Revisiting Multi-View Stereo: A Sequence-to-Sequence Formulation

Aoxiang Fan **Corentin Dumery** **Nicolas Talabot** **Pascal Fua**
CVLab, EPFL
Switzerland
{aoxiang.fan, corentin.dumery, nicolas.talabot, pascal.fua}@epfl.ch

Abstract

Computing accurate geometry from multi-view images is a fundamental problem in computer vision. Recent feed-forward (FF) models jointly estimate 3D geometry and camera parameters, but they typically suffer from geometry distortion caused by reconstruction ambiguity, even when ground-truth camera parameters are supplied. In this paper, we study the multi-view stereo (MVS) problem with known camera parameters and propose a novel approach that bridges conventional MVS and FF methods. Rather than casting MVS as a sequence-to-one mapping that predicts depth only for a single reference view, we reformulate it as a sequence-to-sequence task, akin to FF models, that jointly predicts geometry for all input views. We introduce a global transformer-based architecture with two components that explicitly exploit camera-induced priors: ray-map embeddings that inject camera parameters into image patch tokens, making the transformer camera-aware, and a unified global cost volume that replaces conventional per-view cost volumes to jointly capture 3D structure across all views. Extensive experiments on multiple public benchmarks show our approach achieves state-of-the-art performance, surpassing both MVS and FF reconstruction baselines.

1 Introduction

3D reconstruction from multi-view images is a fundamental and long-standing problem in computer vision that has seen significant progress in recent years. Conventionally, the reconstruction task is addressed by Structure-from-Motion (SfM) and Multi-view Stereo (MVS), where both classical and learning-based methods have been developed. Classical methods [13, 14, 34, 39, 44, 51] are able to reconstruct very accurate 3D points by extracting image correspondences and 3D triangulation. However, this often results in sparse and incomplete reconstructions, requiring complex densification techniques in post-processing [13, 14, 51].

Learning-based MVS methods [6, 10, 21, 66] have been proposed to address this issue, leveraging deep learning to infer per-pixel dense depth maps directly from multi-view images. The key innovation of these methods is **cost volumes** as a camera-induced prior, which are constructed by regularly sampling 3D points along per-pixel depth dimension and store view-consistency information for each point. In this way, cost volumes offer very effective regularization for geometry estimation. Nevertheless, they are typically constructed on a per-view basis, effectively reducing the problem to a sequence-to-one formulation that predicts only the depth map of a reference view, even when multiple views are provided as input. The sequence-to-one design not only reduces efficiency but also restricts geometric estimation accuracy due to the lack of global alignment modeling.

Among the most notable advances in recent years are feed-forward (FF) models [4, 23, 28, 59, 60, 63, 65] designed for general settings and adopt a sequence-to-sequence formulation to jointly estimate camera parameters and depth/point maps for all input views. However, due to the inherent difficulty

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

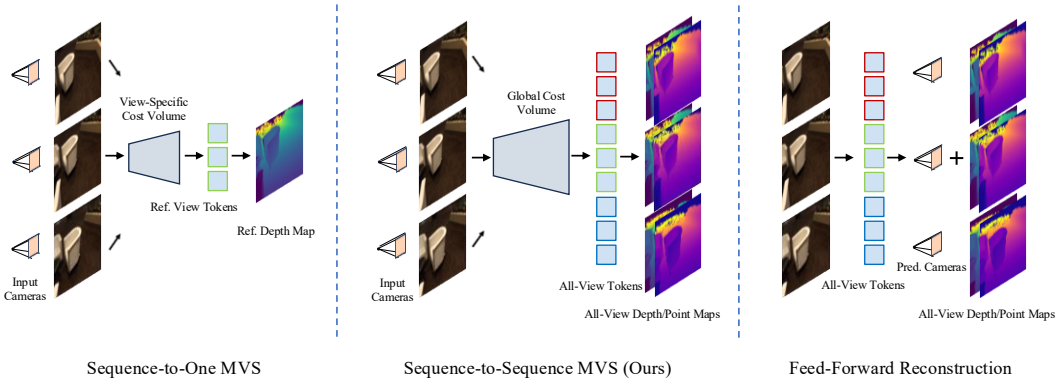


Figure 1: The differences of formulation between sequence-to-one multi-view stereo methods, our proposed sequence-to-sequence multi-view stereo method and feed-forward reconstruction methods.

of this joint estimation problem, the resulting reconstructions are often less accurate or more distorted. This remains true even when ground-truth camera parameters are provided, as the model still attempts to estimate and update them.

In this paper, we study the MVS problem that conditions on known camera parameters. These parameters provide greater control over the reconstruction process and can be readily obtained in practice using SfM or other camera calibration methods. Our approach bridges MVS and FF methods: like feed-forward models, it adopts a sequence-to-sequence formulation with a global Transformer-based architecture, while incorporating camera-induced priors to ensure geometric accuracy. Specifically, we consider two types of such priors. Firstly and naturally, we need to incorporate camera parameters into the transformer-based architecture to make the model camera-aware. Secondly, we propose to adapt the conventional cost-volume idea to our sequence-to-sequence setting, providing valuable regularization over the underlying 3D scene geometry. We highlight the differences of our formulation and the existing ones in Fig. 1.

To integrate these priors into our framework, we introduce two key designs. First, inspired by [28], we transform the camera intrinsics and extrinsics into ray maps, which are then tokenized and merged with image tokens to encode camera information. Second, we introduce a Global Cost Volume, which, unlike conventional cost volumes that have to be repeatedly constructed per-view, captures the global 3D structure of the scene across all views in a unified representation. We provide a diagram illustration of our method in Fig. 2.

Our model operates in a sequence-to-sequence manner, in contrast to existing sequence-to-one MVS methods that also incorporate camera-induced priors. Experimental results demonstrate that it achieves superior reconstruction results compared to both MVS approaches and recent sequence-to-sequence feed-forward models, establishing a new state of the art for multi-view reconstruction with known camera parameters.

2 Related Works

Structure-from-Motion. Structure-from-Motion (SfM) [18, 34, 34–36, 39, 44] aims to jointly estimate camera parameters and sparse 3D structure from multi-view images. Classical pipelines, such as COLMAP [44], Pix4D [39] and OpenMVG [34], decompose the problem into stages including feature extraction, matching, triangulation, and bundle adjustment, and remain strong baselines under well-constrained settings [3, 18, 30, 42]. Recent progress has focused on improving robustness through learned components, particularly in feature detection and matching. These include sparse matching pipelines with local feature detection [9, 54] and matching [29, 43, 68], and dense matching pipelines without feature detectors [11, 12, 50, 53]. Beyond modular pipelines, end-to-end differentiable approaches such as VGGSfM [58] jointly optimize camera poses and scene structure, improving performance in challenging scenarios. Recently, VidMap [37] has been proposed for video-based SfM by leveraging temporal information. Nevertheless, dense reconstruction remains a challenge in this line of work, subject to expensive and complex post-processing steps [7, 13, 14, 51].

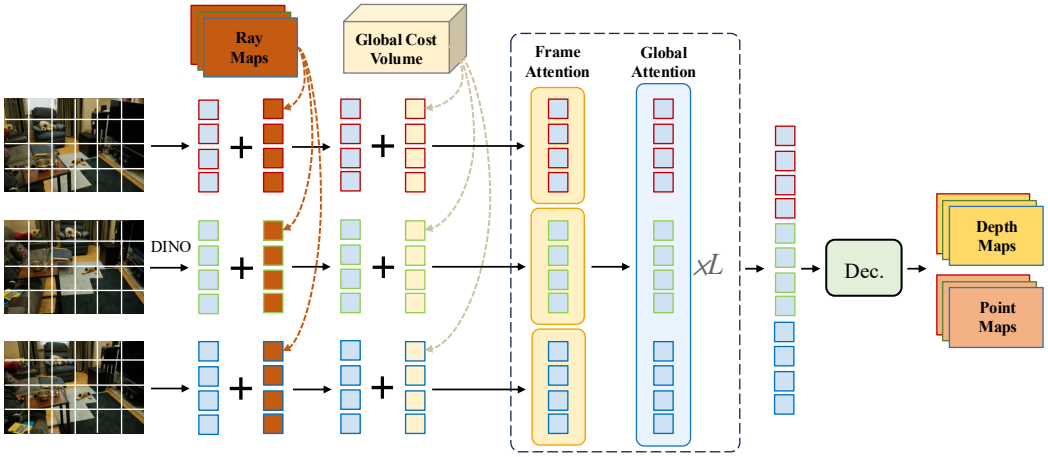


Figure 2: Overview of the proposed sequence-to-sequence multi-view stereo method. For each input view, we extract view tokens using DINOv2 backbone, and incorporate camera-induced priors into the model with ray-map embeddings and global-cost-volume embeddings. The tokens are then fed into recurring attention layers to aggregate frame and global information, which are fed to Decoding heads to produce depth maps and point maps of each input view.

Learning-based Multi-View Stereo. Recent MVS methods are mostly learning-based [6, 10, 17, 21, 31, 38, 57, 66], focusing on dense reconstruction given calibrated cameras typically provided by SfM. Traditional methods rely on handcrafted priors or global optimization, while learning-based approaches, such as MVSNet [66], leverage cost volumes and 3D convolutional networks to infer accurate depth maps. These models significantly improve reconstruction quality, particularly in challenging regions like texure-less structures or wide baselines. Recent work, including MVSA [21], emphasizes generalization across diverse scenes and viewing conditions. Despite these advances, most MVS methods construct cost volumes on a per-view basis, limiting their ability to capture global scene structure. Moreover, their sequence-to-one formulation restricts predictions to a reference view, leaving multi-view consistency to be handled implicitly.

Feed-Forward Reconstruction. Feed-Forward approaches reformulate multi-view reconstruction as a direct prediction problem using neural networks, often built on transformer architectures. Early works focus on pairwise inputs, with methods such as DUSt3R [61] directly predicting aligned point maps without requiring camera parameters, and MASt3R [26] improving geometric consistency and scale estimation. Multi-view feed-forward models, including VGGT [59], Fast3R [65], MapAnything [23], π^3 [63], DepthAnything3 [28], VGGT-Ω [60], and DVLT [4], aggregate features across views using global attention to jointly estimate camera parameters and scene geometry in a single forward pass. However, global attention incurs quadratic complexity with respect to the number of input views, limiting scalability. Approaches such as FastVGGT [47] and SparseVGGT [56] improve efficiency through token reduction or sparse attention. Despite these advances, jointly estimating camera parameters and geometry remains highly challenging, and such models are usually prone to reconstruction inaccuracies or distortion.

3 Method

In this section, we describe the proposed method, termed Sequence-to-Sequence MVS (S2S-MVS). We start by introducing the setting and formulating the addressed problem in Sec. 3.1, as well as explaining the basic architecture of our framework. Then, we separately explain how the two camera-induced priors can be incorporated into our basic architecture in Sec. 3.2 and Sec. 3.3.

3.1 Problem Formulation

Definitions and Notations. We consider as input a sequence of V RGB images $(I_i)_{i=1}^{V}$, where $I_i \in \mathbb{R}^{3\times H\times W}$, capturing a 3D scene. Differently from most feed-forward models, we follow MVS

setting to assume camera parameters are known, which are typically from SfM models and is the premise for studying camera-induced priors. Formally, our model's output is written as

$$f\left((I_i)_{i=1}^V, (\mathbf{g}_i)_{i=1}^V\right) = (D_i, P_i)_{i=1}^V, \tag{1}$$

where $\mathbf{g}_i$ denotes the known camera parameters of image I_i. For each frame I_i, the model predicts a depth map $D_i \in \mathbb{R}^{H\times W}$ and a point map $P_i \in \mathbb{R}^{3\times H\times W}$. The known **camera parameters** $\mathbf{g}_i$ include both extrinsics $(\mathbf{R}_i, \mathbf{t}_i)_{i=1}^V$ and intrinsics $(\mathbf{K}_i)_{i=1}^V$.

We define the image domain of I_i as $\mathcal{I}(I_i) = \{1, \ldots, H\} \times \{1, \ldots, W\}$, namely, the set of all pixel coordinates. The **depth map** D_i assigns to each pixel $\mathbf{y} \in \mathcal{I}(I_i)$ a positive scalar depth value $D_i(\mathbf{y}) \in \mathbb{R}^+$ with respect to the i-th camera. The **point map** P_i associates each pixel $\mathbf{y}$ with its corresponding 3D location $P_i(\mathbf{y}) \in \mathbb{R}^3$. Following VGGT [59], these point maps are defined in a *viewpoint-invariant* manner: all 3D points are expressed in the coordinate system of the first camera $\mathbf{g}_1$, which serves as the world reference frame.

For sequence-to-sequence modeling, our network employs a global transformer architecture, following existing feed-forward models. Next, we explain this base architecture.

Image Feature Encoding. This module produces powerful monocular feature representations for input images. Specifically, we adopt the ViT-L/16 encoder of DINOv3 [48], initialized with its pretrained weights. The model processes images using $k \times k$ patches with $k = 16$, yielding feature maps of spatial size $\frac{H}{k} \times \frac{W}{k}$. As a result, each image is decomposed into a collection of K token embeddings, denoted by $\mathrm{t}_i^I \in \mathbb{R}^{K\times C}$. We take these tokens from four intermediate layers of the encoder, which together with the cost volume, form the input tokens of the alternating-attention network, as explained in Sec. 3.3.

Frame and Global Attention. Consistent with existing works [4, 28, 59], our transformer architecture alternates between frame-wise self-attention and global self-attention. Frame-wise self-attention is applied to the tokens of each input image independently, which focuses on per-view scene information. Global self-attention is applied to all tokens across all input images, aggregating crucial multi-view information to facilitate reconstruction.

Dense Predictions. We take the processed tokens from four attention layers spread across the network, which are fused and progressively upsampled using learned convolutional filters. Following VGGT-Ω [60], we replace the blocks operating above 1/4 of the input resolution with a lightweight upsampling head via a single MLP followed by a pixel-shuffle operator to output full-resolution dense depth maps $(D_i)_{i=1}^V$, from which the point maps $(P_i)_{i=1}^V$ are obtained with the ray maps (Sec. 3.2). In addition to depth maps, the same head outputs a per-pixel aleatoric uncertainty map $\Sigma_i \in \mathbb{R}^{H\times W}$ for each view, obtained from an extra output channel. This uncertainty map reflects the model's confidence in its own depth and point predictions and is used to reweight the regression losses, as detailed next.

3.2 Camera Parameter Encoding

Camera parameters are typically expressed in the literature in two ways. In VGGT [59], camera tokens are defined for each frame and serve as special tokens that participate in both frame and global attentions. Alternatively, in DepthAnything3 [28], a depth-ray representation is used as the regression objective.

In our setting, we assume the camera parameters are known, which allows us to encode camera information from the beginning. We opt for a ray map embedding approach over regressed camera tokens, which directly injects information in a per-patch manner and stabilizes training.

For each pixel $\mathbf{y}$, we define its camera ray $\mathbf{r} \in \mathbb{R}^6$ as the concatenation of a ray origin $\mathbf{t} \in \mathbb{R}^3$ and a ray direction $\mathbf{d} \in \mathbb{R}^3$, i.e., $\mathbf{r} = (\mathbf{t}, \mathbf{d})$. The ray direction is computed by first backprojecting $\mathbf{y}$ into the camera coordinate system and then transforming it into the world frame:

$$\mathbf{d} = \mathbf{R}\mathbf{K}^{-1}\mathbf{y}.$$

where $\mathbf{R}$ and $\mathbf{t}$ represent the rotation and translation in world frame. By collecting the ray origins and directions over all pixels, we obtain a dense ray map $\mathbf{M} \in \mathbb{R}^{H\times W\times 6}$. Note that $\mathbf{d}$ is left unnormalized, allowing its norm to retain the corresponding projection scale. Given the predicted depth, the associated 3D point in world coordinates can then be recovered as

$$P = \mathbf{t} + D(\mathbf{y}) \cdot \mathbf{d}.$$

With this representation, point maps can be generated in a unified and efficient manner by combining the predicted depth and ray maps using element-wise operations.

Incorporation into Transformer. We apply an element-wise signed logarithm $\mathrm{sign}(\mathbf{M})\log(1+|\mathbf{M}|)$ to the ray map, patchify it with the image patch size and embed each patch linearly into ray tokens $\mathrm{r}^I = \bigcup_{i=1}^{V}\{\mathrm{r}_i^I\}$. These tokens are added to the patch tokens after the frame-wise attention of every layer ℓ, $\mathrm{t}^{(\ell)} \leftarrow \mathrm{t}^{(\ell)} + \tanh(\gamma_\ell)\,\mathrm{r}^I$, where the learnable scalar γ_ℓ is initialized to zero so that camera information is introduced gradually.

3.3 Cost Volume Construction

Cost volume is a pivotal module in most existing learning-based MVS methods, providing crucial camera-induced scene priors for multi-view reconstruction. Next, we will explain the conventional cost volume construction first, then turn to our proposed global cost volume.

3.3.1 Per-View Cost Volume

Conventionally, the construction of a cost volume needs a reference view and a set of hypothesized depth values $(d_i)_{i=1}^{T}$ to be predefined [6, 10, 21, 66]. To capture low-level features from images, a convolutional backbone is usually used for cost volume construction, giving feature maps $(\mathcal{F}_i)_{i=1}^{V}$. The cost volume is constructed by warping feature maps $\mathcal{F}_i$ from each source view to the reference one using hypothesized depth values $(d_i)_{i=1}^{T}$.

The warping is formulated as a standard 3D reprojection process. Given a pixel $\mathbf{y}$ in the reference image and a depth hypothesis d, its projected location $\hat{\mathbf{y}}$ in the source image is computed. Pair-wise feature correlation at position $\mathbf{y}$ is

$$c_i^{(d)}(\mathbf{y}) \;= < \mathcal{F}_0(\mathbf{y}), \hat{\mathcal{F}}_i^{(d)}(\mathbf{y}) >, \tag{2}$$

where $\hat{\mathcal{F}}_i^{(d)}$ denotes the warped i-th source feature map at depth d. The correlation values of each source view is then aggregated, for instance by taking an average, to a scalar cost volume in shape $T \times \frac{H}{s} \times \frac{W}{s}$, where s is the pixel-wise subsampling ratio.

3.3.2 Global Cost Volume

Repeating the per-view construction with every view as the reference makes the cost grow quadratically with V. We instead build a single global cost volume shared by all views.

Cost Value Computation. We drop the reference view and define a cost value for any 3D point $\mathbf{p}$ in the world frame. Projecting $\mathbf{p}$ into view i gives $\lambda_i\tilde{\mathbf{y}}_i = \mathbf{K}_i(\mathbf{R}_i\mathbf{p}+\mathbf{t}_i)$, where $\tilde{\mathbf{y}}_i$ are the homogeneous coordinates of the projection $\mathbf{y}_i$ and λ_i is the depth of $\mathbf{p}$. Let $\mathcal{V}(\mathbf{p})$ be the set of k views with $\lambda_i > 0$ and $\mathbf{y}_i$ inside the image, and $\mathbf{f}_i = \mathcal{F}_i(\mathbf{y}_i)$ the bilinearly sampled features. The cost value is the mean correlation over all pairs of distinct views that see $\mathbf{p}$:

$$c(\mathbf{p}) \;=\; \frac{1}{k(k-1)} \sum_{\substack{i,j\in\mathcal{V}(\mathbf{p})\\ i\neq j}} < \mathbf{f}_i, \mathbf{f}_j >, \tag{3}$$

with $c(\mathbf{p}) = 0$ if $k < 2$. It treats all views symmetrically, and since $\sum_{i\neq j} < \mathbf{f}_i, \mathbf{f}_j >= \|\sum_i \mathbf{f}_i\|^2 - \sum_i \|\mathbf{f}_i\|^2$, it can be accumulated one view at a time with memory independent of V.

Scene Representation in Contracted Space. As unbounded scenes offer no bounding box, we place $\mathbf{C}$ on a uniform grid in a contracted space. We back-project T log-spaced depth hypotheses $(d_t)_{t=1}^{T}$ at every pixel of every view, normalize these frustum samples by their median $\mathbf{o}$ and median distance ρ to it, and apply the L_∞ contraction of mip-NeRF 360 [2], which maps $\mathbb{R}^3$ onto $[-2,2]^3$:

$$\mathrm{contract}(\mathbf{x}) = \mathbf{x} \;\text{ if } \|\mathbf{x}\|_\infty \le 1, \quad \left(2 - \tfrac{1}{\|\mathbf{x}\|_\infty}\right)\tfrac{\mathbf{x}}{\|\mathbf{x}\|_\infty} \;\text{ otherwise}, \qquad \mathbf{x} = \tfrac{\mathbf{p}-\mathbf{o}}{\rho}. \tag{4}$$

$\mathbf{C}$ is obtained by evaluating Eq. (3) on a uniform grid over $[-2,2]^3$ whose size is independent of V, and each view-specific cost volume $\mathbf{C}_i \in \mathbb{R}^{T\times\frac{H}{s}\times\frac{W}{s}}$ is read from it by trilinear interpolation, so the correlation cost grows only linearly with V.

Incorporation into a Transformer. Similarly to MVSA [21], a convolutional network Φ of residual blocks downsamples $\mathbf{C}_i$, with the T hypotheses as channels, from $\frac{H}{4} \times \frac{W}{4}$ to the token resolution $\frac{H}{k} \times \frac{W}{k}$, fusing the image features of the two shallower encoder layers at $\frac{1}{4}$ and $\frac{1}{8}$ resolution and of the two deeper ones at $\frac{1}{16}$:

$$\mathrm{t}_i^I = \Phi\left(\mathbf{C}_i, \mathrm{t}_i^{I,(1)}, \ldots, \mathrm{t}_i^{I,(4)}\right).$$

The resulting tokens are fed into the frame and global attention layers and eventually decoded into depth maps and point maps.

3.4 Training

Training Loss. Our loss includes a depth regression loss term, a point map regression term, and a normal loss term:

$$\mathcal{L} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} + \mathcal{L}_{\text{normal}}. \tag{5}$$

For the depth regression loss, we need to normalize the predicted depths to be consistent with the construction of cost volumes. Specifically, depth is estimated by applying a sigmoid function σ to the logits x_{ij} returned by prediction heads, which is then scaled by the depth range defined by the cost-volume depth-sampling bounds:

$$\hat{D}_{i,j} = \exp\left(\log(d_1) + \log(d_T/d_1) \cdot \sigma(x_{ij})\right). \tag{6}$$

Rather than treating every pixel as equally reliable, we let the model down-weight regions it is uncertain about (e.g., due to occlusions or textureless surfaces) using the predicted uncertainty map $\Sigma_{i,j} = \Sigma_i(\mathbf{y}_j)$, following the aleatoric uncertainty formulation of [59, 61]. The depth regression loss is then formulated as

$$\mathcal{L}_{\text{depth}} = \frac{1}{HW} \sum_{i,j} \Sigma_{i,j} \left|\log \hat{D}_{i,j} - \log D_{i,j}^{\text{gt}}\right| \; - \; \alpha \log \Sigma_{i,j}. \tag{7}$$

The point map loss is defined following VGGT [59]: we first represent all quantities in the coordinate system of the first camera, then calculate the mean Euclidean distance s from the 3D points in the point map P^{gt} to the origin, and use this value as a normalization scale when predicting point map:

$$\mathcal{L}_{\text{pmap}} = \frac{1}{HW} \sum_{i,j} \Sigma_{i,j} \left\|\hat{P}_{i,j} - \frac{1}{\mathsf{s}} P_{i,j}^{\text{gt}}\right\| \; - \; \alpha \log \Sigma_{i,j}. \tag{8}$$

We share the same uncertainty map Σ_i between the depth and point map losses, since both quantities are derived from the same underlying geometry, and set $\alpha = 0.2$ in our experiments. Since the normal map $\mathbf{N}$ is computed from the depth map and intrinsics via finite differences, it is ill-defined at image boundaries and at pixels lacking valid ground-truth depth. We therefore restrict the normal loss to a validity mask $\mathcal{M}_i \in \{0,1\}^{H \times W}$ marking pixels with well-defined ground-truth normals:

$$\mathcal{L}_{\text{normal}} = \frac{1}{\sum_{i,j} \mathcal{M}_{i,j}} \sum_{i,j} \mathcal{M}_{i,j} \left(1 - \hat{\mathbf{N}}_{i,j} \cdot \mathbf{N}_{i,j}\right). \tag{9}$$

Training Data. We train on a mixture of 18 public real and synthetic datasets covering indoor, outdoor, object-centric and driving scenes: ScanNet [8], Hypersim [41], TartanAir and its V2 release [62], Virtual KITTI 2 [5], BlendedMVG [67], MVS-Synth [20], Taskonomy [69], MegaDepth [27], CO3D [40], Replica [49], Aria Synthetic Environments [1], CA-1M [25], renderings of TRELLIS 3D assets [64], UnrealStereo4K [52], Spring [33], Mapillary Metropolis [32] and Kubric-4D [16, 55]. We do not use all the sequences available in these datasets, but only a subset of them. Training views are selected per dataset following DVLT [4].

Implementation Details. The image encoder is initialized with the pretrained DINOv3 weights, and the matching features are extracted with the first block of ResNet18 [19] at $\frac{H}{4} \times \frac{W}{4}$. The transformer has 8 layers with an embedding dimension of 768. Following DVLT [4], each training sample has 2 to 18 views, 640 pixels wide with an aspect ratio drawn per batch from $[0.33, 1]$, and each batch holds as many samples as fit in $20{,}000$ patch tokens per GPU. We use $T = 128$ depth hypotheses; the depth range is taken from the ground truth and randomly widened during training, and estimated from the camera baselines with a coarse-to-fine sweep at inference. We train with AdamW (learning rate 10^{-4}, 10^{-5} for the image encoder, weight decay 10^{-4}, cosine schedule) for $160k$ iterations in bf16 precision on 4 A100 GPUs over 4 days.

Methods	ETH3D		KITTI		DTU		ScanNet		Tanks&Temples		Runtime (s)
	AUC.05	AUC.10	AUC.05	AUC.10	AUC.05	AUC.10	AUC.05	AUC.10	AUC.05	AUC.10	
VGGT	0.06	0.21	0.01	0.03	0.68	0.84	0.34	0.59	0.02	0.08	0.499
Fast3R	0.02	0.10	0.00	0.02	0.16	0.31	0.15	0.38	0.01	0.06	0.153
MapAnything	0.04	0.18	0.00	0.01	0.07	0.19	0.15	0.40	0.02	0.10	1.635
Pi3	0.20	0.47	0.03	0.13	0.62	0.79	0.50	0.72	0.03	0.13	0.235
DepthAnything3	0.17	0.44	0.03	0.13	0.70	0.84	0.42	0.63	0.03	0.13	0.492
DVLT	0.06	0.23	0.01	0.05	0.41	0.66	0.47	0.69	0.02	0.09	0.277
VGGT-Ω	0.21	0.52	0.09	0.33	0.70	0.84	0.46	0.68	0.03	0.12	0.296
MapAnything-w/c	0.02	0.07	0.00	0.00	0.19	0.41	0.20	0.44	0.01	0.06	1.425
Pi3x-w/c	0.21	0.51	0.03	0.13	0.60	0.78	0.52	0.74	0.03	0.14	0.276
DepthAnything3-w/c	0.24	0.55	0.08	0.31	0.72	0.85	0.48	0.66	0.03	0.15	0.521
MVSFormer++	0.30	0.46	0.44	0.62	0.80	0.82	0.16	0.25	0.23	0.37	2.236
MVSA	0.33	0.50	0.42	0.62	0.81	0.84	0.35	0.53	0.25	0.43	2.155
S2S-MVS	0.37	0.58	0.47	0.67	0.87	0.92	0.42	0.62	0.28	0.44	0.339

Table 1: Comparisons with state-of-the-art sequence-to-sequence feed-forward models on reconstruction accuracy using the RMVDB benchmark. The -w/c suffix indicates that the model uses ground truth camera parameters as input; MVSA, MVSFormer++ and S2S-MVS are MVS methods that always require ground truth cameras. The best, second-best and third-best results in each column are highlighted.

4 Experiments

In this section, we evaluate the performances of our model and the state-of-the-art competitors including both sequence-to-sequence and sequence-to-one methods on 3D reconstruction tasks. Following this, we will also present an ablation study to validate the effectiveness of the proposed ray-map embedding and global-cost-volume designs.

Baselines. Our main competitors are the feed-forward sequence-to-sequence methods, including VGGT [59], Fast3R [65], MapAnything [23], Pi3 [63], DepthAnything3 [28], VGGT-Ω [60] and DVLT [4]. Among these methods, VGGT, Fast3R, VGGT-Ω and DVLT do not take camera parameters as input, we will use the estimated camera parameters to produce point maps in our evaluation. MapAnything, Pi3 and DepthAnything3 support camera parameters as input, we will additionally evaluate these models with ground truth camera parameters as input, to form direct and fair comparison against our method. All methods take the first-camera frame as the world frame. We also compare against MVSFormer++ [6] and MVSA [21], which are the state-of-the-art sequence-to-one models to the best of our knowledge.

Benchmark. We use the RMVDB benchmark [46] to evaluate all methods, which consists of five multi-view datasets, i.e., KITTI [15], ScanNet [8], ETH3D [45], DTU [22] and Tanks & Temples [24]. Except for ScanNet, whose training split is part of our training data, none of these datasets is used for training. These datasets cover a diverse set of real-world scenarios, including driving sequences, room scans, building scans, and tabletop objects. We use all 8-frame tuples provided in the RMVDB for evaluation.

Metrics. We use relative per-point errors between predicted and ground truth point maps to compute our main evaluation metric. To account for the different scales in different scenes, we first compute the scale of each predicted or ground truth point maps as the mean distance to the origin, and normalize the point maps accordingly before evaluation. Then the relative per-point error is computed as the point errors divided by the ground-truth depth of that point. We report the robust Area Under the Curve (AUC) statistics of the computed relative per-point errors at finer 0.05 and coarser 0.10.

4.1 Comparison with State-of-the-Art Methods

Table 1 compares our S2S-MVS with recent sequence-to-sequence feed-forward reconstruction models and sequence-to-one MVS models on the RMVDB benchmark. Overall, S2S-MVS achieves the best reconstruction accuracy on all datasets except ScanNet, obtaining the highest AUC.05/AUC.10

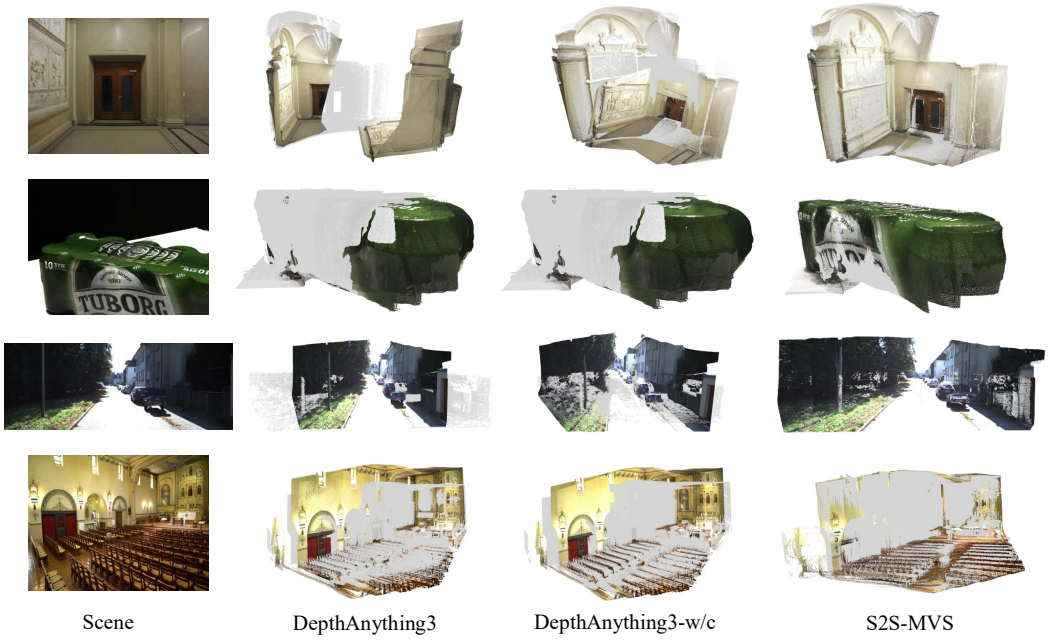


Figure 3: Qualitative point map prediction results of our proposed method compared to DA3. We overlay the predicted point maps and gray-coded ground truth maps to highlight the accuracy of alignment.

scores on ETH3D, KITTI, DTU and Tanks&Temples and showing strong generalization to both indoor structured scenes and outdoor driving scenes.

In addition to camera-free feed-forward methods, S2S-MVS also compares favorably to feed-forward and MVS methods that take camera parameters as input, demonstrating the effectiveness of our sequence-to-sequence MVS formulation. On ScanNet, which is part of the training data of most feed-forward models as well as ours, several feed-forward models (e.g., Pi3, VGGT-Ω and DVLT) obtain higher accuracy, while S2S-MVS still clearly outperforms the other MVS baselines (MVSFormer++ and MVSA).

We also compare runtime efficiency in Table 1. S2S-MVS runs in the same range as the sequence-to-sequence feed-forward models and substantially faster than heavier feed-forward pipelines such as MapAnything. More importantly, compared with the sequence-to-one MVS baselines that achieve the closest reconstruction accuracy to ours, S2S-MVS is considerably faster than both MVSFormer++ and MVSA, while exceeding their accuracy on all five datasets. This shows that our sequence-to-sequence formulation combines the accuracy of dedicated MVS pipelines with the efficiency of feed-forward reconstruction models.

We also provide qualitative comparisons in Fig. 3. In the figure, we compare with DepthAnything3 in both w/ and w/o camera settings, and visualize the predicted point maps overlaid with the gray-coded ground truth point maps. We can see that S2S-MVS produces more accurate reconstructions compared to other methods. The performance is more pronounced on the last scene in the figure from Tanks&Temples dataset, where the reconstructed geometries of other methods suffer from 3D anisotropic scale distortion, leading to an incorrect aspect ratio of the reconstructed scene. In contrast, our method does not suffer from this issue.

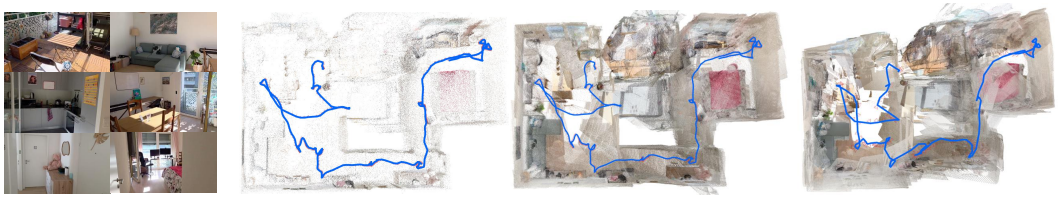
4.2 Further Analysis

Global Alignment. The benefit of the sequence-to-sequence design of S2S-MVS is that it can achieve better global alignment of the reconstructed geometry across views, as the model can directly learn to reason about the global geometry from the input image sequences. In contrast, sequence-to-one methods like MVSA reconstruct each view independently, which can lead to misalignment issues across views. We provide a qualitative comparison in Fig. 4, where we visualize the predicted



Scene MVSA S2S-MVS

Figure 4: Comparison of point map prediction results between our method and MVSA. The proposed sequence-to-sequence design of S2S-MVS achieves better global alignment of the reconstructed geometry.



Scene VidMap VidMap + S2S-MVS VidMap + DepthAnything3

Figure 5: Dense video-based reconstruction results. We use camera parameters estimated by VidMap as input to our S2S-MVS model and DepthAnything3 to predict point maps. Our method produces more accurate reconstructions compared to DepthAnything3, which does not benefit from known camera parameters.

point maps from S2S-MVS and MVSA, where our method produces more globally consistent reconstructions across views.

Combined with SfM. Our model can be easily combined with Structure-from-Motion (SfM) techniques to accomplish dense reconstruction. In this experiment, we use VidMap [37] as the SfM method to estimate camera parameters from input image sequences extracted from a video. The camera parameters are provided to our S2S-MVS model and also DepthAnything3 to predict point maps. We visualize the dense point cloud reconstruction results in Fig. 5. We can see that our method produces more accurate dense reconstructions compared to DepthAnything3, which creates misaligned point clouds and camera offsets.

4.3 Ablation Study

We also conduct an ablation study to analyze the impact of different components in our model. The results are summarized in Tab. 2. We evaluate the variants of our model without the ray-map embedding and without the global cost volume, respectively. We can see that both components contribute significantly to the overall performance of our model. Remove ray map embedding leads to degradation in global alignment, and affect the performance. When training without the global cost volume, the performance drops more significantly, which shows the importance of the global cost volume design in providing strong regularization for the model.

Methods	ETH3D		KITTI		DTU		ScanNet		Tanks&Temples	
	AUC@.05	AUC@.10	AUC@.05	AUC@.10	AUC@.05	AUC@.10	AUC@.05	AUC@.10	AUC@.05	AUC@.10
w/o Ray Map Embedding	0.35	0.55	0.44	0.62	0.80	0.88	0.35	0.50	0.24	0.40
w/o Global Cost Volume	0.25	0.46	0.33	0.53	0.61	0.72	0.30	0.47	0.20	0.38
Full Model	0.37	0.58	0.47	0.67	0.87	0.92	0.42	0.62	0.28	0.44

Table 2: Ablation study on the effectiveness of the proposed ray map embedding and global cost volume designs.

5 Discussion and Conclusion

In this paper, we propose S2S-MVS, a novel sequence-to-sequence feed-forward model for multi-view stereo reconstruction. Our model is designed to directly predict depth maps for each view from input image sequences, from which point maps are obtained with the ray maps, assuming camera parameters to be known. We introduce a ray-map embedding design to encode the camera information of the input views, and a global cost volume design to provide strong regularization for geometry prediction. We evaluate our model on the RMVDB benchmark, and show that it achieves superior reconstruction accuracy compared to state-of-the-art sequence-to-sequence feed-forward models. However, compared to other feed-forward models, our model is still limited by the scale of training data and thus does not generalize well in certain scenarios. Nonetheless, we believe that our S2S-MVS model provides a promising direction for efficient and accurate multi-view stereo reconstruction, and can be further improved by exploring more advanced architectures and training strategies.

Acknowledgments and Disclosure of Funding

This work was funded in part by the Swiss National Science Foundation.

References

[1] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *European conference on computer vision*, pages 247–263. Springer, 2024.

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5460–5469, 2022.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[4] Alessandro Burzio, Tobias Fischer, Sven Elflein, Qunjie Zhou, Riccardo de Lutio, Jiawei Ren, Jiahui Huang, Shengyu Huang, Marc Pollefeys, Laura Leal-Taixé, Zan Gojcic, and Haithem Turki. Déjà View: Looping transformers for multi-view 3D reconstruction. *arXiv preprint arXiv:2605.30215*, 2026.

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.

[6] Chenjie Cao, Xinlin Ren, and Yanwei Fu. MVSFormer++: Revealing the devil in transformer's details for multi-view stereo. *arXiv preprint arXiv:2401.11673*, 2024.

[7] OpenMVS Contributors. OpenMVS. `https://github.com/cdcseacave/openMVS`, 2026. Accessed: 2026-05-06.

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[10] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. TransMVSNet: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8585–8594, 2022.

[11] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17765–17775, 2023.

[12] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19790–19800, 2024.

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[14] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2015.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3354–3361, 2012.

[16] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022.

[17] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020.

[18] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[21] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisin Mac Aodha, Gabriel Brostow, and Jamie Watson. MVSAnywhere: Zero-shot multi-view stereo. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11493–11504, 2025.

[22] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

[23] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. MapAnything: Universal feed-forward metric 3D reconstruction. In *2026 International Conference on 3D Vision (3DV)*. IEEE, 2026.

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[25] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify Anything: Scaling indoor 3D object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22225–22233, 2025.

[26] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[27] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.

[28] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[29] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023.

[30] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE international conference on computer vision*, volume 2, pages 1150–1157. IEEE, 1999.

[31] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview stereo with cascaded epipolar RAFT. In *European conference on computer vision*, pages 734–750. Springer, 2022.

[32] Mapillary. Mapillary Metropolis. `https://www.mapillary.com/dataset/metropolis`, 2021.

[33] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4981–4991, 2023.

[34] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[35] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.

[36] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European conference on computer vision*, pages 58–77. Springer, 2024.

[37] Zador Pataki, Paul-Edouard Sarlin, and Marc Pollefeys. VidMap: Exploiting temporal structure for video-based structure-from-motion. *arXiv preprint arXiv:2607.27194*, 2026.

[38] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8645–8654, 2022.

[39] Pix4D. Pix4D. `https://www.pix4d.com/`, 2026. Accessed: 2026-05-06.

[40] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10881–10891, 2021.

[41] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[42] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE international conference on computer vision*, pages 2564–2571. IEEE, 2011.

[43] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[45] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017.

[46] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. A benchmark and a baseline for robust multi-view depth estimation. In *2022 International Conference on 3D Vision (3DV)*, pages 637–645. IEEE, 2022.

[47] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

[48] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[49] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[51] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[52] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-Nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8938–8948, 2021.

[53] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6268, 2020.

[54] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. *Advances in neural information processing systems*, 33:14254–14265, 2020.

[55] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European conference on computer vision*, pages 313–331. Springer, 2024.

[56] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster VGGT with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025.

[57] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchmatchNet: Learned multi-view PatchMatch stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14203, 2021.

[58] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024.

[59] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[60] Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2026.

[61] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[62] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[63] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[64] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D latents for scalable and versatile 3D generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025.

[65] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[66] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European conference on computer vision*, pages 767–783. Springer, 2018.

[67] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[68] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018.

[69] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the proposed sequence-to-sequence MVS formulation and key components, and these are supported by the method description (Sec. 3) and experiments (Sec. 4).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 5 of the paper, we discuss the limitations of our work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is primarily methodological and empirical and does not present formal theoretical results requiring proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the model design and training setup (Sec. 3) and report the evaluation protocol, benchmarks, and metrics used for comparisons (Sec. 4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code, training and evaluation instructions are publicly released at `https://github.com/AoxiangFan/S2S_MVS_release`. All training and evaluation datasets are public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The training configuration and key implementation choices are summarized in Sec. 3, and the test benchmark setup and metrics are described in Sec. 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Evaluation metrics are reported, but no error bars or statistical significance tests are provided following previous work in the field.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the main training compute and runtime-related details in Sec. 3 (Implementation Details) and include per-method runtime in the experiments section (Sec. 4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and believe the work conforms.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is methodological and evaluated on standard reconstruction benchmarks. We do not identify specific direct societal impacts in the scope of this submission.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The submission does not involve releasing high-risk models or scraped datasets. Experiments use established public benchmarks and standard baselines.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use widely adopted public datasets and prior methods and cite the corresponding sources throughout. Assets are used in accordance with their stated usage terms to the best of our knowledge.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer:[N/A]

    Justification: No new datasets or other assets are released as part of this submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subject research is conducted, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important or non-standard component of the proposed method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.