# SD-MVS: Segmentation-Driven Deformation Multi-View Stereo with Spherical Refinement and EM Optimization

**Zhenlong Yuan[1], Jiakai Cao[1], Zhaoxin Li[2, 3*], Hao Jiang[1], Zhaoqi Wang[1]**

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]Agricultural Information Institute, Chinese Academy of Agricultural Sciences
[3]Key Laboratory of Agricultural Big Data, Ministry of Agriculture and Rural Affairs
yuanzhenlong21b@ict.ac.cn, caojiakai21@mails.ucas.ac.cn,
cszli@hotmail.com, {jianghao, zqwang}@ict.ac.cn

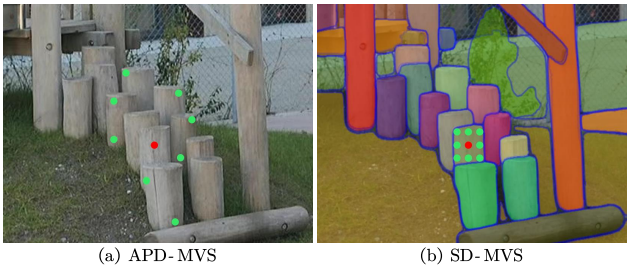
## Abstract

In this paper, we introduce Segmentation-Driven Deformation Multi-View Stereo (SD-MVS), a method that can effectively tackle challenges in 3D reconstruction of textureless areas. We are the first to adopt the Segment Anything Model (SAM) to distinguish semantic instances in scenes and further leverage these constraints for pixelwise patch deformation on both matching cost and propagation. Concurrently, we propose a unique refinement strategy that combines spherical coordinates and gradient descent on normals and pixelwise search interval on depths, significantly improving the completeness of reconstructed 3D model. Furthermore, we adopt the Expectation-Maximization (EM) algorithm to alternately optimize the aggregate matching cost and hyperparameters, effectively mitigating the problem of parameters being excessively dependent on empirical tuning. Evaluations on the ETH3D high-resolution multi-view stereo benchmark and the Tanks and Temples dataset demonstrate that our method can achieve state-of-the-art results with less time consumption.

## Introduction

Multi-view stereo (MVS) is a technique that employs images to reconstruct 3D objects or scenes. Its application spans various fields, including autonomous driving (Orsingher et al. 2022), augmented reality (Cao et al. 2021), and robotics (Li, Gogia, and Kaess 2019).

Recently, PatchMatch-based methods (Schönberger et al. 2016; Xu and Tao 2019; Lee et al. 2021) exhibits remarkable capabilities in sub-pixel reconstruction for large-scale imagery while being reliable for unstructured image set. These methods typically initiate by computing the matching cost of fixed patches between images, then proceeding with propagation and refinement for accurate depth estimation. Nonetheless, they typically encounter difficulties in textureless areas where the absence of texture results in unreliable depth estimations. To address this issue, several techniques have been introduced, including plane prior (Xu and Tao 2020), superpixel-wise planarization (Romanoni and Matteucci 2019), epipolar geometry (Xu et al. 2020) and confidence-based interpolation (Li et al. 2020). Yet when facing large textureless areas, these methods perform unsatisfactory and leave room for further improvement.



(a) APD-MVS (b) SD-MVS

Figure 1: Comparative analysis of patch deformation strategies between APD-MVS and our approach. APD-MVS (a) selects green anchor pixels from pixels characterized by similar colors but may have inconsistent depths to help reconstruct central red pixel, leading to potential inaccuracy. Conversely, our method (b) utilizes neighboring pixels inside the segmentation boundary for reconstruction.

Differently, learning-based methods leverages network to build learnable 3D cost volumes and thereby ameliorating the reconstruction quality. Several methods (Yao et al. 2019; Yan et al. 2020) attempt to employ the gated recurrent unit (GRU) to provide a more rational interpretation in reconstruction, while this often leads to unaffordable time and memory cost. Others (Su and Tao 2023) try to utilize residual learning module to refine depth estimates by rectifying the upsampling errors. Yet, such networks typically lack generalization when facing scenes different from training datasets, posing challenges for their practical application.

Edges in the color image are usually consistent with depth boundaries. Thus, edge information plays a pivotal role in both computation of PatchMatch and construction of 3D cost volumes. Nonetheless, problems like shadows and occlusions in complicated scenes tend to weaken the linkage between edge and depth boundaries. Consequently, several methods (Yuesong Wang et al. 2023) struggle to harness edge information effectively, often skipping edges and consequently calculating regions with inconsistent depth, leading to detail distortion, as shown in Fig. 1. Additionally, certain superpixel segmentation approaches (Kuhn, Lin, and Erdler 2019) face challenges in precisely segmenting edges

*Corresponding Author.

and lack semantic information to broaden receptive field. Differently, as an instance segmentation model, the Segment Anything Model (SAM) (Kirillov et al. 2023) can subtly mitigates the aforementioned disturbances, thereby segmenting instances with different depths across diverse scenes.

Therefore, we introduce SD-MVS, a PatchMatch-based method that integrates SAM-based instance segmentation to better exploits edge information for patch deformation. Specifically, we first employ the instance segmentation results derived from SAM to adaptively deform the patches for matching cost and propagation, thereby accommodating the distinct characteristics of different pixels. Moreover, we employ multi-scale matching cost and propagation scheme to extract diverse information, addressing the challenges posed by textureless areas. To optimize memory consumption, we introduce an architecture promoting multi-scale consistency in parallel, consequently reducing the program's runtime.

Moreover, we propose the spherical gradient refinement to optimize previous refinement strategies. Concerning with normal refinement, we randomly select two orthogonal unit vectors perpendicular to the current normal for perturbation and incorporate gradient descent to further refine perturbation directions in subsequent rounds, thereby improving the accuracy for each hypothesis. Regarding depth refinement, we adopt pixelwise search interval derived from the deformed patch for local perturbations.

Furthermore, we introduce an EM-based hyperparameter optimization to address the issue of empirical determination of hyperparameters in existing methods. By alternately optimizing the aggregated cost and the hyperparameters, we implement an excellent strategy for automatic parameter tuning, thereby facilitating a balanced consideration against diverse information. Evaluation results on the ETH3D and the Tanks and Temples benchmarks illustrate that our method surpasses the existing state-of-the-art (SOTA) methods.

In summary, our contributions are as follows:

- Based on SAM segmentation, we propose an adaptive patch deformation with multi-scale consistency on both matching cost and propagation to better utilize image edge information and memory cost.
- We introduce the spherical gradient refinement, which leverages spherical coordinates and gradient descent on normals and employs pixelwise search interval to constrain depths, thereby enhancing search precision.
- We propose the EM-based hyperparameter optimization by adopting the EM algorithm to alternately optimizing the aggregate cost and the hyperparameters.

## Related Work

**Traditional MVS Methods** Traditional Multi-View Stereo (MVS) algorithms can primarily be classified into four categories (Seitz et al. 2006): voxel-based methods (Vogiatzis et al. 2007), surface evolution-based methods (Cremers and Kolev 2011) , patch-based methods (Bleyer, Rhemann, and Rother 2011), and depth-map based methods (Yao et al. 2019). Our methodology aligns with the last category, where depth maps are generated from images and their corresponding camera parameters, further leading to point cloud construction via fusion. Within this category, PatchMatch-based methods are the most well-known subclass. Numerous innovative PatchMatch-based methods have been proposed and accomplished a great enhancement in both accuracy and completeness. ACMM (Xu and Tao 2019) uses multi-view consistency and cascading structure to tackle reconstruction of textureless areas, while subsequent works such as ACMMP (Xu et al. 2022) further introduce a plane-prior probabilistic graph model and thus provide plane hypothesis for textureless areas. In contrast, TAPA-MVS (Romanoni and Matteucci 2019) and PCF-MVS (Kuhn, Lin, and Erdler 2019) employ superpixel for image segmentation and planarization of textureless areas. However, the reconstruction performance in textureless areas is contingent upon the actual segmentation and fitting of the superpixels. CLD-MVS (Li et al. 2020) incorporate a confidence estimator to interpolate unreliable pixels, but their definition way of the confidence makes the result susceptible to occlusion and highlights. MAR-MVS (Xu et al. 2020) leverages epipolar geometry to determine the optimal neighborhood images and scale for pixels, yet its fixed patch size limits its adaptability across various application scenarios. APD-MVS (Yuesong Wang et al. 2023) employs patches with adaptive deformation strategy and pyramid architecture, but the time consumption of its iterative process poses a challenge in large-scale datasets.

**Learning-based MVS Methods** Unlike traditional MVS methods that suffer from hand-crafted image features, learning-based MVS methods typically leverage convolutional neural networks to extract high-dimensional image features, thereby enabling a more rational 3D reconstruction. MVSNET (Yao et al. 2018) has pioneered the construction through introducing differentiable 3D cost volumes using deep neural network, enabling numerous methods for further research. Certain classic multi-stage methods, including Cas-MVSNet (Gu et al. 2020), utilize a coarse-to-fine strategy to refine and upscale depth from low-resolution, thereby reducing the cost volumes while expanding the receptive field. In terms of memory reduction, several methods like Iter-MVS (Wang et al. 2022a) leverage GRU to regulate the 3D cost volumes along the depth direction. Concerning feature extraction, AA-RMVSNET (Wei et al. 2021) aggregates multi-scale variable convolution for adaptive feature extraction. Additionally, MVSTER (Wang et al. 2022b) integrates the transformer architecture into MVS tasks to capture multi-dimensional attention feature information. Despite these advancements, it is worth noting that numerous learning-based MVS methods risk severe degradation when applied to target domains that deviate from the training set.

## Method

Given a series of input images $I = \{I_i | i = 1, ..., N\}$, each one with specific camera parameters $P_i = \{K_i, R_i, C_i\}$. Our goal is to estimate the depth map $D_i$ for each image and subsequently merge them into a 3D point cloud. Fig. 2 illustrates our overall pipeline, specific design of each component will be detailed in subsequent sections.

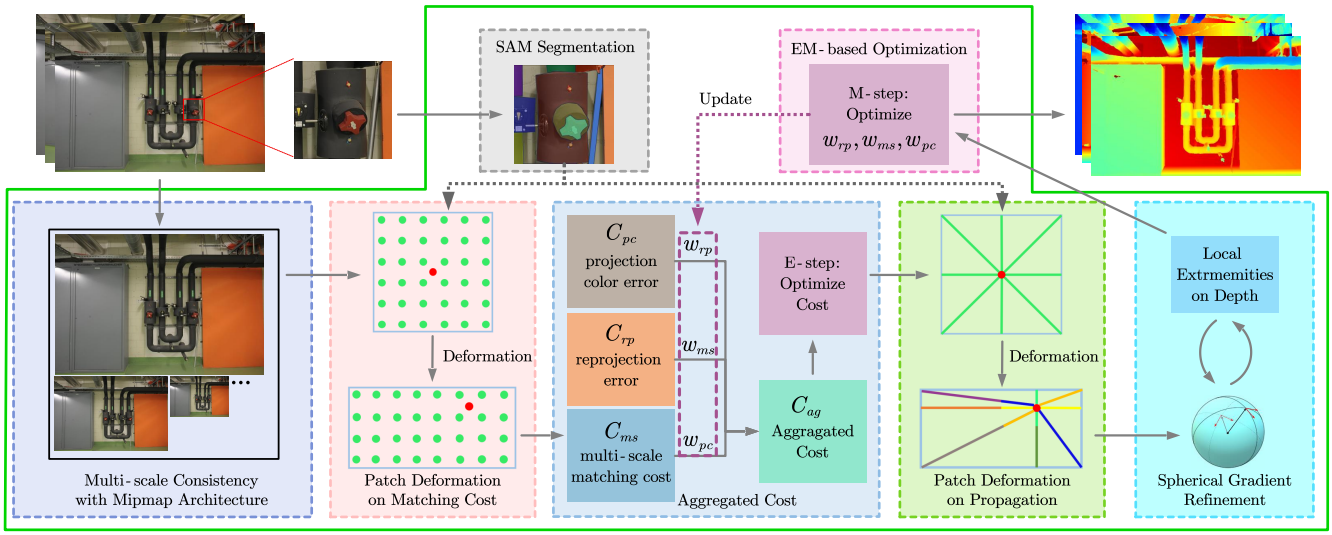



Figure 2: An illustrated pipeline of our proposed method. Images with multi views are initially downsampled and further allocated into our multi-scale architecture. Through leveraging the SAM-based segmentation, we carry out patch deformation on the matching cost to gain multi-scale matching costs $C_{ms}$. By integrating $C_{ms}$ with the projection color error $C_{pc}$ and the reprojection error $C_{rp}$, the aggregated cost is acquired. Then we again employ the SAM-based segmentation for patch deformation in propagation, succeeded by load-balancing within each search domain. Subsequently, we alternately iterates spherical gradient refinement on normals and pixelwise search interval on depths for enhanced accuracy. Finally, we employ EM-based optimization for the hyperparameter tuning of $w_{ms}$, $w_{rp}$, $w_{pc}$ and reassign them for the next iteration procedure.

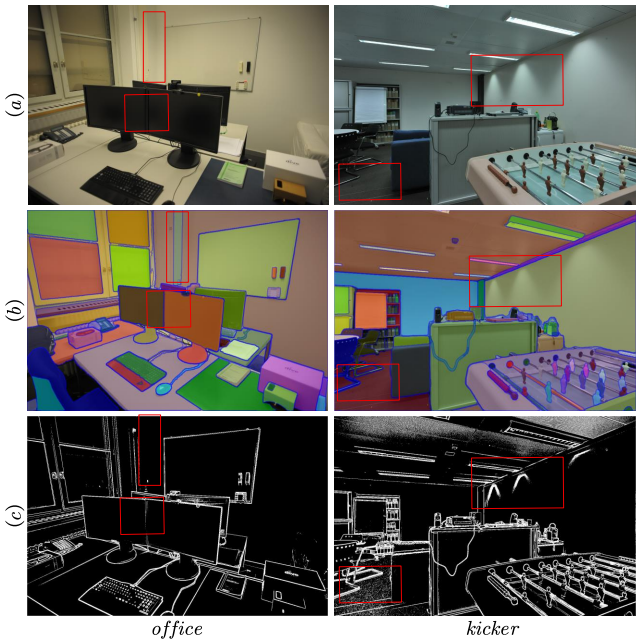



Figure 3: Comparative analysis of patch deformation strategies between the SAM-based instance segmentation and the Canny edge detection on partial scenes of ETH3D dadaset (*office* and *kicker*). From top to bottom, (a), (b) and (c) respectively show the original images, the SAM-based segmentation results and the Canny edge detection results. Representative areas in red boxes illustrate the advantages of SAM-based segmentation over Canny edge detection.

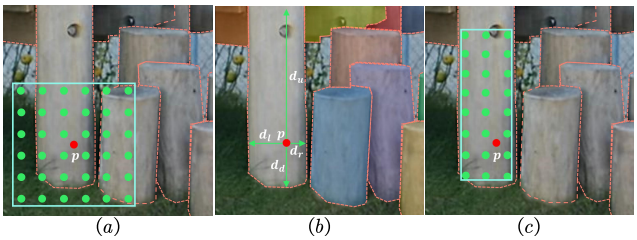
## Why Using Segment Anything Model?

The Segment Anything Model (SAM) can effectively discriminate between different instances, extracting subtle edge while neglecting strong illumination disturbances. To validate its effectiveness, we conduct the SAM-based instance segmentation and the Canny edge detection for patch deformation on partial scenarios of ETH3D datasets.

As shown in Fig. 3, when confronting with scenarios characterized by extensive similar colors and occlusion like *office*, SAM can effectively separate edges that exhibit similar colors on both sides with inconsistent depths, whereas Canny edge detection simply ignores them. Additionally, textureless areas like floors and walls in *kicker* can be effectively separated into different instances through SAM segmentation without illumination interference. In contrast, Canny edge detection incorrectly detects these illumination areas as edges, adversely affecting patch deformation.

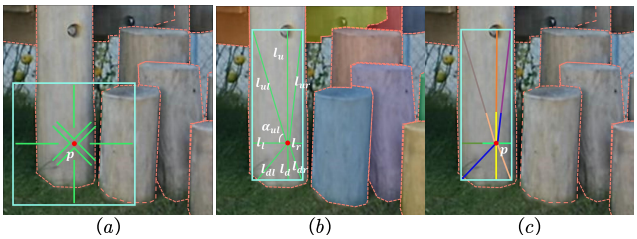
## Segmentation-Driven Patch Deformation

**Patch Deformation on Matching Cost** Some recent methods (Wang et al. 2021; Yuesong Wang et al. 2023) attempt to leverage patch deformation to improve matching cost or propagation scheme. As shown in Fig. 1, due to their insufficiency in exploiting edge information, they often cross boundary and reference areas with discontinuous depths, thereby yielding unsatisfactory results, especially when confronting with scenarios characterized by extensive similar colors and occlusions like forests and farmlands. Simultaneously, superpixel-based segmentation approaches (Romanoni and Matteucci 2019) also struggle in precisely




(a) (b) (c)

Figure 4: Patch deformation on matching cost. (a) is the matching cost scheme from ACMMP, (b) shows the distance of each directions and (c) illustrates the deformed patch.




(a) (b) (c)

Figure 5: Patch deformation on propagation. (a) is the propagation pattern of ACMMP, (b) depicts the length of each propagation branch, and (c) illustrates different search domains with different colors.

recognizing certain critical edges within these scenarios. They also lack instance semantic information to broaden receptive field, thereby meet pixelwise characteristic.

SAM segmentation can mitigate this issue as it separates different instances to extract subtle edges information while neglecting robust illumination disturbance. Consequently, we can leverage instance segmentation to better exploit and further introduce edge information into patch deformation. Specifically, we perform instance segmentation using SAM for input image $I_i$ to generate masks for diverse instances, denoted as $\mathcal{F}$. Hence we have $M = \mathcal{F}(I_i)$, where $M$ is an image mask whose size is consistent with $I_i$.

For each pixel $p$, we compute the bilateral weighted adaption of normalized cross correlation score (NCC) (Schönberger et al. 2016) between reference images $I_i$ and source image $I_j$, which can be calculated as follows:

$$\rho\left(p, \mathrm{W}_p^i\right) = \frac{cov\left(\mathrm{W}_p^i, \mathrm{W}_p^j\right)}{\sqrt{cov\left(\mathrm{W}_p^i, \mathrm{W}_p^i\right) cov\left(\mathrm{W}_p^j, \mathrm{W}_p^j\right)}} \quad (1)$$

where $cov$ is weighted covariance, $\mathrm{W}_p^i$ and $\mathrm{W}_p^j$ are respectively the corresponding images patches on image $I_i$ and $I_j$.

The goal of minimizing the matching cost is to obtain the optimal matching depths via the computation of color differences. However, when objects with varying depths exhibit similar colors, they are susceptible to generating matching inaccuracies, as shown in Fig. 4(a). Therefore, we introduce patch deformation to compute matching cost upon the sample patch W intersecting with different instances.

Specifically, we first measure the distances from the corresponding central pixel $p$ to the left, right, lower and upper boundaries of $M$, denoted respectively as $d_l$, $d_r$, $d_d$, and $d_u$. Then we can deform the shape of W to match these boundaries. The new shape of deformed patch can be defined as:

$$\left[\frac{d_l + d_r}{d_l + d_r + d_d + d_u} L, \frac{d_d + d_u}{d_l + d_r + d_d + d_u} L\right] \quad (2)$$

where $L$ denotes the side length of the square patch before patch deformation. Additionally, we reposition the patch's center by adding an offset:

$$\Delta o(p) = \left(\frac{d_l - d_r}{d_l + d_r} L_h, \frac{d_d - d_u}{d_u + d_d} L_v\right) \quad (3)$$

where $L_h$ and $L_v$ are respectively the horizontal and vertical length of deformed patch. The new center of the sample patch now becomes $p + \Delta o(p)$.

Both patch deformation and center offset allow pixels positioned at boundary regions to orient their patches more intensively towards the center of its own instance. Enhancing the receptive field for homogenous pixels in such approach can yield more robust results, consequently reducing potential errors in estimation. Note that considering the runtime, we restrict the number of calculations for each window such that the number of calculations after deformation never surpasses the initial total number $(L/2)^2$.

**Patch Deformation on Propagation** After SAM-based instance segmentation, pixels within the same instance typically exhibit similar depths, whereas noticeable depth discontinuities frequently arise at the boundaries between instances. Considering that propagation involves updating potential depths and normals within the surrounding area for each pixel, depth discontinuities will inevitably impact propagation. Consequently, we leverage patch deformation to adaptively alter the propagation scheme.

The adaptive checkerboard propagation scheme (Xu and Tao 2019) is conducted by introducing the optimal hypotheses from four near and four far search domains, as illustrated in Fig. 5 (a). However, his search domain between two adjacent diagonal directions is too dense, which leads to an imbalanced search space density and a risk of selecting redundant values. Hence we modify its oblique direction into a straight line extending to the corner of each patch.

Subsequently, we propose patch deformation on propagation via SAM, which adjusts the propagation patch shape and direction for each pixel. As illustrated in Fig. 5 (b), we adapt the propagation directions according to the shape of the surrounding mask. Specifically, denoting $l_l$, $l_r$, $l_d$, and $l_u$ as the length from the central pixel $p$ to the left, right, lower and upper edges of the patch, respectively, we obtain:

$$l_u = \frac{d_u}{d_u + d_d} L_v, l_l = \frac{d_l}{d_l + d_r} L_h \quad (4)$$

Both $l_r$ and $l_d$ can be obtained similarly. Therefore, the directions and lengths of slanted branch $l_{ul}$ is given by:

$$l_{ul} = \sqrt{{l_u}^2 + {l_l}^2}, \alpha_{ur} = \arctan\left(\frac{l_u}{l_l}\right) \quad (5)$$

where $l_{ul}$ refers to the length of the up-right branch, and $\alpha_{ur}$ represents the angle between the upward branch and the up-right branch. Corresponding lengths and directions of other three slanted branches can be obtained similarly.

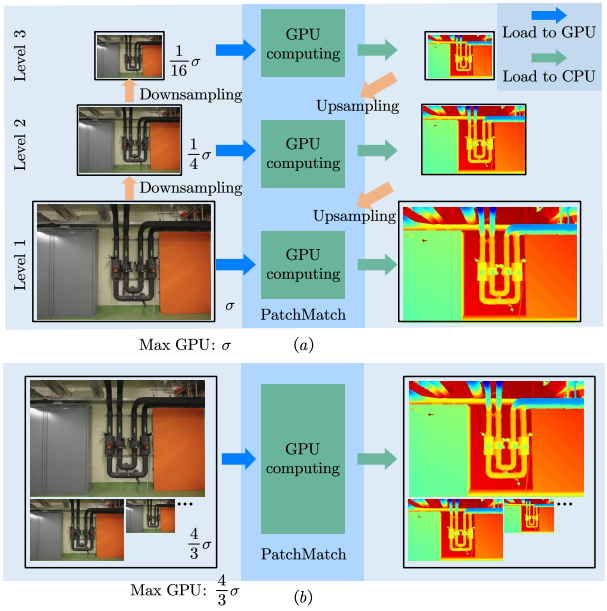



Figure 6: Different design architectures between ACMMP and our method. (a) illustrates the cascading network architectures employed in ACMMP, whereas (b) depicts our method with multi-scale architecture.

Having adjusted all directions and lengths, we encounter another challenge: the searching domain for each branch is unbalanced. Since the process of selecting a pixel with the minimal cost is essentially a spatial neighborhood search, an imbalance will emerge due to the different length of branches. The search along a shorter branch is suffered from unreliable results due to its minor search domain.

To address this, we accordingly modify the searching strategy in the propagation scheme, as shown in Fig. 5 (c). Specifically, we employ eight different colors to depict separate search domains on the eight directions centered on $p$. Instead of taking the central pixel $p$ as the dividing point, we use the midpoint of the sum of the lengths of two opposite branches to divide the search domain. In experiments, pixels with the same color are grouped into the same domain, with CUDA operators balance the load of searching for minima within each color-specific region. Therefore, our proposed strategy ensures load-balance across all directions and allows for faster convergence.

**Multi-scale Consistency** Many conventional methods adopt cascading architectures by sequentially loading different scales of images into GPU, as shown in Fig. 6 (a). This may result in a time-consuming performance due to the limited transfer speed between CPU and GPU. Therefore, we draw inspiration from mipmap (Williams 1983) in computer graphics, a technique to load different scales of images in parallel at once, to replace the previous cascading architecture into our proposed parallel architecture.

Specifically, we first perform image downsampling in the CPU. Subsequently, multi-scale images are assembled and loaded together into the GPU, as depicted in Fig. 6 (b). Then multi-scale images are processed together through matching cost, propagation and refinement in the GPU. Finally, all predicted depth images are transferred back into the CPU. Denoting the maximum memory consumption of ACMMP cascading architectures as $\sigma$, and the number of memory read operations as $k$, this technique enables us to load all scales of images in the GPU memory at a reasonable cost of $\frac{4}{3}\sigma$ instead of sequentially loading images, thereby eliminating the need for $k-1$ additional memory read operations.

Based on this architerture, we further introduce multi-scale consistency on matching cost and propagation. Regarding matching cost, we first apply SAM segmentation on the $k$-th level downsampled image. Based on segmentation results, we construct deformed patch and further compute $k$-th level matching cost, denoted as $c_k$. Therefore, the multi-scale matching cost is given by:

$$C_{ms} = \frac{\sum_k c_k}{k} \tag{6}$$

Concerning with propagation, the multi-scale consistency aggregates the search domain for all scales in each direction, yielding a total of eight distinct search domains. Conclusively, eight values with the lowest cost within each domain are chose as new hypothesis for further computation.

**Aggregated Cost** During the patch-matching phase, we consider not only the multi-scale matching cost $C_{ms}$, but also the reprojection error $C_{rp}$ and the projection color gradient error $C_{pc}$. $C_{rp}$ proposed in ACMMP validates depth estimation from geometric consistency. $C_{pc}$ measures color consistency between current pixel $p_i$ in reference image $I_i$ and its corresponding pixel $p_j$ in source images $I_j$:

$$C_{pc} = \max\left\{\left\|\nabla I_j\left(p_j\right) - \nabla I_i\left(p_i\right)\right\|, \tau\right\} \tag{7}$$

where $\nabla$ represents the Laplacian Operator, $p_j$ denotes pixel in image $I_j$ the projected by pixel $p_i$ in $I_i$, and $\tau$ is the truncation threshold to robustify the cost against outliers. With these terms, our the aggregated costs $C_{ag}$ can be given by:

$$C_{ag} = w_{ms}C_{ms} + w_{rp}C_{rp} + w_{pc}C_{pc} \tag{8}$$

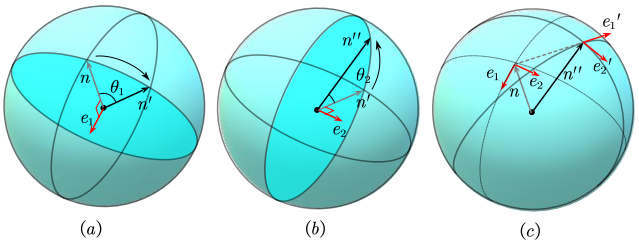
where $w_{ms}$, $w_{rp}$, and $w_{pc}$ respectively represent the aggregation weights of each component.

## Spherical Gradient Refinement

Two types of refinement strategies are adopted in ACMMP: 1. **Local perturbations**, which is the local search conduct by perturbing the current depth and normal with a small value; 2. **Random selection**, which achieves global search to suit potential depth discontinuities by assigning a random value. Since the edge information has already been segmented out through SAM, we only need to consider local perturbations. Given depth $d$ and normal $n = (n_x, n_y, n_z)$ in Cartesian coordinates, new depth $d'$ and normal $n'$ after the local perturbation can be defined by:

$$\begin{cases} d' \leftarrow d + \delta_d \\ n' \leftarrow \mathbf{VN}\left(n_x + \delta_x, n_y + \delta_y, n_z + \delta_z\right) \end{cases} \tag{9}$$

where $\mathbf{VN}$ is a normalization function ensuring $\|n'\| = 1$, and $\delta$ denotes a random value chosen from a fixed interval.




(a) (b) (c)

Figure 7: Spherical Gradient Refinement Procedure. (a) illustrates the rotation from $n$ to $n'$, (b) illustrates the rotation from $n'$ to $n''$. (c) respectively indicates two old and new orthogonal perturbation directions $e_1$, $e_2$ and $e_1'$, $e_2'$.

However, the strategy is incompatible with the definition of normal. It introduces a higher sensitivity to axes with smaller values during the search process, resulting in an unequal ratio of change on $xyz$ axes. Therefore, we propose the spherical gradient descent refinement, which utilize a structured representation to converge more accurate hypotheses.

**Spherical Coordinate** As shown in Fig. 7, given the normalized normal, we first randomly choose two orthogonal vectors, $e_1$ and $e_2$, perpendicular to the normal $n$ as the perturbation direction. We then use the angles $\theta_1$ and $\theta_2$ as the degree of rotation for iterative refinement. The normal first undergoes a counterclockwise rotation by $\theta_1$ degrees around $e_1$ as the rotation axis. Subsequently, the normal is further rotated counterclockwise by $\theta_2$ degrees around $e_2$ as the rotation axis. According to Rodrigues' rotation formula, the ultimate updated normal $n''$ is given by:

$$\begin{cases} n' = cos\theta_1 \cdot n + sin\theta_1(e_1 \times n) \\ n'' = cos\theta_2 \cdot n' + sin\theta_2(e_2 \times n') \end{cases} \tag{10}$$

This is analogous to sliding a vertex directed by the normal on the surface of a sphere, which ensures the preservation of normalization for the normal vector both before and after rotation. By finding two orthogonal bases perpendicular to the normal for refinement, it can be ensured that perturbations in each direction are equivalent. This approach aligns more closely with the geometric essence of the normal, which is defined on a sphere rather than individual axes in the $xyz$ coordinate system. As a result, our approach boosts the robustness and stability during the refinement process.

**Gradient Descent** We also utilize gradient descent in our method. The primary merit of gradient descent lies in its ability to logically restrict the search space to the vicinity of probable solutions. Denoting the number of total iterations as $N_{max}$, the rotation angle $\theta$ for the $i$-th round is randomly selected from range $\left[0, 5 * 2^{N_{max}-i}\right]$. After one round of refinement for depth $d$ and normal $n$, we determine the new direction for local perturbations $e_1'$ and $e_2'$ based on the result of the previous search. As such, we get:

$$\begin{cases} e_1' \leftarrow n'' - n \\ e_2' \leftarrow e_1' \times n'' \end{cases} \tag{11}$$

Here, $e_1'$ is aligned with the vector sum of the previous round's perturbation, while $e_2'$ is a vector perpendicular to both $n'$ and $e_1'$, as shown in Fig. 7(c). The primary merit of gradient descent lies in its ability to restrict the search domain of neighbourhood solutions. Each round of search takes place on the orthogonal plane defined by the previous search direction and the current normal direction, thereby enabling faster convergence to the optimal solution.

**Pixelwise Depth Interval Search** ACMMP employs a fixed interval for local perturbations on depth, while static perturbation range cannot adapt well to locally varying scene depth. Addressing this, we introduce pixelwise depth search interval chosen within the deformed patch.

Specifically, for each pixel, we extract the depth values of all pixels encompassed by its deformed patch, and choose the maximal and minimal values from this set as depth boundary for perturbations. Additionally, considering our iterative refinement strategy, during the $i$-th iteration, the pixelwise search interval is chosen within the deformed patch gained from $i$-th downsampled image, thereby narrowing the perturbation interval to yield more accurate hypothesis.

## EM-based Hyperparameters Optimization

While computing the aggregated matching cost, the hyperparameters of each component is typically determined empirically, which may result in suboptimal outcomes for different scenes. To mitigate this, we leverage the Expectation-Maximization (EM) algorithm to alternately optimize the hyperparameters and the aggregated cost, thereby enhancing both the robustness and effectiveness of our method.

**E-Step: Optimize $C_{ag}$** By fixing $w_{ms}$, $w_{rp}$, and $w_{pc}$, we can optimize the aggregated cost $C_{ag}$, formulated as:

$$\min_{C_{ms},C_{rp},C_{pc}} C_{ag} = w_{ms}C_{ms} + w_{rp}C_{rp} + w_{pc}C_{pc} \tag{12}$$

After optimization, we can get the optimal depth estimation under current hyperparameters.

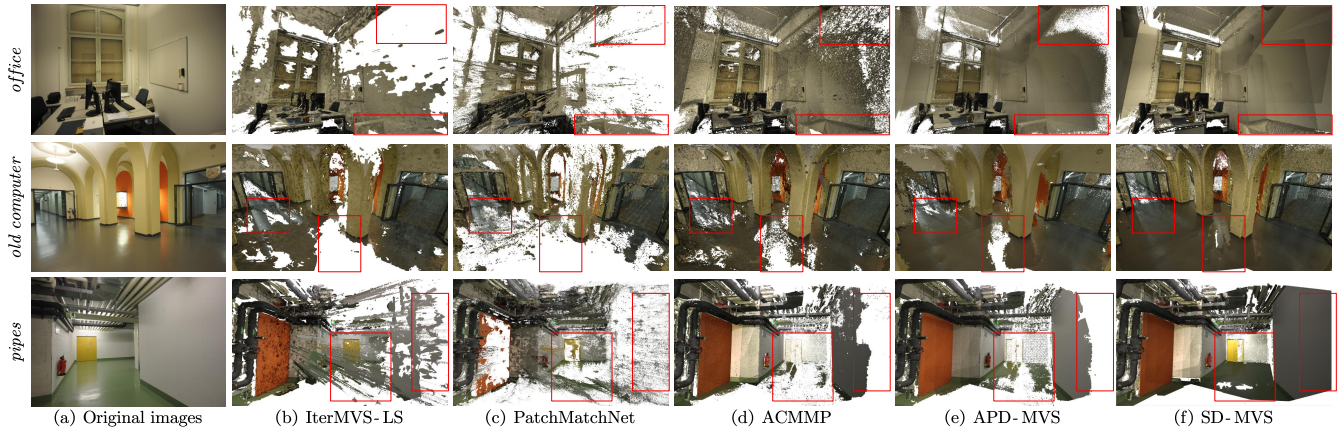
**M-Step: Optimize $w_{ms}$, $w_{rp}$, $w_{pc}$** By fixing $C_{ms}$,$C_{rp}$ and $C_{pc}$, we can optimize $w_{ms}$, $w_{rp}$ and $w_{pc}$, defined by:

$$\begin{aligned} \min_{w_{ms},w_{ms},w_{pc}} \quad & C_{ag} = w_{ms}C_{ms} + w_{rp}C_{rp} + w_{pc}C_{pc}, \\ s.t. \quad & w_{ms} + w_{rp} + w_{pc} = 1, \\ & w_{ms}, w_{rp}, w_{pc} > \eta \end{aligned} \tag{13}$$

All hyperparameters are required to exceed a minimal value $\eta$, and we implement a normalization constraint ensuring that their sum equals 1 to mitigate significant variances. Following the E-step optimization, we can alternatively optimize the hyperparameters and feed them back into the E-step for the next round of aggregated cost optimization.

Since it may be challenging to obtain the analytical solution to the optimization problem in M-step, we will use numerical optimization methods such as Newton's method (Qi and Sun 1993) to obtain the optimal solutions for $w_{ms}$, $w_{rp}$, and $w_{pc}$. A comprehensive formula derivation of the optimization can be found in supplementary material.

In practical situations, there might be partial pixels with depth estimation errors when all pixels are selected. Hence, we only select pixels where SIFT features can be matched between different images, and then calculate the aggregate cost between the pixels corresponding to these features.




(a) Original images (b) IterMVS-LS (c) PatchMatchNet (d) ACMMP (e) APD-MVS (f) SD-MVS

Figure 8: An illustration of the qualitative results on partial scenes of ETH3D datasets (*office*, *old computer*, and *pipes*). Some challenging areas are shown in red boxes. It is obvious that our methods outperform others, especially in large textureless areas.

| Method | Train | | | Test | | |
|---|---|---|---|---|---|---|
| | Acc. | Comp. | $F_1$ | Acc. | Comp. | $F_1$ |
| PatchMatchNet | 64.81 | 65.43 | 64.21 | 69.71 | 77.46 | 73.12 |
| IterMVS-LS | 79.79 | 66.08 | 71.69 | 84.73 | 76.49 | 80.06 |
| MVSTER | 68.08 | 76.92 | 72.06 | 77.09 | 82.47 | 79.01 |
| EPP-MVSNet | 82.76 | 67.58 | 74.00 | 85.47 | 81.79 | 83.40 |
| EPNet | 79.36 | 79.28 | 79.08 | 80.37 | **87.84** | 83.72 |
| COLMAP | **91.85** | 55.13 | 67.66 | **91.97** | 62.98 | 73.01 |
| PCF-MVS | 84.11 | 75.73 | 79.42 | 82.15 | 79.29 | 80.38 |
| MAR-MVS | 81.98 | 77.19 | 79.21 | 80.24 | 84.18 | 81.84 |
| ACMP | 90.12 | 72.15 | 79.79 | 90.54 | 75.58 | 81.51 |
| ACMMP | 90.63 | 77.61 | 83.42 | 91.91 | 81.49 | 85.89 |
| APD-MVS | 89.14 | **84.83** | 86.84 | 89.54 | 85.93 | 87.44 |
| SD-MVS (ours) | 89.63 | 84.52 | **86.94** | 88.96 | 87.49 | **88.06** |

Table 1: Quantitative results on ETH3D benchmark at threshold $2cm$ . Our method accomplishes the best $F_1$ score.

| Method | Intermediate | | | Advanced | | |
|---|---|---|---|---|---|---|
| | Pre. | Rec. | $F_1$ | Pre. | Rec. | $F_1$ |
| PatchMatchNet | 43.64 | 69.37 | 53.15 | 27.27 | 41.66 | 32.31 |
| CasMVSNet | 47.62 | 74.01 | 56.84 | 29.68 | 35.24 | 31.12 |
| IterMVS-LS | 47.53 | 74.69 | 56.94 | 28.70 | 44.19 | 34.17 |
| MVSTER | 50.17 | 77.50 | 60.92 | 33.23 | 45.90 | 37.53 |
| EPP-MVSNet | 53.09 | 75.58 | 61.68 | **40.09** | 34.63 | 35.72 |
| EPNet | **57.01** | 72.57 | **63.68** | 34.26 | **50.54** | **40.52** |
| COLMAP | 43.16 | 44.48 | 42.14 | 31.57 | 23.96 | 27.24 |
| PCF-MVS | 49.82 | 65.68 | 55.88 | 34.52 | 35.36 | 35.69 |
| ACMP | 49.06 | 73.58 | 58.41 | 34.57 | 42.48 | 37.44 |
| ACMMP | 53.28 | 68.50 | 59.38 | 33.79 | 44.64 | 37.84 |
| APD-MVS | 55.58 | 75.06 | 63.64 | 33.77 | 49.41 | 39.91 |
| SD-MVS (ours) | 53.78 | **77.63** | 63.31 | 35.53 | 47.37 | 40.18 |

Table 2: Quantitative results on TNT dataset. Our method accomplishes competitive $F_1$ score with SOTA methods.

# Experiments

## Datasets and Implementation Details

We evaluate our work on both ETH3D high-resolution benchmark (Schöps et al. 2017) and Tanks and Temples benchmark (TNT) (Knapitsch et al. 2017). We compare our work against state-of-the-art learning-based methods including PatchMatchNet (Wang et al. 2021), IterMVS-LS (Wang et al. 2022a), MVSTER (Wang et al. 2022b), EPP-MVSNet (Ma et al. 2021), EPNet (Su and Tao 2023) and traditional MVS methods including COLMAP (Schönberger et al. 2016), PCF-MVS (Kuhn, Lin, and Erdler 2019), MAR-MVS (Xu et al. 2020), ACMP (Xu and Tao 2020), ACMMP (Xu et al. 2022) and APD-MVS (Yuesong Wang et al. 2023).

Note that experiments is carried out on downsampled images with half of the original resolution in ETH3D, and on original images in TNT. Concerning parameter setting, $\{w_{ms}, w_{rp}, w_{pc}, L, k, \tau, N_{max}, \eta\} = \{1, 0.2, 0.2, 11, 3, 2, 3, 0.1\}$. In cost calculation, we take the matching strategy of every other row and column.

Our method is implemented on a system equipped with an Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz and an NVIDIA GeForce RTX 3080 graphics card. We take ACMP (Xu and Tao 2020) as the backbone of our method.

## Results on ETH3D and TNT

Qualitative results on ETH3D are illustrated in Fig. 8. It is obvious that our method reconstructs the most comprehensive results, especially in large textureless areas like floors, walls and doors, without introducing conspicuous detail distortion. More qualitative results on ETH3D and TNT benchmark can be referred in supplementary material.

Tab. 1 and Tab. 2 respectively present quantitative results on the ETH3D and the TNT benchmark. Note that the first group is learning-based methods and the second is traditional methods. Meanwhile, the best results are marked in bold while the second-best results are underlined. Our method achieves the highest $F_1$ score on ETH3D datasets, giving rise to state-of-the-art performance. Meanwhile, our method achieves competitive results with SOTA methods in TNT datasets like EPNET and APD-MVS, falling short by less than $0.5\%$ in $F_1$ score. Especially, our method shows significant improvement in completeness in both datasets, demonstrating its robustness in recovering textureless areas.

| Method | $2cm$ | | | $10cm$ | | |
|---|---|---|---|---|---|---|
| | Acc. | Comp. | $F_1$ | Acc. | Comp. | $F_1$ |
| w/. ACM. Cost | **90.16** | 74.61 | 81.27 | **98.01** | 89.04 | 93.16 |
| w/o. Adp. Cost | 89.92 | 78.01 | 83.42 | 97.92 | 91.87 | 94.71 |
| w/o. Mul. Cost | 89.84 | 79.94 | 84.55 | 97.9 | 93.36 | 95.53 |
| w/. ACM. Pro. | 89.83 | 79.96 | 84.52 | 97.91 | 93.58 | 95.54 |
| w/o. Adp. Pro. | 89.57 | 81.74 | 85.38 | 97.81 | 94.96 | 96.29 |
| w/o. Mul. Pro. | 89.69 | 81.97 | 85.54 | 97.87 | 95.17 | 96.44 |
| w/o. Ref. | 86.75 | 70.45 | 77.6 | 97.04 | 85.37 | 90.72 |
| w/. Gip. Ref. | 89.3 | 78.51 | 83.43 | 97.74 | 91.56 | 94.48 |
| w/. ACM. Ref. | 89.42 | 79.83 | 84.25 | 97.79 | 92.64 | 95.11 |
| w/o. EM A | 89.74 | 78.16 | 83.45 | 97.89 | 91.78 | 94.57 |
| w/o. EM B | 89.45 | 79.87 | 84.27 | 97.81 | 93.05 | 95.3 |
| SD-MVS | 89.63 | **84.52** | **86.94** | 97.85 | **96.74** | **97.28** |

Table 3: Quantitative results of the ablation studies on ETH3D benchmark to validate each proposed component.

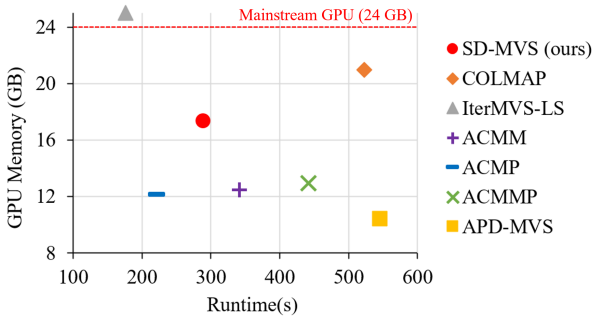



Figure 9: GPU memory usage (GB) and runtime (second) between different methods on ETH3D training datasets.

## Memory and Runtime Comparison

To demonstrate the efficiency of our method, we compare both GPU memory usage and runtime among various methods on ETH3D training datasets, as depicted in Fig. 9. Note that all experiments are executed on original images whose number have been standardized to 10 across all scenes. Moreover, to exclude the impact of unrelated variables, all methods are conducted on a same system, whose hardware configuration has been specified in previous section.

Concerning learning-based methods, while IterMVS-LS exhibits the shortest runtime, its memory overhead exceeds the maximum capacity of mainstream GPUs. Other state-of-the-art (SOTA) learning-based methods also suffer from excessive memory consumption, making them impractical for the reconstruction of large-scale outdoor scenarios.

Although SD-MVS consumes approximately one-third more memory usage than traditional SOTA methods like APD-MVS and ACMMP, our runtime is only half of them, thanks to our multi-scale consistency architecture. Therefore, our method strikes the optimal balance between time and memory usage without sacrificing performance, demonstrating its effectiveness and practicality.

## Ablation Studies

We validate the rationale behind the design of each part of our method through ablation studies, as shown in Tab. 3.

**Matching Cost with Adaptive Patch** In terms of matching cost, we respectively remove patch deformation (w/o. Adp. Cost), multi-scale consistency (w/o. Mul. Cost) and both of them (w/. ACM. Cost). Since w/. ACM. Cost has neither deformable nor multi-scale, it produces the worst results. w/o. Mul. Cost slightly outperformed w/o. Adp. Cost, yet both are inferior to SD-MVS, implying that patch deformation contribute more than multi-scale consistency.

**Adaptive Propagation with Load-balancing** In terms of propagation, we respectively remove patch deformation (w/o. Adp. Pro.), multi-scale consistency (w/o. Mul. Pro.) and apply propagation scheme from ACMMP (w/. ACM. Pro.). Given that patches in ACMMP do not deform in accordance with the patch, its performance fell short of expectations. Both w/o. Adp. Pro. and w/o. Mul. Pro. delivered similar results, yet fell short in comparison to SD-MVS, indicating that both patch deformation and multi-scale consistency on propagation are equally crucial.

**Spherical Gradient Refinement** In terms of refinement, we respectively remove refinement (w/o. Ref.), exchange the refinement module into Gipuma (Galliani, Lasinger, and Schindler 2015) (w/. Gip. Ref.) and switch the refinement module into ACMMP (w/. ACM. Ref.). As observed, the absence of refinement significantly diminishes the results. However, introducing Gipuma refinement brings about noticeable progress, with further advancements achieved after adopting ACMMP refinement. Nonetheless, both refinement methods are worse than SD-MVS, proving the necessity of spherical gradient refinement.

**EM-based Hyperparameters Optimization** We conduct two experiments (w/o. EM A and w/o. EM B) by removing EM-based Optimization and respectively setting $(w_{ms}, w_{rp}, w_{pc})$ to $(1, 0.5, 0.5)$ and $(1, 0.2, 0.2)$. The results highlight the impact of hyperparameter settings on the final results. Furthermore, their inferior performances compared to SD-MVS evidences the importance of automatic parameter tuning by the proposed EM-based Optimization.

## Conclusion

In this paper, we presented SD-MVS, a novel MVS method designed to effectively address challenges posed by textureless areas. The proposed method consists of an adaptive patch deformation with multi-scale consistency, a spherical gradient refinement and EM-based hyperparameter optimization. Our method has achieved state-of-the-art performance on ETH3D high-resolution benchmark, while being memory-friendly and with less time cost. In the future, we will tackle difficulty in highlight areas in matching cost and view selection strategy in pursuit of superior performance.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 62172392, the Central Public-interest Scientific Institution Basal Research Funds(No. Y2022QC17) and the Innovation Research Program of ICT CAS (E261070).

## References


Bleyer, M.; Rhemann, C.; and Rother, C. 2011. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 14.1–14.11.

Cao, M.; Zheng, L.; Jia, W.; Lu, H.; and Liu, X. 2021. Accurate 3-D Reconstruction Under IoT Environments and Its Applications to Augmented Reality. *IEEE Trans. Ind. Inf.*, 17(3): 2090–2100.

Cremers, D.; and Kolev, K. 2011. Multiview Stereo and Silhouette Consistency via Convex Functionals over Convex Domains. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6): 1161–1174.

Galliani, S.; Lasinger, K.; and Schindler, K. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 873–881.

Gu, X.; Fan, Z.; Zhu, S.; Dai, Z.; Tan, F.; and Tan, P. 2020. Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2492–2501.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. Graph.*, 36(4).

Kuhn, A.; Lin, S.; and Erdler, O. 2019. Plane Completion and Filtering for Multi-View Stereo Reconstruction. In *Proc. DAGM German Conf. (GCPR)*, volume 11824, 18–32.

Lee, J. Y.; DeGol, J.; Zou, C.; and Hoiem, D. 2021. PatchMatch-RL: Deep MVS with Pixelwise Depth, Normal, and Visibility. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 6138–6147.

Li, Z.; Gogia, P. C.; and Kaess, M. 2019. Dense Surface Reconstruction from Monocular Vision and LiDAR. In *Proc. IEEE Conf. Robot. Automat. (ICRA)*, 6905–6911.

Li, Z.; Zuo, W.; Wang, Z.; and Zhang, L. 2020. Confidence-Based Large-Scale Dense Multi-View Stereo. *IEEE Trans. on Image Process.*, 29: 7176–7191.

Ma, X.; Gong, Y.; Wang, Q.; Huang, J.; Chen, L.; and Yu, F. 2021. EPP-MVSNet: Epipolar-assembling based Depth Prediction for Multi-view Stereo. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 5712–5720.

Orsingher, M.; Zani, P.; Medici, P.; and Bertozzi, M. 2022. Revisiting PatchMatch Multi-View Stereo for Urban 3D Reconstruction. In *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 190–196.

Qi, L.; and Sun, J. 1993. A nonsmooth version of Newton's method. *Math. Program.*, 58: 353–367.

Romanoni, A.; and Matteucci, M. 2019. TAPA-MVS: Textureless-Aware PAtchMatch Multi-View Stereo. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 10412–10421.

Schöps, T.; Schönberger, J. L.; Galliani, S.; Sattler, T.; Schindler, K.; Pollefeys, M.; and Geiger, A. 2017. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*.

Schönberger, J. L.; Zheng, E.; Frahm, J.-M.; and Pollefeys, M. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, volume 9907, 501–518.

Seitz, S.; Curless, B.; Diebel, J.; Scharstein, D.; and Szeliski, R. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, volume 1, 519–528.

Su, W.; and Tao, W. 2023. Efficient Edge-Preserving Multi-View Stereo Network for Depth Estimation. In *Proc. of the AAAI Conf. Artif. Intell. (AAAI)*, 2348–2356.

Vogiatzis, G.; Hernandez Esteban, C.; Torr, P. H.; and Cipolla, R. 2007. Multiview Stereo via Volumetric Graph-Cuts and Occlusion Robust Photo-Consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12): 2241–2246.

Wang, F.; Galliani, S.; Vogel, C.; and Pollefeys, M. 2022a. IterMVS: Iterative Probability Estimation for Efficient Multi-View Stereo. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 8596–8605.

Wang, F.; Galliani, S.; Vogel, C.; Speciale, P.; and Pollefeys, M. 2021. PatchmatchNet: Learned Multi-View Patchmatch Stereo. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 14189–14198.

Wang, X.; Zhu, Z.; Huang, G.; Qin, F.; Ye, Y.; He, Y.; Chi, X.; and Wang, X. 2022b. MVSTER: Epipolar Transformer for Efficient Multi-view Stereo. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, volume 13691, 573–591.

Wei, Z.; Zhu, Q.; Min, C.; Chen, Y.; and Wang, G. 2021. AA-RMVSNet: Adaptive Aggregation Recurrent Multi-view Stereo Network. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 6167–6176.

Williams, L. 1983. Pyramidal Parametrics. In *Proc. of the 10th Annu. Conf. on Comput. Graph. and Interact. Techn. (SIGGRAPH)*, 1–11.

Xu, Q.; Kong, W.; Tao, W.; and Pollefeys, M. 2022. Multi-Scale Geometric Consistency Guided and Planar Prior Assisted Multi-View Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–18.

Xu, Q.; and Tao, W. 2019. Multi-Scale Geometric Consistency Guided Multi-View Stereo. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 5478–5487.

Xu, Q.; and Tao, W. 2020. Planar Prior Assisted PatchMatch Multi-View Stereo. In *Proc. of the AAAI Conf. Artif. Intell. (AAAI)*, volume 34, 12516–12523.

Xu, Z.; Liu, Y.; Shi, X.; Wang, Y.; and Zheng, Y. 2020. MARMVS: Matching Ambiguity Reduced Multiple View Stereo for Efficient Large Scale Scene Reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 5980–5989.

Yan, J.; Wei, Z.; Yi, H.; Ding, M.; Zhang, R.; Chen, Y.; Wang, G.; and Tai, Y.-W. 2020. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 674–689.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. MVS-Net: Depth Inference for Unstructured Multi-view Stereo. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, volume 11212, 785–801.

Yao, Y.; Luo, Z.; Li, S.; Shen, T.; Fang, T.; and Quan, L. 2019. Recurrent MVSNet for High-Resolution Multi-View Stereo Depth Inference. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 5520–5529.

Yuesong Wang; Zhaojie Zeng; Tao Guan; Wei Yang; Zhuo Chen; Wenkai Liu; Luoyuan Xu; and Yawei Luo. 2023. Adaptive Patch Deformation for Textureless-Resilient Multi-View Stereo. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 1621–1630.