finally, photometric consistency loss completes local error correction. This layered logic—"global constraints first, dense coverage second, local optimization last"—overcomes the limitations of single constraints while significantly enhancing the algorithm's adaptability to complex scenes, all while ensuring reconstruction accuracy.Specifically, the execution logic of this workflow can be divided into three key stages.

3.3.1 Construction of Geometric Priors

This section employs LightGlue to efficiently match local features in stereo images. By obtaining sparse feature correspondence pairs and applying the RANSAC algorithm to eliminate mismatches, a high-precision set of matched pairs S is derived. Subsequently, the intrinsic matrix E is solved to define the global geometric range for subsequent SGM disparity searches, preventing SGM from getting stuck in local optima.

3.3.2 Construction of Geometric Priors

This section combines SGM stereo matching with dual soft and hard constraints for collaborative optimization. First, the essential matrix decomposition products R and t obtained from preceding steps are treated as hard constraints. Based on the imaging model of the stereo camera and the inter-camera pose relationship represented by R and t, reasonable disparity intervals for different pixel positions can be derived. This strictly limits the disparity search range [d_min, d_max] for SGM. This hard constraint fundamentally eliminates erroneous disparity values beyond geometrically plausible ranges, significantly reducing SGM computational complexity while preventing the algorithm from getting stuck in local optima within invalid disparity ranges. Its core relationship is the disparity-to-depth conversion shown in Equation 3.

Next, soft constraints are introduced through Census transform-based matching cost calculation and multi-directional cost aggregation to enhance pixel-level matching reliability. SGM's core soft constraint logic is implemented via a cost function and aggregation strategy, detailed as follows:

Initial matching cost is calculated as the Hamming distance between corresponding pixel neighborhood features in left and right images, effectively improving robustness to lighting variations. The cost function formula is:

$$\mathrm{C}_{\mathrm{init}}\left(\mathrm{x},\mathrm{y},\mathrm{d}\right)=\mathrm{Hamming}\left(\mathrm{T}_{\mathrm{l}}\left(\mathrm{x},\mathrm{y}\right),\mathrm{T}_{\mathrm{r}}\left(\mathrm{x}-\mathrm{d},\mathrm{y}\right)\right) \tag{17}$$

Among these, $T_l(x,y)$, $T_r(x-d,y)$represent the Census transform encodings of the left and right eye images at positions (x, y) and (x-d, y), respectively. Hamming$(\cdot)$ denotes the Hamming distance calculation formula.

To address the unreliability of initial cost values in sparsely textured regions, SGM performs cost aggregation across eight directions. During aggregation, the final aggregated cost is obtained by weighting and summing the initial costs of neighboring pixels. If pixel (u, v) belongs to the matching point pair set S, then its true disparity d^* is cost-weighted:

$$\mathrm{C_{weight}}(\mathrm{u,v,d}) = \{ \begin{matrix} \mathrm{C_{init}}(\mathrm{u,v,d^*}) \times \mathrm{w_{low}} & (\mathrm{d} = \mathrm{d^*}) \\ \mathrm{C_{init}}(\mathrm{u,v,d}) \times \mathrm{w_{high}} & (\mathrm{d} \neq \mathrm{d^*}) \end{matrix} \tag{18}$$

Among these, w_{low} is the low weight, aiming to reduce the cost of true disparity, typically set to 0.1. w_{high} is the high weight, aiming to increase the cost of false disparity, typically set to 2. When a pixel (u,v) is within the neighborhood of a constraint point (within a 3×3 window), the cost of d^* is slightly increased to encourage the neighborhood's disparity to converge toward the true value. If, during aggregation, the cumulative cost corresponding to a constraint point's disparity significantly exceeds that of other disparities (), the cumulative cost for that point is forcibly adjusted to the minimum value, ensuring its disparity is correctly selected.

Finally, perform constraint filtering for left-right consistency checks. If the right-image disparity corresponding to a constraint point differs from d^*, mark that region as a “high-confidence region” and skip conventional outlier removal. Replace constraint point locations in the disparity map directly with d^* to obtain the initial disparity map d_{init}. Derive the initial depth map using Equation 3.

3.3.3 Construction of Geometric Priors

In single-frame luminance optimization, images from the left and right eyes are captured at the same moment, exhibiting stronger luminance consistency. The disparity map is first optimized by minimizing luminance loss between the left and right eyes. If the pixel value for the left eye is $p_L(u,v)$ and its corresponding pixel for the right eye is $p_R(u-d,v)$, the luminance loss is calculated as:

$$\mathrm{E_{photo}}(\mathrm{p_L}) = \| \mathrm{I_{L,t}}(\mathrm{p_L}) - \mathrm{I_{R,t}}(\mathrm{p_R}) \|_2^2 + \lambda \| \nabla \mathrm{I_{L,t}}(\mathrm{p_L}) - \nabla \mathrm{I_{R,t}}(\mathrm{p_R}) \|_2^2 \tag{19}$$

Here, $I_{L,t}(p_L)$ and $I_{R,t}(p_R)$ represent the left and right images, ∇ denotes the gradient operator, λ signifies the gradient loss weight. Subsequently, a disparity smoothing loss is incorporated to maintain smooth disparity transitions between adjacent pixels while preserving image boundaries, thereby enhancing the plausibility and visual consistency of the results.

$$\mathrm{E_{smooth}}(\mathrm{p_L}) = \sum_{\mathrm{q} \in \mathrm{N}(\mathrm{p_L})} \| \mathrm{d}(\mathrm{p_L}) - \mathrm{d}(\mathrm{q}) \|_2^2 \cdot \mathrm{e}^{-\beta \| \mathrm{I_{L,t}}(\mathrm{p_L}) - \mathrm{I_{L,t}}(\mathrm{q}) \|_2^2} \tag{20}$$

Here, $N(p_L)$ denotes the 4-neighborhood of p_L, β is the weighting factor, and $\| d(p_L) - d(q) \|_2^2$ measures the disparity difference between pixel p_L and pixel q within its neighborhood $N(p_L)$. Therefore, the joint optimization function for single-frame photometric optimization is:

$$\begin{gathered} E_{total} = \alpha_1 E_{sparse} + \alpha_2 E_{photo} + \alpha_3 E_{smooth} \\ E_{sparse = \sum_{p \in S} \| d(p) - d_i^* \|_2^2} \end{gathered} \tag{21}$$

It comprises three components: the LightGlue sparse constraint loss, photometric loss, and smoothing loss. The weights α_1, α_2, and α_3 for each loss are set to 10, 1, and 0.1, respectively. By minimizing E_{total} using the Levenberg -Marquar -dt (LM) algorithm, the disparity map d_{opt1} is updated.

Subsequently, leveraging luminance consistency between keyframes, depth and pose are further optimized to eliminate single-frame noise. Convert the depth map into a point cloud by transforming the optimized disparity map into a 3D point cloud P = {(X, Y, Z)} using Equation 8. Next, project the point cloud P onto the reference frame via the initial pose T_{init} to obtain the projected pixels $p_s = \pi(T_{init} \cdot P)$. Calculate the photometric loss $E_{photo-s}$ between the reference frame and the keyframe using Equation 19. Thus, the joint function for multi-frame photometric optimization is:

$$E_{total=s} = \alpha_2 E_{photo-s} + \alpha_3 E_{smooth} + \alpha_4 E_{reg} \tag{22}$$

Among these, E_{reg} represents the depth regularization term. During optimization, depth and pose are fixed separately to minimize photometric error, yielding the optimized depth map Z_{opt} and pose estimate T_{opt}.

Finally, based on the optimized depth map and pose estimation results, the data is transformed into the world coordinate system $P_w = T_{opt} \cdot P$ to obtain world coordinates. Subsequently, the ICP algorithm is employed to complete the registration of the current frame's point cloud with the global map's point cloud. The registered point cloud is then integrated into the global map, synchronously updating the map data.

Through this comprehensive workflow, the stereo SLAM system outputs dense depth maps that balance detail and reliability, providing a high-precision geometric foundation for subsequent scene modeling and map construction. Particularly in complex environments with numerous dynamic objects and sparse textures, the multi-layered constraint design ensures more stable dense mapping results, better supporting the positioning and navigation requirements of the SLAM system.

4. Empirical Analysis

All experiments in this paper were conducted on a single computer configured with 32GB of RAM, an Intel Core i9-14900HX CPU, and an NVIDIA GeForce RTX 4060 Laptop GPU, running Ubuntu 20.04 as the algorithm execution environment. The EuRoc [20], KITTI [21], and TUM [22] datasets were selected to evaluate the system's dense mapping performance in dynamic environments with low texture density. To ensure experimental stability, each sequence was tested repeatedly. Common evaluation metrics were employed: Absolute Trajectory Error (ATE), Relative Position Error (RPE), trajectory comparison plots, and dense point cloud maps. ATE measures the global deviation between the estimated and true trajectories, suitable for assessing cumulative error after prolonged SLAM operation. RPE represents the error in pose changes between adjacent frames, evaluating the system's drift over time. In the tables, RMSE values indicate performance, where lower values denote superior performance. “F” indicates the system failed to complete the sequence, and bolded text signifies the system achieving the best performance in that sequence. Experimental results were evaluated using the Evaluation of Visual Odometry and SLAM (EVO) software.

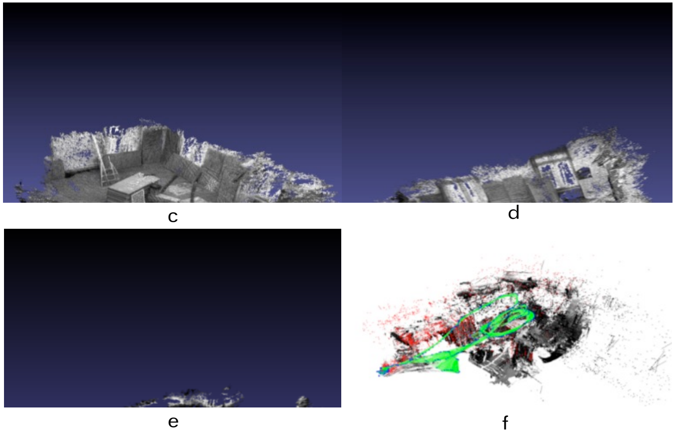
First, to validate performance in static, high-texture, and low-texture environments, experiments were conducted using the EuRoc dataset. Relevant experimental data has been previously reported. Testing was performed on 9 sequences from the EuRoc dataset. Results are shown in Table 1.

Table 1: Results of system comparison

Algorithm Sequence	ORB-LINE-SLAM	ORB-SLAM3	PL-SLAM	UV-SLAM	AIRSLAM	Ours
	ATE(RMSE)					
MH01	0.038	0.036	0.0416	0.161	0.028	0.027
MH02	F	0.033	0.0522	0.179	0.035	0.031
MH03	0.041	0.047	0.0399	0.176	0.042	0.038
MH04	0.044	0.052	0.0641	0.291	0.051	0.049
MH05	0.045	0.082	0.0697	0.189	0.064	0.052
V102	F	0.069	0.0523	0.071	0.045	0.042
V103	F	0.142	0.0826	0.094	0.070	0.072
V201	0.061	0.077	0.0659	0.078	0.062	0.053
V202	0.058	0.093	0.0568	0.085	0.060	0.055

The proposed algorithm achieved optimal performance across all six test sequences, demonstrating an average accuracy improvement of 28% over ORB-SLAM3, 20% over PL-SLAM, 58% over UV-SLAM, and 8.3% over AIRSLAM. The dense point cloud results generated by the algorithm are shown in Figure 3. Figure a and Figure b present the dense point cloud reconstructions for sequences V102 and V103, respectively, which accurately align with the camera perspective and fully restore the actual indoor scenes captured. Figures c and d correspond to the acquisition scene and reconstruction results for sequence MH01. Figure c displays the dense point cloud generated from this sequence, while Figure d shows the core elements relied upon for point cloud reconstruction, including keyframes, color matching, and trajectory information. The reconstruction quality in Figures a and b is visibly superior to that in Figures c and d. The reasons are analyzed as follows: During acquisition of sequences V102 and V103, the images contained rich and complete scene information, providing ample basis for dense reconstruction. In contrast, the MH01 sequence features sparse texture characteristics and complex depth hierarchies. Compounded by the drone's limited data collection to upper regions, the final reconstruction clearly displays only the upper-layer structure, resulting in relatively constrained overall performance.

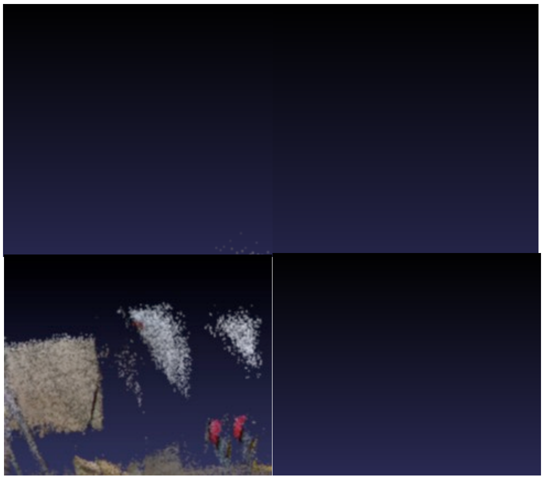
Figure 3: Dense point cloud reconstruction results for V102, V103 and MH01



To comprehensively validate system performance, this paper adapts the established method from Reference[23] for converting the TUM dataset to the KITTI format. Format adaptation and experimental validation were conducted on four sequences: f2_longoffice, f2_xyz, f2_dishs, and f2_slam3. The resulting dense point cloud reconstructions are shown in Figure 4.

Among these, the first three sequences focus on reconstructing and restoring scene feature details, while the fourth sequence, f2_slam3, resembles the V-series scenes in the EuROC dataset, employing a handheld camera to capture spatial

Figure 4: Dense point cloud reconstruction results for TUM dataset



traversal data through mobile acquisition. The reconstruction results demonstrate that the proposed system exhibits outstanding performance across various scenarios—whether reconstructing single targets, multi-object compositions, or large-scale spatial environments. All objects and structural details within the scenes remain clearly discernible, fully validating the system's effectiveness and robustness in dense point cloud mapping tasks.

Finally, to validate the system's performance in dynamic environments, experiments were conducted using the KITTI dataset. To ensure comparability of results, the system was tested on the first 11 sequences with ground truth labels and compared with other state-of-the-art systems. Specific details are presented in Table 2.

Table 2: Results of system comparison

Algorithm	ORB-SLAM2	ORB-SLAM3	PL-SLAM	OURS
Sequence	ATE(RMSE)			
00	1.267	1.214	1.243	1.195
01	10.054	13.197	12.096	9.854
02	6.832	6.032	6.206	6.132
03	0.357	1.256	0.554	0.342
04	1.833	1.054	1.136	0.639
05	3.799	1.082	1.053	0.968
06	3.102	3.494	0.858	1.324
07	1.838	0.872	0.968	0.772
08	4.857	3.612	3.427	3.054
09	3.972	1.803	1.752	1.664
10	1.267	1.386	1.246	1.286

As shown in Table 2, our system achieves optimal performance on 8 out of 11 test sequences in the KITTI dataset. The results for sequences 00, 01, and 08 demonstrate that our system exhibits particularly significant advantages for large-scale scene sequences. To further visually validate the positioning accuracy, we compare the trajectory estimation results of our system with those of ORB-SLAM3.

As shown in Figure 5, the figure displays a trajectory comparison between sequences 00 and 08 along with a detailed enlargement. The enlarged details clearly reveal that the trajectories generated by the proposed system exhibit higher alignment with the ground truth and superior positioning robustness. This enhanced positioning accuracy also lays a solid foundation for high-quality dense reconstruction. The corresponding dense point cloud reconstruction results are shown in Figure 6, where the top panels (a, b, c) display the trajectory maps, and the bottom panels (d, e, f) present the generated dense point cloud maps.

As shown in Figure 6, the proposed system demonstrates outstanding comprehensive performance: it can fully reproduce the driving trajectory captured during vehicle data collection, with continuous and gap-free trajectories throughout the journey, accurately reconstructing the motion path during the acquisition process; It also achieves high-quality dense mapping of road surfaces. The point cloud is dense and complete, with no noticeable sparse areas or voids (shadows represent the shadows of roadside objects under sunlight). Both the overall structure and local details of the road surface are accurately reproduced, further validating the system's trajectory estimation accuracy in dynamic and low-texture environments and the completeness of its dense mapping.

Figure 5: KITTI dataset trajectory map

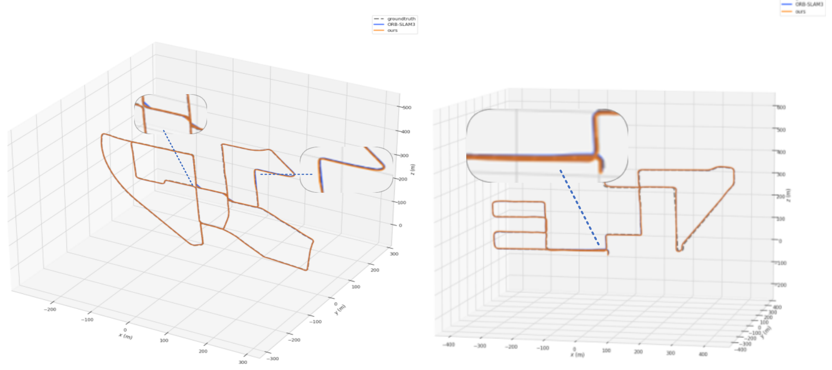
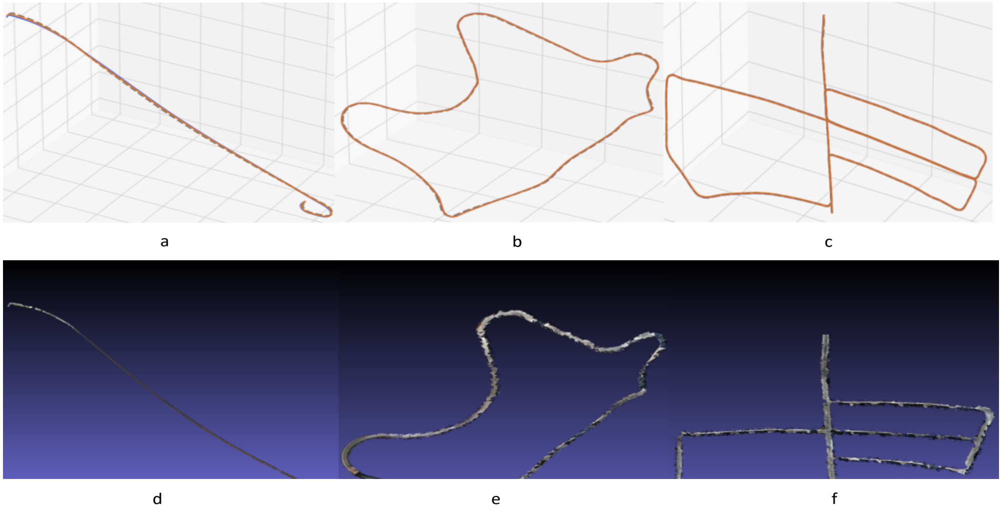


Figure 6: Comparison of KITTI dataset trajectory maps and dense point cloud maps



5. Conclusion

To address issues such as insufficient feature extraction, inaccurate localization, and sparse dense point clouds in visual SLAM systems under low-texture and dynamic environments, this paper proposes an improved stereo visual SLAM dense mapping solution based on PLCD-SLAM. This solution integrates point-line features with dual soft-hard depth constraints, effectively enhancing the system's robustness and mapping accuracy in complex scenes. Experimental validation conclusively demonstrates the effectiveness and superiority of the improved algorithm. Comparative tests on three major public datasets—KITTI, EuROC, and TUM (converted to stereo format)—reveal that our system achieves significantly lower absolute trajectory error (RMSE) than mainstream algorithms such as ORB-SLAM3, PL-SLAM, and UV-SLAM. UV-SLAM. The system achieved optimal performance on 8 out of 11 test sequences in the KITTI dataset. In experiments on EuROC and TUM converted sequences, the system not only demonstrated outstanding localization robustness but also generated dense point clouds that accurately reproduced scene details. It consistently delivered stable high performance, whether reconstructing small-scale indoor features or mapping complete road surfaces in outdoor vehicle-mounted scenarios.

In summary, the proposed improvement effectively addresses core bottlenecks in visual SLAM under complex environments, balancing positioning accuracy, environmental adaptability, and mapping quality. This provides reliable technical support for practical applications such as autonomous driving and indoor robot navigation. Future research may further explore multi-sensor fusion to tackle extreme environments and expand application boundaries.

References

[1] Zhang C, Huang T, Zhang R, Yi X. PLD-SLAM: A New RGB-D SLAM Method with Point and Line Features for Indoor Dynamic Scene. ISPRS International Journal of Geo-Information. 2021; 10(3):163.

[2] C. Forster, M. Pizzoli and D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, 15-22.

[3] Zhang J, Singh S. LOAM: Lidar odometry and mapping in real-time[C]//Robotics: Science and systems. 2014, 2(9): 1-9.

[4] Gomez-Ojeda R, Moreno F A, Zuniga-Noël D, et al. PL-SLAM: A stereo SLAM system through the combination of points and line segments[J]. IEEE Transactions on Robotics, 2019, 35(3): 734-746.

[5] Xu, Kuan et al. AirSLAM: An Efficient and Illumination-Robust Point-Line Visual SLAM System. IEEE Transactions on Robotics 2024, (41): 1673-1692.

[6] DeTone D, Malisiewicz T, Rabinovich A. Superpoint: Self-supervised interest point detection and description[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 224-236.

[7] HANG Chenyang, YANG Jian. A Visual SLAM Method Coupled with Adaptive Point-Line Features and IMU[J]. Geomatics and Information Science of Wuhan University, 2025, 50(10): 2048-2063.

[8] Gallagher L, Kumar V R, Yogamani S, et al. A hybrid sparse-dense monocular slam system for autonomous driving[C]//2021 European Conference on Mobile Robots (ECMR). IEEE, 2021: 1-8.

[9] Wimbauer F, Yang N, Von Stumberg L, et al. MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 6112-6122.

[10] Liu Y, Dong S, Wang S, et al. Slam3r: Real-time dense scene reconstruction from monocular rgb videos[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. 2025: 16651-16662.

[11] Yan C, Qu D, Xu D, et al. Gs-slam: Dense visual slam with 3d gaussian splatting[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 19595-19604.

[12] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]//European conference on computer vision. Cham: Springer International Publishing, 2014: 834-849.

[13] Wang R, Schworer M, Cremers D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras[C]//Proceedings of the IEEE international conference on computer vision. 2017: 3903-3911.

[14] Teed Z, Deng J. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras[J]. Advances in neural information processing systems, 2021, 34: 16558-16569.

[15] Mur-Artal, Raul and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 2016, (33):1255-1262.

[16] Xue et al. Construction of Dense Stereo Maps for Orchards Based on Adaptive Threshold ORB Feature Extraction [J]. Transactions of the Chinese Society of Agricultural Machinery, 2024.

[17] Wang et al. Research on Cross-Attention-Driven Dense Mapping Algorithm for Outdoor Stereo Vision SLAM [J]. Journal of Chongqing Technology and Forestry University (Natural Science), 2025.

[18] Hu Y, Zhen W, Scherer S. Deep-learning assisted high-resolution binocular stereo depth reconstruction[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 8637-8643.

[19] Koestler L, Yang N, Zeller N, et al. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo[C]//Conference on Robot Learning. PMLR, 2022: 34-45.

[20] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157-1163.

[21] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset[J]. The international journal of robotics research, 2013, 32(11): 1231-1237.

[22] Schubert D, Goll T, Demmel N, et al. The TUM VI benchmark for evaluating visual-inertial odometry[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1680-1687.

[23] Yang N, Stumberg L, Wang R, et al. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1281-1292.

Funding

This work was supported in part by the Doctoral Fund of Henan Polytechnic University (No. B2021-62, No. B2021-63), and in part by the Photoelectric Detection and Sensing Integrated Engineering Technology Research Center of Henan Province (No. KFZ202509).

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights