2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DepthMesh: A Dual-End Complementary Online Depth Estimation and Mesh Reconstruction

Jiaqi Yang[1], Dazhao Fan[1*], Xingbin Yang[2], Jiabin Yang[3], Song Ji[1], Yang Dong[1]
Ming Li[1], and Aosheng Wang[1]

Abstract— **We present a novel dual-end complementary method for online depth estimation and mesh reconstruction, termed DepthMesh. Unlike most existing state-of-the-art methods that produce either only depth online or surface mesh offline, our method tightly couples online multiview depth estimation and Truncated Signed Distance Function (TSDF) reconstruction to achieve fast online mesh reconstruction. For each keyframe from 6DoF tracking, we first obtain the prior depth and normal maps via ultra-fast raycasting from TSDF, which is incrementally fused from historical keyframe depths. Then, these priors, combined with segmentation results, are used to generate local planar hypotheses that optimize both depth accuracy and computational efficiency. Finally, the optimized depth estimates further enhance the accuracy of mesh reconstruction. Through this dual-end complementary mechanism, our system achieves high accuracy and efficiency. Experiments with qualitative and quantitative evaluations on the ScanNetV2 and self-collected datasets demonstrate the effectiveness of our method. Our method can generate depth and mesh online with accuracy (< 3 cm) on mobile devices, which is useful for robotic autonomous navigation and mixed reality applications such as real-time occlusion and collision handling.**

I. Introduction

Visual-based online 3D reconstruction, leveraging its significant advantage of low cost, has been widely applied in fields such as robot navigation, autonomous driving, and mixed reality [1], [2], [3]. Given the relative maturity of visual SLAM technology [4], [5], the key to achieving high-quality online 3D reconstruction lies in how to rapidly compute the accurate depth of images and the mesh surface.

Traditional depth estimation methods based on dense matching [6], [7] typically use fixed-window descriptors to compute matching costs. These methods can effectively reduce the time consumed in matching calculations, thereby enabling fast computation of image depth maps, but they are difficult to apply in online reconstruction on mobile devices. Subsequent works, such as 3D Modeling [8], MobileFusion [9], and Mobile3DRecon [10], have enabled online reconstruction on mobile devices after platform-specific optimizations. However, these methods did not consider the impact of large disparity variations on matching costs during depth estimation, resulting in limited depth accuracy and completeness. Mobile3Dscanner [11] uses a DNN-based post-processing approach to improve the completeness of depth map, but it loses depth map details. To address this issue, [12], [13] use precomputed surface normals to constrain matching cost, which can effectively improve depth accuracy in areas with weak textures and large disparity variations. However, these surface normals are typically obtained via the dominant slanted plane assumption [13], virtual normals [14], and depth maps [15]. The normal calculation strategies of these methods are relatively complex and computationally expensive, making it difficult to meet the requirements of online depth calculation.

In recent years, several studies have attempted to optimize depth estimation by leveraging deep learning Convolutional neural networks (CNNs) for feature extraction. For instance, the hierarchical structure of CNNs [16], [17], [18] is used to predict depth and normal geometric features from images, and geometric consistency constraints are applied to regularize depth variations, thereby improving the accuracy and completeness of depth in regions with large disparity variations. Another approach employs a method of cost encoding and convolutional aggregation to encode photometric features [19], normal features [20], and additional geometric constraint features [21] into the cost volume within the depth hypothesis space, refining the depth accuracy of images through convolutional aggregation. However, the complex processes of feature encoding and aggregation often incur high computational costs, making them similarly difficult to meet the requirements of online depth estimation. To balance computation time and accuracy, SimpleRecon [22] designed a simple 2D CNN architecture. By simply combining planar scan feature bodies with geometric loss functions, multiview keyframes and geometric metadata were effectively integrated into the cost volume, significantly reducing the computational load. DoubleTake [23] leverages the geometric mesh obtained from [22] to further improve the accuracy of depth estimation, but it is difficult to run on mobile platforms due to its higher computational complexity. In addition, by probabilistically fusing a semi-dense stereo algorithm with a 2D convolutional depth network [24], depth accuracy can also be optimized in real time. However, although these methods can rapidly infer complete depth maps, the accuracy of depth maps predicted solely by convolutional networks (based on cost volume prediction) still fails to reach the level achieved by traditional methods through dense matching cost calculation.

To address the aforementioned issues, this paper proposes

This work was supported by the National Natural Science Foundation of China (No.42371459, No.41971427, No.42401550), and the Henan Provincial Natural Science Foundation (No.242300421665).

[1]Jiaqi Yang, Dazhao Fan, Song Ji, Yang Dong, Ming Li, and Aosheng Wang are with the School of Surveying and Mapping, Information Engineering University, Zhengzhou 450001, China. E-mail: {yjq22919, fdzcehui, jisong_chxy, liming12102022, aosheng_wang}@163.com, {dongyang33}@aliyun.com

[2]Xingbin Yang is with Vivo Central Research Institute, Shanghai 200120, China. E-mail: 1126779429@qq.com

[3]Jiabin Yang is with ByteDance Inc., Beijing 100098, China. E-mail: yangkamau95@aliyun.com

* Corresponding author: Dazhao Fan

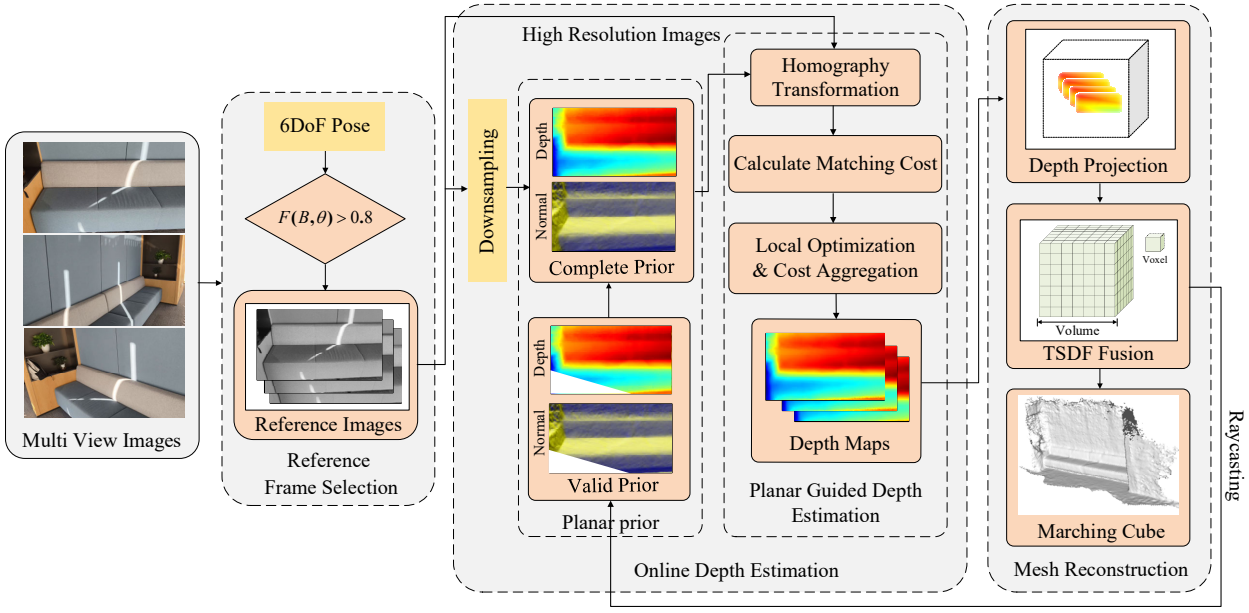


Fig. 1. Flowchart of the dual-end complementary online depth estimation and mesh reconstruction algorithm.

a dual-end complementary online depth estimation and mesh reconstruction method. By leveraging the Truncated Signed Distance Function (TSDF) raycasting based on incremental depth fusion from historical frames, the proposed method tightly couples the processes of online depth estimation and mesh reconstruction, thereby achieving the unification of efficiency and accuracy for online mesh reconstruction. The main contributions of this paper are as follows:

- A coupled framework for online depth estimation and mesh reconstruction is designed. Through the dual-end complementary process between depth estimation and mesh reconstruction, both the speed and accuracy of mesh reconstruction are effectively improved.
- A plane prior-guided matching cost calculation method is proposed, which leverages planar priors to constrain the cost calculation range while obtaining more accurate matching costs for surfaces with large disparities.
- The image segmentation results are introduced to optimize both local and semi-global cost aggregation processes, thereby improving depth accuracy in regions with weak textures and large disparity variations, and enhancing the robustness of the reconstruction algorithm in both indoor and outdoor scenarios.

II. Proposed Method

The overall process used in this paper is shown in Fig. 1, which consists of three modules: multiview reference frame selection, online depth estimation, and incremental mesh reconstruction. First, given a set of multiview frame images, on the basis of the range of the baseline length and camera inclination angle, combined with the camera pose, an appropriate image is dynamically selected as the reference frame. Second, the depth and normal priors obtained through the incremental TSDF raycasting are injected into the downsampled depth estimation process for the reference frame. This enables the acquisition of complete and reliable depth and normal priors, which are then incorporated as planar priors into the plane-guided depth estimation process of the original high-resolution image, and the corresponding image depth maps are calculated online. Finally, we use the incremental TSDF mesh reconstruction method to complete the online reconstruction of the entire scene.

A. Multiview Reference Frame Selection

Given a set of multiview images with camera poses, considering that depth is inversely proportional to disparity, the adjacent reference frames should have sufficient baseline spacing and maintain as much overlap as possible to provide a stable disparity and a basis for depth estimation. Therefore, the scoring function $F(B,\theta)$ between the multiview baseline length $B_{f_i}^{f_j}$ and the camera inclination angle $\theta_{f_i}^{f_j}$ is used to dynamically select reference frames. Generally, the top two frames sorted by score are chosen as the reference frames f_{i-1} and f_{i+1}. In our experiments, we set $F(B,\theta) > 0.8$ as the minimum threshold.

$$\begin{aligned} F(B,\theta) &= S_{f_i}^{f_j}(B) * S_{f_i}^{f_j}(\theta) \\ S_{f_i}^{f_j}(B) &= \mathbf{exp}(-(b_{f_i}^{f_j} - b_E)^2 / \sigma^2), \\ S_{f_i}^{f_j}(\theta) &= \mathbf{max}(\theta_E / \theta_{f_i}^{f_j}, 1) \end{aligned} \tag{1}$$

where $S_{f_i}^{f_j}(B)$ and $S_{f_i}^{f_j}(\theta)$ represent the baseline score and the camera inclination angle score for the current frame f_i and the adjacent reference frames f_j, respectively. b_E and θ_E are their respective expected values ($b_E = 0.6m$, $\theta_E = 15°$), σ denotes the standard deviation of the baseline.

B. Online Depth Estimation

To estimate the dense depth map online, this paper uses incremental TSDF raycasting and downsampling depth estimation to obtain complete and reliable depth and normal priors, thereby calculating the cost of multiview precise matching under the plane prior constraint and obtaining the final depth maps.

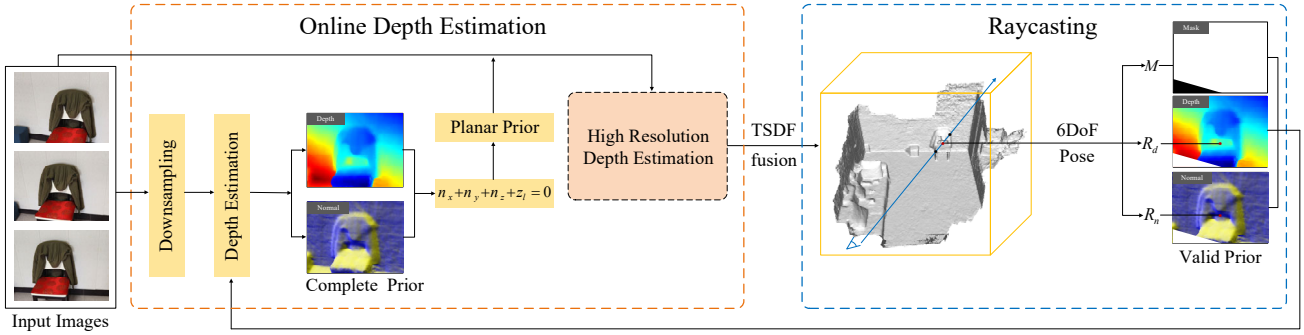


Fig. 2. Process of image plane prior calculation.

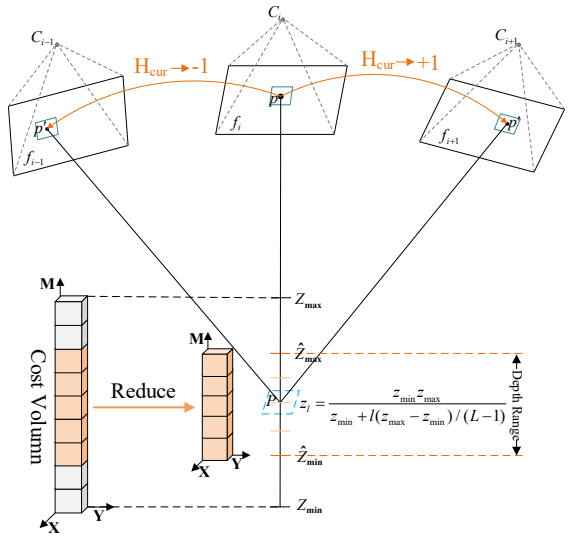


Fig. 3. Matching cost computation guided by the planar priors. $[\hat{z}_{min}, \hat{z}_{max}]$ represents the constrained depth search range, and the orange area in the cost volume represents the matching cost within the constrained range.

Planar Prior Calculation — In this work, the raycasting method [25] is used to calculate the valid mask, depth, and normal prior maps for the current frame from the TSDF fusion of the incremental mesh reconstruction (see the blue part in Fig. 2). Considering that the depth and normal priors obtained from the TSDF deeply fused from historical frames may have certain information deficiencies, to fill these missing areas, the raycasting results of the incremental TSDF are injected into the process of online downsampling depth estimation, and the depth and normal priors from downsampling depth estimation are used to fill the gaps in raycasting priors to obtain complete and reliable image priors. Subsequently, the plane priors of the current frame are obtained using the plane condition equation $n_x + n_y + n_z + z_l = 0$, where n_x, n_y, and n_z are the normals of different pixels, and z_l represents the pixel depth at the discrete sampling position l. These priors are then used as the planar priors to constrain the cost calculation and cost aggregation process of the high-resolution image, thereby reducing computational time and improving the accuracy of depth estimation. Fig. 2 illustrates the calculation process of the planar priors.

Planar Prior-Guided Cost Computation — For each high-resolution image, a nonequidistant discrete sampling strategy is adopted to perform discrete sampling in the depth search range $[z_{min}, z_{max}]$ to adapt to the depth variations of different scenes. L is the total number of discrete sampling positions ($L = 63$). Each sampling position corresponds to a matching cost value, and the l-th discrete sampling depth value z_l is as follows:

$$z_l = \frac{z_{min} z_{max}}{z_{min} + l(z_{max} - z_{min})/(L-1)}. \tag{2}$$

Considering that the computation of the matching cost process over the entire scene depth space range is complex and time consuming, the planar prior depth is used to constrain the original $[z_{min}, z_{max}]$ (see Fig. 3), thereby effectively reducing the depth search range for matching cost computations and significantly decreasing the computation time for the matching cost.

Furthermore, considering that pixels with large disparity variations are prone to generating incorrect matching costs during the multiview frame matching cost calculation process, the planar priors are used to calculate the local homographic transformation matrix $\boldsymbol{H}_{f_i}^{f_j}$ between multiple view frames (see (3)). Through $\boldsymbol{H}_{f_i}^{f_j}$, local image warp is performed for f_j to reduce the fronto-parallel bias in the matching cost calculation, thereby achieving consistency of the matching cost calculation window between f_i and f_j and improving the accuracy of the matching calculation. Notably, for pixels without planar priors, image pose information is used to project the pixels to f_j to obtain the corresponding pixel coordinates according to (4).

$$p' = \boldsymbol{H}_{f_i}^{f_j} \boldsymbol{p},\ \boldsymbol{H}_{f_i}^{f_j} = K_{f_j}(\tilde{R} - \frac{1}{z_p} t \boldsymbol{n}_p^T) K_{f_i}^{-1}, \tag{3}$$

$$p' = z_p h_{f_i}^{f_j} p + \Delta_{f_i}^{f_j},\ h_{f_i}^{f_j} = K_{f_j} R_{f_j}^T R_{f_i} K_{f_i}^{-1},\ \Delta_{f_i}^{f_j} = K_{f_j} R_{f_j}^T (T_{f_i} - T_{f_j}), \tag{4}$$

Where p' is the homogeneous coordinates of the projection point of a pixel p on f_j, and z_p denotes the discrete sampling depth corresponding to pixel p. $\boldsymbol{n}_p$ is the planar prior normal vector. K is the intrinsic matrix of the camera, and $[\tilde{R} \quad t]$ represents the relative transformation matrix of pixel p with respect to pixel p'. R and T are the rotation and translation matrices from the camera to the world coordinate system.

To calculate the matching cost for pixel p at the discrete sampling depth position l_p (the corresponding sampling depth is z_p), the center-symmetric census matching cost algorithm [26] is applied to compute the matching cost value $C_{f_i}^{f_j}$:

$$WCT_{n,m}(u,v) = \underset{(i,j)\in I_{n,m}}{\otimes} s(P(\mathbf{H}_{f_j}^{f_i}[u-i, v-j, 1]), P(\mathbf{H}_{f_j}^{f_i}[u+i, v+j, 1])),$$
$$C_{f_i}^{f_j}(u,v,l_p) = \text{Hamming}(WCT_{f_i}(u,v), WCT_{f_j}(u,v)), \tag{5}$$

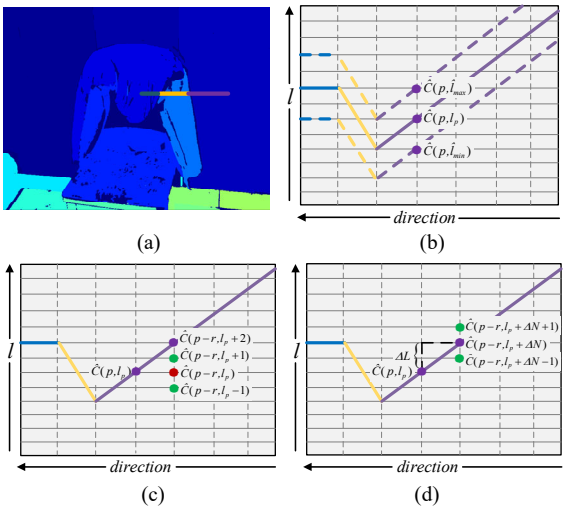


Fig. 4. Local and semi-global cost optimization process. (a) Image segmentation label map. (b) SGM with planar prior depth constraints. (c) Original SGM results. (d) SGM refined with the proposed cost penalty.

Where $WCT_{n,m}$ represents the bit chain of the central pixel (u,v). $\otimes$ is a bit-level concatenation symbol, which is used to concatenate the bit values calculated by the sign function $s(u,v)=0, if\ u \geq v; s(u,v)=1, otherwise$, $P(u \pm i, v \pm i)$ are the grayscale intensity of all pixels in symmetric regions on the $I_{n\times m}$ neighborhood window.

Local and Semi-global Cost Optimization — The original semi-global matching (SGM) strategy [6] constructs a global energy function $E(l)$ based on the matching cost C, which consists of a data term $C(p,l_p)$ and a smooth term $\varphi(l_p,l_q)$:

$$E(l)=\sum_p C(p,l_p)+\sum_{q\in N_p}\varphi(l_p,l_q)\,. \tag{6}$$

Although the aforementioned cost matrix C has been optimized accordingly, pixels in areas with weak texture features and large disparity variations may lack local correlation information, making cost aggregation prone to errors. The flood-fill segmentation [27] result is incorporated into the cost aggregation process, constraining local cost aggregation and cost penalties through the color similarity function $Label(p)$ (see Fig. 4(a)):

$$Label(p)=I_p\cdot\Gamma(\left|I_p-I_q\right|\leq\varepsilon)\,,\ q\in N_p\,, \tag{7}$$

Where N_p represents the four neighboring pixels of pixel p, I_p is the grayscale intensity of pixel p. $\Gamma(\cdot)$ is a judgment function, and ε is the color tolerance threshold ($\varepsilon=5$).

To enhance the smoothness of the depth maps, the local cost optimization of pixels in the same area is carried out using a weighted fusion formula (see (8)). To curtail computational overhead, cost aggregation is performed every 5 pixels:

$$\begin{cases}\hat{C}(p,l_p)=C(p,l_p)+\underbrace{w_{j-step}*C(p(i,j-step),l_p)}_{horizontal\ direction}\\ \qquad\qquad\underbrace{+w_{j+step}*C(p(i,j+step),l_p)}_{horizontal\ direction}\\ \hat{C}(p,l_p)=C(p,l_p)+\underbrace{w_{j-step}*C(p(i,j-step),l_p)}_{vertical\ direction}\\ \qquad\qquad\underbrace{+w_{j+step}*C(p(i,j+step),l_p)}_{vertical\ direction}\end{cases}, \tag{8}$$

where $\hat{C}(p,l_p)$ represents the aggregation value of pixel p in the horizontal and vertical directions, respectively. An aggregation optimization sequence of first horizontal and then vertical aggregation is followed. w is the weight between the pixel p and the pixels in the step neighborhood ($step=5$). According to the principle of "weighting only within the same region", the weights of the neighboring pixels are assigned a value of 0 or 1, thus optimizing the aggregation cost.

To solve (6) online, the planar prior depth constraint cost aggregation process is applied (see Fig. 4(b)). For the prior depth of pixel p, the nearest prior depth position l_p is calculated using (2). By setting the depth position variation δ_l ($\delta_l=8$) within a certain tolerance range $[\hat{l}_{min},\hat{l}_{max}]$ ($\hat{l}_{max}=l_p+\delta_l$, $\hat{l}_{min}=l_p-\delta_l$) to limit the cost aggregation range, a uniform penalty P_2 is applied to depth positions outside the tolerance range, thereby optimizing the time consumption of cost aggregation. Furthermore, considering that the traditional SGM strategy imposes incorrect cost penalties on pixels in areas with weak texture features and large disparity variations (see the red path cost in Fig. 4(c)), the discrete depth jump step ΔL of pixels is used to shift the original depth position l_p to a depth position $l_p+\Delta L$ consistent with the surface direction (see Fig. 4(d)), achieving a zero-cost penalty transition for pixel p, thereby improving the global accuracy of the depth maps without additional time consumption. The aggregation cost for each r direction is as follows:

$$\begin{aligned}&if\ \ l_p>\hat{l}_{\max}(p)\ \ or\ \ l_p<\hat{l}_{\min}(p)\\ &\quad L_r(p,l_p)=\hat{C}(p,l_p)+P_2\\ &else\\ &\quad L_r(p,l_p)=\hat{C}(p,l_p)+\\ &\quad \min_{l_q\in[\hat{l}_{\min},\hat{l}_{\max}]}(L_r(p-r,l_q)+\psi(l_p,l_q))-\\ &\quad \min_{k\in[\hat{l}_{\min},\hat{l}_{\max}]}L_r(p-r,k)\\ &end\end{aligned}\quad,\ \varphi(l_p,l_q)=\begin{cases}0, & if\ l_p+\Delta L=l_q\\ P_1, & if\ \left|l_p+\Delta L-l_q\right|=1\\ P_2, & if\ \left|l_p+\Delta L-l_q\right|>1\end{cases}, \tag{9}$$

where q represents a pixel in the $p-r$ direction, ΔL is obtained by rounding the difference between l_p and l_{p-r}:

$$\Delta L=round(l_p-l_{p-r})\,. \tag{10}$$

The optimal depth sampling position $\tilde{l}_p$ is determined using the WTA (Winner Takes All) algorithm:

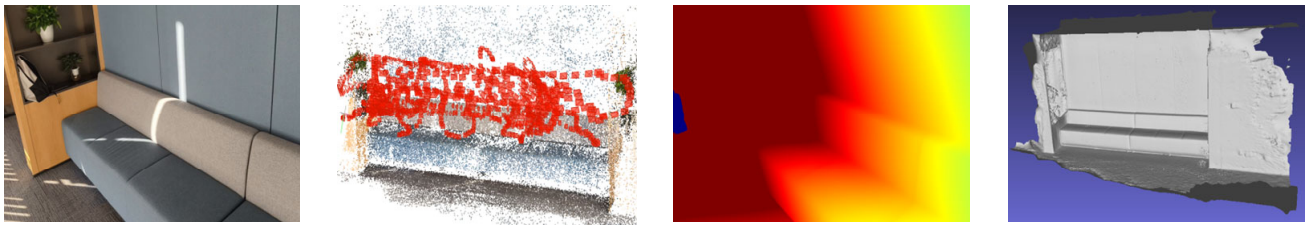
$$\tilde{l}_p=argmin\sum_{r=1}^{8}L_r(p,l_p)\,, \tag{11}$$

Where $\sum_{r=1}^{8}L_r(p,l_p)$ is the final aggregated cost value, and the optimal depth value $\tilde{z}_p$ of the pixel is determined using (2). The final depth maps are generated by traversing all pixels.

C. Incremental Mesh Reconstruction

In this paper, an incremental TSDF mesh reconstruction method is employed, in which a weighted fusion algorithm is applied to optimize the mesh reconstruction process, avoiding the repetitive calculation of image pixels from historical frames and effectively reducing the computational time and storage consumption.

Given an initialized 3D voxel mesh, by using image poses and according to the coordinate transformation relationship



(a) Keyframe Image (b) 6DoF Pose (c) GT Depth (d) GT Mesh

Fig. 5. Example of the self-collected dataset. The self-collected dataset includes keyframe images (with a resolution of 800×600), corresponding ground truth (GT) depth maps for the keyframes, and the GT mesh of the scene. The GT mesh was obtained through offline map reconstruction using the RGBD depth map from Kinect. The GT depth for each keyframe was obtained by first registering the keyframe image to the Kinect map to acquire the 6DoF (6 Degrees of Freedom) pose of the keyframe and then rendering the depth using the scene mesh.

in (12), the depth maps from multiple views are projected onto the 3D voxel mesh, thereby obtaining the depth value z_v corresponding to each voxel.

$$P_v = \tilde{z}_p K_{f_i}^{-1} R_{f_i} \hat{p} + T_{f_i} \,. \tag{12}$$

For each voxel V, the weighted average fusion formula in (13) is used to incrementally calculate and update the TSDF value and weight of voxels within the truncation distance $[-\tau, \tau]$ $(\tau = 3cm)$:

$$T_t^p(V) = \frac{T_{t-1}^p(V) W_{t-1}(V) + sgn[(\tilde{z}_p - z_v) * \min(z_v, \tau)] W_t(V)}{W_{t-1}(V) + W_t(V)}, \tag{13}$$

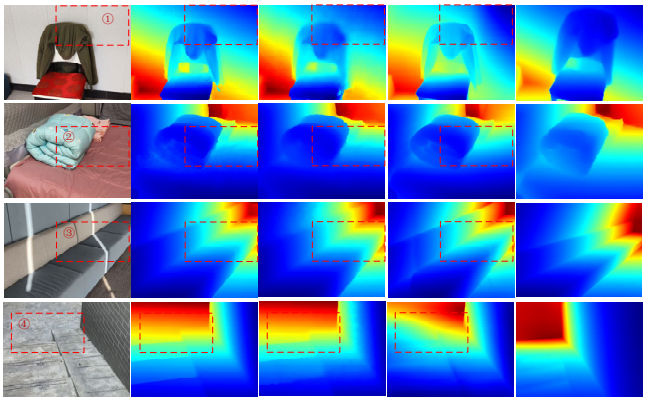
Where $T_t^p(V)$ and $T_{t-1}^p(V)$ represent the TSDF values of voxel V at times t and $t-1$. W denotes the weight value, $W_t(V) = W_{t-1}(V) + 1$. $sgn(\cdot)$ is the sign judgment function, $\min(\cdot)$ is the auxiliary function. z_v represents the depth value corresponding to V. When $z_v \leq \tau$, the voxel is incrementally updated according to (13); otherwise, the voxel is no longer updated to avoid invalid calculations and noise and to increase the robustness of the result.

After all the voxel values are updated, the marching cubes algorithm [28] is employed to extract the isosurface with a zero weighted sum of distances from the updated voxel mesh, as the reconstructed scene surface, thereby generating the final mesh.

III. Experimental Results and Evaluation

To validate the effectiveness of the proposed method, we conducted quantitative and qualitative evaluations of both the depth maps and mesh reconstruction results using the public ScanNetV2 dataset and a self-collected dataset (including four sets of indoor and outdoor data collected with a mobile phone). An example of the GT data for one of the scenarios is shown in Fig. 5. Notably, the keyframes in the self-collected dataset were obtained through SLAM. In addition, the same Scan-Netv2 dataset as that used to validate SimpleRecon was applied to evaluate the accuracy of the depth maps. Since SimpleRecon has specific field of view requirements for the input image, we uniformly resized the ScanNetV2 images and the depth maps to a resolution of 800*600 as the inputs for the algorithm.

The proposed algorithm was developed on the Ubuntu 22.04 system and implemented entirely in C++ (with acceleration using OpenCL 2.0). The experiments were conducted on a hardware platform consisting of an Intel(R) Core (TM) i7-13700KF CPU, with a minimum frequency of 800 MHz and a maximum frequency of 5400 MHz, and an NVIDIA RTX 4080 graphics card.



RGB GT Depth Ours SimpleRecon 3D Scanner

Fig. 6. Depth estimation results obtained for the four self-collected dataset. From blue to red indicates that the distance between the camera and the scene changes from close to far. The red boxes highlight the regions with the most significant depth differences.

TABLE I. Quantitative Comparison Based on the ScanNetv2 Depth Benchmark. The Best Results are Shown in Bold

Method	ScanNetV2			
	Abs Err /(m) ↓	Abs Rel /(%) ↓	Sq Rel /(%) ↓	$\delta < 1.05$ ↑
DPSNet	0.1552	7.950	2.990	49.36
MVDepthNet	0.1648	8.480	3.430	46.71
DELTAS	0.1497	7.860	2.760	48.64
SimpleRecon	0.0885	**4.340**	1.250	73.16
Ours	**0.0784**	4.670	**1.150**	**78.11**

A. Depth Estimation Evaluation

Table I presents a comparison of the quantitative results obtained with our method and other state-of-the-art deep learning methods [29], [30], [31], [22] on the ScanNetV2. Compared with all methods, our method performs the best overall. In particular, regarding the absolute error, our method reduces the error by more than 1 cm compared to SimpleRecon.

To thoroughly evaluate the effectiveness of the proposed depth estimation method, Fig. 6 shows a comparison of the qualitative depth results of our method, two other methods that support online depth estimation, and the GT depth maps on the four self-collected datasets. Notably, the 3D Scanner depth and mesh are acquired by iPad Pro app, which uses dToF (Direct Time of Flight) sensor to capture dense depth maps within 5 meters. It is obvious that the depth visualization results produced by our method are closest to the ground truth. As shown by the red box ① in Fig. 6, the depth scale and smoothness of our method in the chair and wall regions are superior to those of SimpleRecon. This finding indicates that

TABLE II. QUANTITATIVE DEPTH RESULTS FOR THE SELF-COLLECTED DATASET

Scene	Method	Abs Diff /(m)↓	Abs Rel /(%)↓	Sq Rel /(%)↓	MSE /(m)↓	RMSE /(m)↓	$RMSE_{log}$ /(m)↓
Room	SimpleRecon	0.0994	5.3382	0.8779	0.0165	0.1285	0.0397
	3D Scanner	0.0297	1.8939	0.3410	0.0058	0.0762	**0.0227**
	Ours	**0.0164**	**0.9548**	**0.3019**	**0.0050**	**0.0706**	0.0377
Bed	SimpleRecon	0.0654	2.9561	0.5118	0.0140	0.1183	0.0204
	3D Scanner	0.0296	1.9534	0.1795	0.0030	0.0552	0.0150
	Ours	**0.0177**	**1.0292**	**0.0930**	**0.0017**	**0.0409**	**0.0108**
Sofa	SimpleRecon	0.3433	15.301	5.5201	0.1302	0.3608	0.0743
	3D Scanner	0.0440	2.3845	**0.1574**	**0.0032**	**0.0566**	**0.0124**
	Ours	**0.0219**	**1.0465**	0.2132	0.0044	0.0662	0.0135
Outdoor	SimpleRecon	0.3950	14.096	5.8194	0.1665	0.4080	0.0678
	3D Scanner	0.0176	0.9098	0.0460	0.0010	0.0314	0.0069
	Ours	**0.0129**	**0.5681**	**0.0325**	**0.0007**	**0.0259**	**0.0053**

TABLE III. QUANTITATIVE COMPARISON OF THE MESH RECONSTRUCTION RESULTS ACROSS FOUR DIFFERENT SCENARIOS

Scene	Method	Comp ↑	Precision ↑	Recall ↑	F1-score ↑	MAE ↓	SD ↓
Room	SimpleRecon	0.8358	0.8516	0.7763	0.8122	0.0352	0.0471
	3D Scanner	0.9042	0.9231	**0.9471**	**0.9349**	0.0362	0.0484
	Ours	**0.9354**	**0.9744**	0.8587	0.9129	**0.0111**	**0.0172**
Bed	SimpleRecon	0.8347	0.8765	0.8094	0.8416	0.0313	0.0408
	3D Scanner	**0.9355**	**0.9624**	0.9053	**0.9330**	0.0325	0.0441
	Ours	0.9047	0.9395	**0.9135**	0.9263	**0.0186**	**0.0235**
Sofa	SimpleRecon	0.6719	0.8233	0.6541	0.7290	0.0543	0.0553
	3D Scanner	**0.8572**	**0.9375**	**0.8512**	**0.8923**	**0.0193**	**0.0296**
	Ours	0.8381	0.8892	0.8144	0.8501	0.0290	0.0423
Outdoor	SimpleRecon	0.7185	0.6330	0.5164	0.5688	0.0744	0.0598
	3D Scanner	0.9342	0.9485	**0.8665**	0.9057	0.0194	0.0224
	Ours	**0.9599**	**0.9898**	0.8534	**0.9165**	**0.0067**	**0.0110**

TABLE IV. AVERAGE ONLINE COMPUTATIONAL TIME PER FRAME. THE COMPUTING TIME FOR EACH FRAME IN THE FOUR SCENARIOS IS NOT SIGNIFICANTLY DIFFERENT. ACCORDING TO THE SEQUENCE OF SCENARIOS IN TABLE III, THE TOTAL NUMBER OF KEYFRAMES IN EACH SCENARIO IS 395, 421, 381, AND 307 FRAMES

Platform	Method	Time [ms/frame]			
		Depth Estimation	***TSDF Integration***	***Raycasting***	***Meshing***
PC	Ours	6.95	0.54	0.93	546.19
	SimpleRecon	88			2000
Xiaomi 12 Pro (SDM8)	Ours	35.30	3.72	4.89	1747.20

the depth estimation method in this paper is more accurate for calculating the matching cost in the depth space. Furthermore, as indicated by the red boxes ② and ③ in Fig. 6, our method incorporates planar prior constraints into both local cost calculation and cost aggregation, resulting in recovered planes that are more accurate and better aligned with the actual surfaces than are those obtained with other methods. As indicated by the red box ④ in Fig. 6, the depth edges produced by our method are significantly richer in detail and higher in accuracy. This is mainly because our method incorporates image segmentation information, which preserves depth edge details. In contrast, SimpleRecon not only loses fine details along the stair edges but also generates noticeable distortions at the ground surface.

Table II presents the quantitative results of depth accuracy evaluation across the four self-collected data scenes. It is evident that the depth estimates obtained with our method are significantly better than those of SimpleRecon across all metrics. Particularly for the "sofa" and "outdoor stairs" datasets, our method achieves a remarkable improvement. This finding indicates that SimpleRecon lacks sufficient generalization capability under varying lighting conditions and for outdoor scenes, whereas our method is not limited by scene type and can achieve stable depth estimation accuracy. Compared with 3D Scanner, our method exhibits inferior performance in terms of depth accuracy for Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Root Mean Squared Error in the logarithmic domain ($RMSE_{log}$), yet it gains obvious advantages in Absolute Difference (Abs Diff) and Absolute Relative Error (Abs Rel). One important reason for this difference is that during the depth estimation process, our method considers excluding the matching costs of nonoverlapping regions of the image, which results in a final depth map with less noise and higher accuracy.

B. Mesh Reconstruction Evaluation

To verify the mesh generation effect of our method, we conducted both qualitative and quantitative assessments of our incremental mesh generation method based on TSDF fusion and compared our method with two other methods and the ground truth mesh. Notably, all three methods generate meshes via TSDF fusion, thus offering strong comparability.

Fig. 7 displays visualizations of the meshes generated in four different scenarios. Our method effectively mitigates the

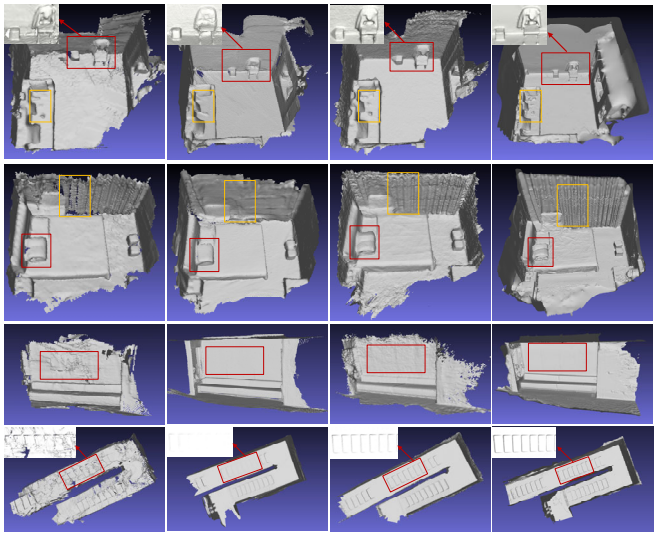


Fig. 7. Qualitative comparison of the mesh reconstruction results. Each row represents different comparative methods applied to one scene, and each column shows the mesh reconstruction results for four distinct scenes under the same method, namely, "Room", "Bed", "Sofa", and "Outdoor Stairs". The red and yellow boxes highlight the areas of local detail.

mesh voids caused by occlusions and incomplete scanning that other methods may produce, while demonstrating superior capability in representing fine details (e.g., chairs, curtains, and staircases). Benefiting from the introduction of planar priors and image segmentation results, our method produces mesh results with sharper edges and more distinct contours. Even in regions with weak textures and large disparity variations, such as walls and floors, our method still achieves good reconstruction results.

As shown in Table III, in terms of completeness (Comp), the meshes reconstructed by our method achieve a completeness rate above 90% in most cases, demonstrating high reconstruction integrity. Regarding accuracy, our method significantly outperforms SimpleRecon across all metrics in the four scenarios with a notable improvement of up to 91% observed in the "Outdoor Stair" scenario. Compared with 3D Scanner, our method exhibits lower Mean Absolute Error (MAE) and Standard Deviation (SD), indicating superior precision. Although the comprehensive F1-score is comparable with a gap within 5%, our method achieves a 1.2% higher F1-score than 3D Scanner specifically in the "Outdoor Stair" scene, further validating its robustness in complex environments.

To visually reflect the distribution of mesh errors, we utilized CloudCompare software to align the meshes obtained by all methods with the corresponding ground truth meshes and further compared the accuracy of the meshes. As shown in Fig. 8, which displays the distribution of mesh errors, our method exhibits a relatively uniform error distribution and less noise on the mesh surface. Furthermore, it achieves superior accuracy in regions with weak textures and large disparity variations, such as floors, walls, and bed surfaces.

C. Algorithm Performance Evaluation

Table IV presents the quantitative time consumption results of our method and SimpleRecon on both a PC and a mobile device. Since SimpleRecon can only be implemented

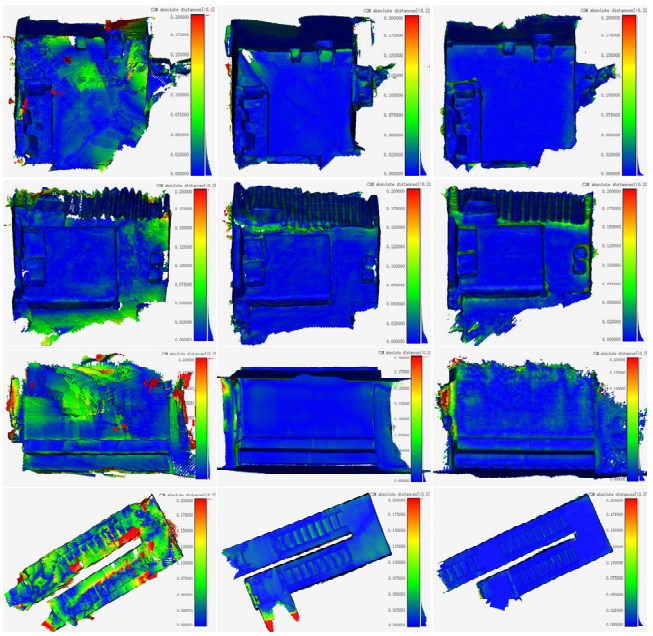


Fig. 8. Overall error distribution results for four groups of scenes. The legend on the right represents the error distribution bands, with colors ranging from blue to red indicating increasing accuracy from high to low. The maximum error is set not to exceed 0.2 meters.

on a PC, in addition to calculating the computational time of our method on the mobile device, we also conducted a time consumption comparison analysis of the two methods on the PC. Our method outperforms SimpleRecon in terms of time consumption across all stages, with the total time reduced by a factor of 3.76 times; on the mobile device, our depth estimation method takes an average of less than 45 ms per frame, and the postprocessing stage of mesh reconstruction (marching cube extraction for the mesh) is completed within 2 seconds.

IV. Conclusion

This paper proposes DepthMesh, a novel dual-end complementary online depth estimation and mesh reconstruction method. The method constructs a coupled online depth estimation and mesh reconstruction framework leveraging incremental TSDF raycasting. It effectively reuses planar prior information from the online mesh to optimize the depth estimation process, while the estimated depth results in turn refine the mesh quality. This dual-end complementary mechanism simultaneously improves both the speed and accuracy of online mesh reconstruction.

Experiments with qualitative and quantitative evaluations on the public ScanNetV2 dataset and a self-collected dataset demonstrate that the proposed method achieves superior depth accuracy and mesh quality compared to most deep learning-based methods. It produces satisfactory reconstruction results even in regions with weak textures and large disparity variations. Moreover, the proposed method achieves a depth estimation speed of 45 ms per frame on mobile devices while maintaining a mesh reconstruction accuracy better than 3 cm.

In future work, we will further consider integrating deep learning methods to improve the prior depth and normal for weakly textured and non-overlapping image regions, thereby

optimizing depth maps and reducing computational time. Considering our method can provide segmentation label information, we will attempt to incorporate the image segmentation results into the mesh reconstruction process to improve the mesh flatness and quality, reduce the number of meshes, and enhance the applicability of this algorithm in various complex scenarios.

REFERENCES

[1] Q. Serdel, C. Grand, J. Marzat, and J. Moras, "Online localisation and colored mesh reconstruction architecture for 3D visual feedback in robotic exploration missions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8690-8697.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907-1915.

[3] C. Li, H. Fan, X. Huang, R. Liang, S. Durvasula, and N. Vijaykumar, "DISORF: A Distributed Online 3D Reconstruction Framework for Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1329-1336, 2025.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874-1890, 2021.

[5] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally Consistent and Tightly Coupled 3D LiDAR Inertial Mapping," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5622-5628.

[6] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328-341, 2008.

[7] Q. Xu, W. Kong, W. Tao, and M. Pollefeys, "Multi-Scale Geometric Consistency Guided and Planar Prior Assisted Multi-View Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4945-4963, 2023.

[8] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "3D Modeling on the Go: Interactive 3D Reconstruction of Large-Scale Scenes on Mobile Devices," in *2015 International Conference on 3D Vision (3DV)*. IEEE, 2015, pp. 291-299.

[9] P. Ondrúška, P. Kohli, and S. Izadi, "MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1251-1258, 2015.

[10] X. Yang, L. Zhou, H. Jiang, Z. Tang, and G. Zhang, "Mobile3DRecon: real-time monocular 3D reconstruction on a mobile phone," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 12, pp. 3446-3456, 2020.

[11] X. Xiang, H. Jiang, G. Zhang, Y. Yu, and H. Bao, "Mobile3DScanner: An online 3D scanner for high-quality object reconstruction with a mobile device," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, pp. 4245-4255, 2021.

[12] D. Scharstein, T. Taniai, and S. N. Sinha, "Semi-global stereo matching with surface orientation priors," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 215-224.

[13] L. Roth and H. Mayer, "Reduction of the fronto-parallel bias for wide-baseline semi-global matching," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W5, pp. 69-76, 2019.

[14] W. Yin, Y. Liu, and C. Shen, "Virtual Normal: Enforcing Geometric Constraints for Accurate and Robust Depth Prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7282-7295, 2022.

[15] D. Ulucan, O. Ulucan, and M. Ebner, "A Scale-space Approach for Surface Normal Vector Estimation from Depth Maps," *SN Computer Science*, vol. 5, no. 6, pp. 1-12, 2024.

[16] H. Zhan, C. S. Weerasekera, R. Garg, and I. Reid, "Self-supervised Learning for Single View Depth and Surface Normal Estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4811-4817.

[17] B. Li, Y. Huang, Z. Liu, D. Zou, and W. Yu, "StructDepth: Leveraging the structural regularities for self-supervised indoor depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12663-12673.

[18] H. Wu, S. Gu, L. Duan, and W. Li, "GeoDepth: From Point-to-Depth to Plane-to-Depth Modeling for Self-Supervised Monocular Depth Estimation," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2025, pp. 11525-11535.

[19] G. Yang, R. Cao, J. Wen, B. Zhao, Q. Li, Y. Huang, L. Lei, X. Chen, A. Lam, and Y. H. Liu, "Multi-View Stereo with Geometric Encoding for Dense Scene Reconstruction," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 12473-12480.

[20] J. Wu, R. Li, H. Xu, W. Zhao, Y. Zhu, J. Sun, and Y. Zhang, "Gomvs: Geometrically consistent cost aggregation for multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20207-20216.

[21] K. Choi, S. Jeong, Y. Kim, and K. Sohn, "Stereo-augmented Depth Completion from a Single RGB-LiDAR image," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13641-13647.

[22] M. Sayed, J. Gibson, J. Watson, V. Prisacariu, M. Firman, and C. Godard, "SimpleRecon: 3D reconstruction without 3D convolutions," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 1-19.

[23] M. Sayed, F. Aleotti, J. Watson, Z. Qureshi, G. Garcia-Hernando, G. Brostow, S. Vicente, and M. Firman, "Doubletake: Geometry guided depth estimation," in *Proceedings of the European Conference on Computer Vision*, 2024, pp. 121-138.

[24] T. Laidlow, J. Czarnowski, and S. Leutenegger, "DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4068-4074.

[25] H. Ray, H. Pfister, D. Silver, and T. A. Cook, "Ray casting architectures for volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 3, pp. 210-223, 1999.

[26] J. Yang, D. Fan, J. Yang, X. Yang, and S. Ji, "A large scale online UAV mapping algorithm for the dense point cloud and digital surface model generation," *Bulletin of Surveying and Mapping*, vol. 0, no. 10, pp. 47-53, 2023.

[27] A. Roychoudhury, M. Missura, and M. Bennewitz, "Plane segmentation using depth-dependent flood fill," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 2210-2216.

[28] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *ACM SIGGRAPH Computer Graphics*, 1998, pp. 347-353.

[29] S. Im, H. G. Jeon, S. Lin, and I. S. Kweon, "DPSNet: end-to-end deep plane sweep stereo," in *International Conference on Learning Representations (ICLR)*. IEEE, 2019.

[30] K. Wang and S. Shen. "Mvdepthnet: Real-time multiview depth estimation neural network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 248-257.

[31] A. Sinha, Z. Murez, J. Bartolozzi, V. Badrinarayanan, and A. Rabinovich, "DELTAS: depth estimation by learning triangulation and densification of sparse points," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 104-121.