CostFormer: Cost Transformer for Cost Aggregation in Multi-view Stereo

Weitao Chen[1] , **Hongbin Xu**[1,2] , **Zhipeng Zhou**[1] , **Yang Liu**[1] , **Baigui Sun**[†1] , **Wenxiong Kang**[†2,3] and **Xuansong Xie**[1]

[1]Alibaba Group

[2]South China University of Technology

[3]Guangdong Artificial Intelligence and Digital Enconomy Laboratory, Pazhou Lab

{hillskyxm, hongbinxu1013}@gmail.com, auwxkang@scut.edu.cn

Abstract

The core of Multi-view Stereo(MVS) is the matching process among reference and source pixels. Cost aggregation plays a significant role in this process, while previous methods focus on handling it via CNNs. This may inherit the natural limitation of CNNs that fail to discriminate repetitive or incorrect matches due to limited local receptive fields. To handle the issue, we aim to involve Transformer into cost aggregation. However, another problem may occur due to the quadratically growing computational complexity caused by Transformer, resulting in memory overflow and inference latency. In this paper, we overcome these limits with an efficient Transformer-based cost aggregation network, namely CostFormer. The Residual Depth-Aware Cost Transformer(RDACT) is proposed to aggregate long-range features on cost volume via self-attention mechanisms along the depth and spatial dimensions. Furthermore, Residual Regression Transformer(RRT) is proposed to enhance spatial attention. The proposed method is a universal plug-in to improve learning-based MVS methods.

1 Introduction

Given a series of calibrated images from different views in one scene, Multi-view Stereo (MVS) aims to recover the 3D information of the observed scene. It is a fundamental problem in computer vision and widely applied to robot navigation, autonomous driving, augmented reality, and etc. Recent learning-based MVS networks [Yao *et al.*, 2018; Gu *et al.*, 2020; Wang *et al.*, 2021] have achieved inspiring success both in the quality and the efficiency of 3D reconstruction. Generally, deep MVS approaches consist of the following five steps: feature extraction from multi-view images via CNN network with shared weights, differentiable warping to align all source features to the reference view, matching cost computation from reference features and aligned source features, matching cost aggregation or regularization, depth or disparity regression.
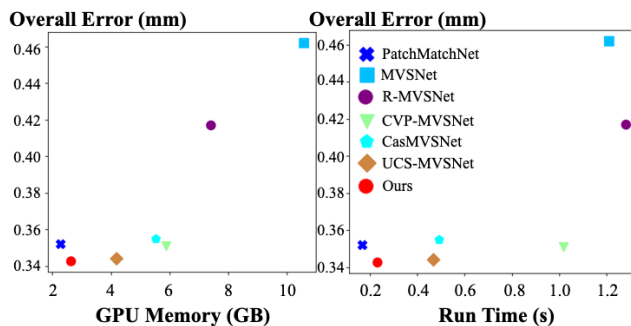

Figure 1: Comparison with state-of-the-art MVS methods on DTU. Relationship between error, GPU memory and run-time with image size 1152×864.

Current progresses in learning-based MVS primarily concentrate on the limitation of reconstruction quality [Wei *et al.*, 2021; Yang *et al.*, 2020a], memory consumption [Yan *et al.*, 2020; Wei *et al.*, 2021], and efficiency [Wang *et al.*, 2021; Wang *et al.*, 2022a]. The basic network architecture of these works is based on the pioneering backbone network called MVSNet [Yao *et al.*, 2018], which provides an elegant and stable baseline. However, instead of taking the inheritance of network design principle in MVSNet [Yao *et al.*, 2018] for granted, we can rethink the task of MVS problem as a dense correspondence problem [Hosni *et al.*, 2012] alternatively. The core of MVS is a dense pixelwise correspondence estimation problem that searches the corresponding pixel of a specific pixel in the reference image along the epipolar line in all warped source images. No matter which task this correspondence estimation problem is applied to, the matching task can be boiled down to a classical matching pipeline [Scharstein and Szeliski, 2002]: (1) feature extraction, and (2) cost aggregation. In learning-based MVS methods, the transition from traditional hand-crafted features to CNN-based features inherently solves the former step of the classical matching pipeline via providing powerful feature representation learned from large-scale data. However, *handling the cost aggregation step by matching similarities between features without any prior usually suffers from the challenges due to ambiguities generated by repetitive patterns or background clutters* [Cho *et al.*, 2021]. Consequently, a typical solution in MVSNet and its variants [Yao *et al.*, 2018; Gu *et al.*, 2020;

---

†Corresponding authors.

‡Appendix is presented in: https://arxiv.org/abs/2305.10320.

Wang *et al.*, 2021] is to apply a 3D CNN or an RNN to regularize the cost volume among reference and source views, rather than directly rely on the quality of the initial correlation clues in cost volume. Although formulated variously in previous methods, these methods either use hand-crafted techniques that are agnostic to severe deformations or inherit the limitation of CNNs, e.g. limited receptive fields, unable to discriminate incorrect matches that are locally consistent.

In this work, we focus on the *cost aggregation step* of cost volume and propose a novel cost aggregation Transformer (**CostFormer**) to tackle the issues above. Our CostFormer is based on Transformer [Vaswani *et al.*, 2017], which is renowned for its global receptive field and long-range dependent representation. By aggregating the matching cost in the cost volume, our aggregation network can explore global correspondences and refine the ambiguous matching points effectively with the help of the self-attention (SA) mechanism in Transformer. Though the promising performances of Vision Transformers have been proven in many applications [Dosovitskiy *et al.*, 2020; Sun *et al.*, 2021], the time and memory complexity of the key-query dot product interaction in conventional SA grow quadratically with the spatial resolution of inputs. Hence, replacing 3D CNN with Transformer may result in unexpected extra occupancy in memory and latency in inference. Inspired by [Wang *et al.*, 2021], we further introduce the Transformer architecture into an iterative multi-scale learnable PatchMatch pipeline. It inherits the advantages of the long-range receptive field in Transformers, improving the reconstruction performance substantially. Meantime, it also maintains a balanced trade-off between efficiency and performance, which is competitive in the inference speed and parameters magnitude compared with other methods.

Our main contributions are as follows:

(1) In this paper, we propose a novel Transformer-based cost aggregation network called CostFormer, which can be plugged into learning-based MVS methods to improve cost volume effectively. (2) CostFormer applies an efficient Residual Depth-Aware Cost Transformer to cost volume, extending 2D spatial attention to 3D depth and spatial attention. (3) CostFormer applies an efficient Residual Regression Transformer between cost aggregation and depth regression, keeping spatial attention. (4) The proposed CostFormer brings benefits to learning-based MVS methods when evaluating DTU [Aanæs *et al.*, 2016], Tanks&Temples [Knapitsch *et al.*, 2017] ETH3D [Schöps *et al.*, 2017], BlendedMVS [Yao *et al.*, 2020] and YFCC[Thomee *et al.*, 2016] datasets.

2 Related Work

2.1 Learning-based MVS Methods

Powered by the great success of deep learning-based techniques, many learning-based methods have been proposed to boost the performance of Multi-view Stereo. MVSNet [Yao *et al.*, 2018] is a landmark for the end-to-end network that infers the depth map on each reference view for the MVS task. Feature maps extracted by a 2D CNN on each view are reprojected to the same reference view to build a variance-based cost volume. A 3D CNN is further used to regress the depth

map. Following this pioneering work, lots of efforts have been devoted to boosting speed and reducing memory occupation. To relieve the burden of huge memory cost, recurrent neural networks are utilized to regularize the cost volume in AA-RMVSNet [Wei *et al.*, 2021]. Following a coarse-to-fine manner to develop a computationally efficient network, a recent strand of works divide the single cost volume into several cost volumes at multiple stages, like CasMVSNet [Gu *et al.*, 2020], CVP-MVSNet [Yang *et al.*, 2020a], UCSNet [Cheng *et al.*, 2020], and etc. Inspired by the traditional PatchMatch stereo algorithm, PatchMatchNet [Wang *et al.*, 2021] inherits the pipeline in PatchMatch stereo in an iterative manner and extend it into a learning-based end-to-end network.

2.2 Vision Transformer

The success of Transformer [Vaswani *et al.*, 2017] and its variants [Dosovitskiy *et al.*, 2020; Liu *et al.*, 2021] have motivated the development of Neural Language Processing in recent years. Borrowing inspiration from these works, Transformer has been successfully extended to vision tasks and proven to boost the performance of image classification [Dosovitskiy *et al.*, 2020]. Following the pioneering work, many efforts are devoted to boosting the development of various vision tasks with the powerful representation ability of Transformer.

In [Li *et al.*, 2021], the application of Transformer in the classic stereo disparity estimation task is investigated thoughtfully. Swin Transformer [Liu *et al.*, 2021] involves the hierarchical structure into Vision Transformers and computes the representation with shifted windows. Considering Transformer's superiority in extracting global content information via attention mechanism, many works attempt to utilize it in the task of feature matching. Given a pair of images, CATs [Cho *et al.*, 2021] explore global consensus among correlation maps extracted from a Transformer, which can fully leverage the self-attention mechanism and model long-range dependencies among pixels. LoFTR [Sun *et al.*, 2021] also leverages Transformers with a coarse-to-fine manner to model dense correspondence. STTR [Li *et al.*, 2021] extends the feature matching Transformer architecture to the task of stereo depth estimation task in a sequence-to-sequence matching perspective. TransMVSNet [Ding *et al.*, 2021] is the most relevant concurrent work compared with ours, which utilizes a Feature Matching Transformer (FMT) to leverage self-attention and cross-attention to aggregate long-range context information within and across images. Specifically, the focus of TransMVSNet is on the enhancement of feature extraction before cost aggregation, while our proposed CostFormer aims to improve the cost aggregation process on cost volume.

3 Methodology

In this section, we introduce the detailed architecture of the proposed CostFormer which focuses on the cost aggregation step of cost volume. CostFormer contains two specially designed modules called Residual-Depth Aware Cost Transformer (RDACT) and Residual Regression Transformer (RRT), which are utilized to explore the relation between pixels within a long range and the relation between different
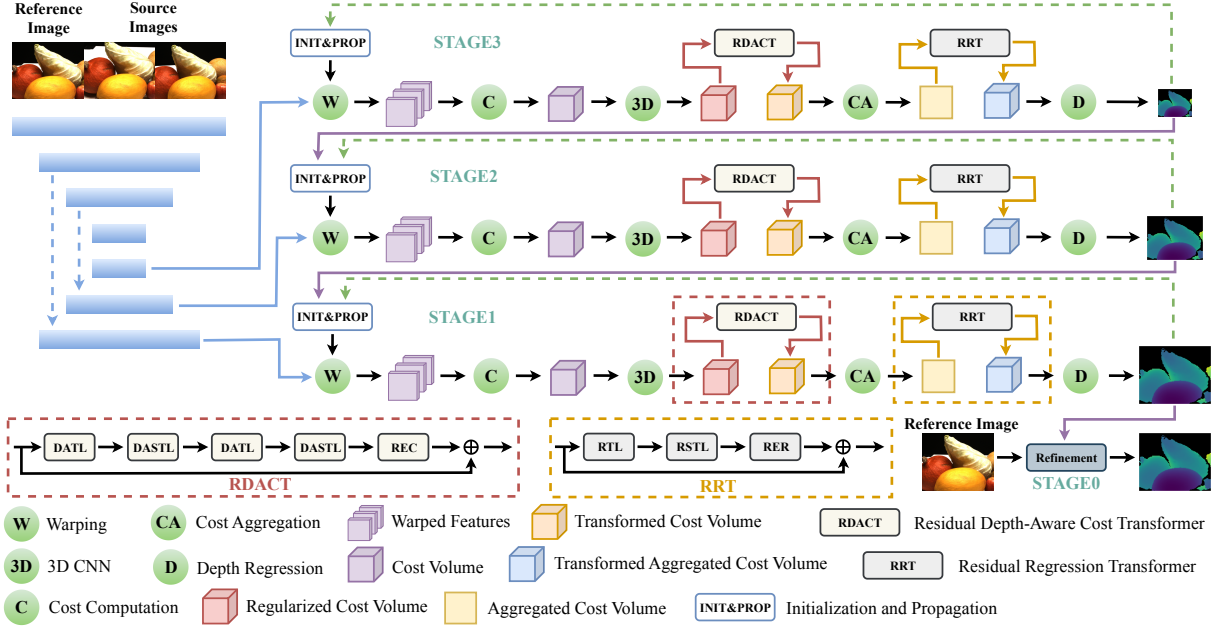
Figure 2: Structure of CostFormer based on PatchMatchNet.

depth hypotheses during the evaluation process. In Section Preliminary, we give a brief preliminary on the pipeline of our method. Then we show the construction of RDACT and RRT respectively. Finally, we show experiments.

3.1 Preliminary

In general, the proposed RDACT and RRT can be integrated with arbitrary cost volume of learning-based MVS networks. Based on the patch match architecture [Wang *et al.*, 2021], we further explore the issue of cost aggregation on cost volume. As shown in Figure 2, CostFormer based on PatchMatchNet [Wang *et al.*, 2021] extracts feature maps from multi-view images and performs initialization and propagation to warp the features maps in source views to reference view. Given a pixel p at the reference view and its corresponding pixel $p_{i,j}$ at the i-th source view under the j-th depth hypothesis d_j is defined as:

$$p_{i,j} = K_i \cdot (R_{0,i} \cdot (K_0^{-1} \cdot p \cdot d_j) + t_{0,i}) \quad (1)$$

where $R_{0,i}$ and $t_{0,i}$ denote the rotation and translation between the reference view and i-th source view. K_0 and K_i are the intrinsic matrices of the reference and i-th source view. The warped feature maps at the i-th source view $F_i(p_{i,j})$ are bilinearly interpolated to remain the original resolution. Then, a cost volume is constructed from the similarity of feature maps, and 3D CNNs are applied to regularize the cost volume. Warped features from all source views are integrated into a single cost for each pixel p and depth hypothesis d_j by computing the cost per hypothesis $S_i(p,j)^g$ via group-wise correction as follows:

$$S_i(p,j)^g = \frac{G}{C} < F_0(p)^g, F_i(p_{i,j})^g > \in \mathbb{R}^G \quad (2)$$

where G is the group number, C is the channel number, $< \cdot, \cdot >$ is the inner product, $F_0(p)^g$ and $F_i(p_{i,j})^g$ are grouped

reference feature map and grouped source feature map at the i-th view respectively. Then they aggregate over the views with a pixel-wise view weight $w_i(p)$ to get $\overline{S}(p,j)$.

Taking no account of Transformer at the cost aggregation (CA) step, a CA module firstly utilizes a small network with 3D convolution with $1\times1\times1$ kernels to obtain a single cost, $\mathcal{C} \in R^{H\times W\times D}$. For a spatial window of K_e pixels $\{p_k\}_{k=1}^{K_e}$ can be organized as a grid, per pixel additional offsets $\{\Delta p_k\}_{k=1}^{K_e}$ can be learned for spatial adaptation. The aggregated spatial cost $\widetilde{\mathcal{C}}(p,j)$ is defined as:

$$\widetilde{\mathcal{C}}(p,j) = \frac{1}{\sum_{k=1}^{K_e} w_k d_k} \sum_{k=1}^{K_e} w_k d_k \mathcal{C}(p + p_k + \Delta p_k, j) \quad (3)$$

where w_k and d_k weight the cost $\mathcal{C}$ based on feature and depth similarity. Given the sampling positions $(p + p_k + \Delta p_k)_{k=1}^{K_e}$, corresponding features from F_0 are extracted via bilinear interpolation. Then group-wise correlation is applied between the features at each sampling location and p. The results are concatenated into a volume on which 3D convolution layers with 1×1×1 kernels and sigmoid non-linearities are applied to output normalized weights $\{w_k\}_{k=1}^{K_e}$. The absolute difference in inverse depth between each sampling point and pixel p with their j-th hypotheses are collected. Then a sigmoid function on the inverted differences is applied to obtain $\{d_k\}_{k=1}^{K_e}$.

The remarkable thing is that such cost aggregation inevitably suffers from challenges due to ambiguities generated by repetitive patterns or background clutters. The local mechanisms in ambiguities exist in many operations, such as local propagation and spatial adaptation by small learnable slight offset. CostFormer significantly alleviates these problems through RDACT and RRT. The original CA module is also repositioned between RDACT and RRT.
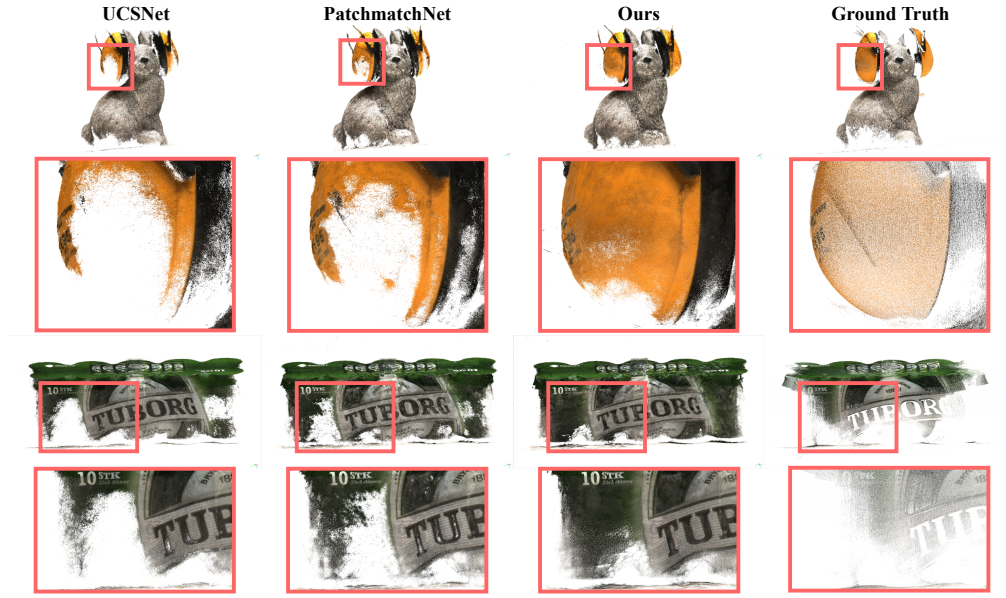
Figure 3: Comparison of different methods on the DTU evaluation set. The backbone of CostFormer is PatchMatchNet here.

After RRT, soft argmin is applied to get the regressed depth. Finally, a depth refinement module is designed to refine the depth regression.

For CascadeMVS and other cascade architectures, Cost-Former can be plugged into similarly.

3.2 Residual Depth-Aware Cost Transformer

In this section, we explore the details of the Residual Depth-Aware Cost Transformer (RDACT). Each RDACT consists of two parts. The first part is a stack of Depth-Aware Transformer layer (DATL) and Depth-Aware Shifted Transformer layer (DASTL), which deal with the cost volumes to explore the relations sufficiently. The second part is the Re-Embedding Cost layer (REC) which recovers the cost volume from the first part.

Given a cost volume $\mathcal{C}_0 \in \mathbb{R}^{H \times W \times D \times G}$, temporary intermediate cost volumes $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_L \in \mathbb{R}^{H \times W \times D \times E}$ are firstly extracted by DATL and DASTL alternatively:

$$\mathcal{C}_k = \text{DASTL}_k(\text{DATL}_k(\mathcal{C}_{k-1})), k = 1, 2, ..., L \quad (4)$$

where DATL_k is the k-th Depth-Aware Transformer layer with regular windows, DASTL_k is the k-th Depth-Aware Transformer layer with shifted windows, E is the embedding dimension number of DATL_k and DASTL_k.

Then a Re-Embedding Cost layer is applied to the last $\mathcal{C}_k$, namely $\mathcal{C}_L$, to recover G from E. The output of RDACT is formulated as:

$$\mathcal{C}_{out} = \text{REC}(\mathcal{C}_L) + \mathcal{C}_0 \quad (5)$$

where REC is the Re-Embedding Cost layer, and it can be a 3D convolution with G output channels. If $E = G$, $\mathcal{C}_{out}$ can be simply formulated as:

$$\mathcal{C}_{out} = \mathcal{C}_L + \mathcal{C}_0 \quad (6)$$

This residual connection allows the aggregation of different levels of cost volumes; $\mathcal{C}_{out}$ instead of $\mathcal{C}_0$ is then aggregated by the original aggregation network described in section 3.1. The whole RDACT is shown in the red window in Figure 2.

Before introducing the construction of DATL and DASTL, we dive into the details of core constitutions called Depth-Aware Multi-Head Self-Attention (DA-MSA) and Depth-Aware Shifted Multi-Head Self-Attention (DAS-MSA). Both DA-MSA and DAS-MSA are based on Depth-Aware Self-Attention Mechanism. In order to explain Depth-Aware Self-Attention Mechanism, we supply the knowledge about Depth-Aware Patch Embedding and Depth-Aware Windows as preliminary.

Depth-Aware Patch Embedding

Obviously, directly applying the attention mechanism for feature maps at pixel-wise level is quite costly in terms of GPU memory usage. In order to tackle this issue, we propose a Depth-Aware Patch Embedding to reduce the high memory cost and get an additional regularization. Specifically, given a grouped cost volume before aggregation $\mathcal{C} \in \mathbb{R}^{H \times W \times D \times G}$, a depth-aware patch embedding is firstly applied to $\mathcal{C}$ to get tokens. It consists of a 3D convolution with kernel size $h \times w \times d$ and a layer normalization. To downsample the spatial sizes of cost volume and keep the depth hypotheses, we set h and w to more than 1 and d as 1. So the sample ratio is adaptive for memory cost and run time. Before convolution, cost volume will be padded to fit the spatial sizes and downsampling ratio. After layer normalization(LN), these embedded patches are further partitioned by depth-aware windows.

Depth-Aware Windows

Beyond the nonlinear and linear global self-attention, local self-attention within a window has been proven to be more effective and efficient. As an example of 2D windows, Swin Transformer [Liu *et al.*, 2021] directly applies multi-head self-attention mechanisms on non-overlapping 2D windows to avoid the big computation complexity of global tokens.

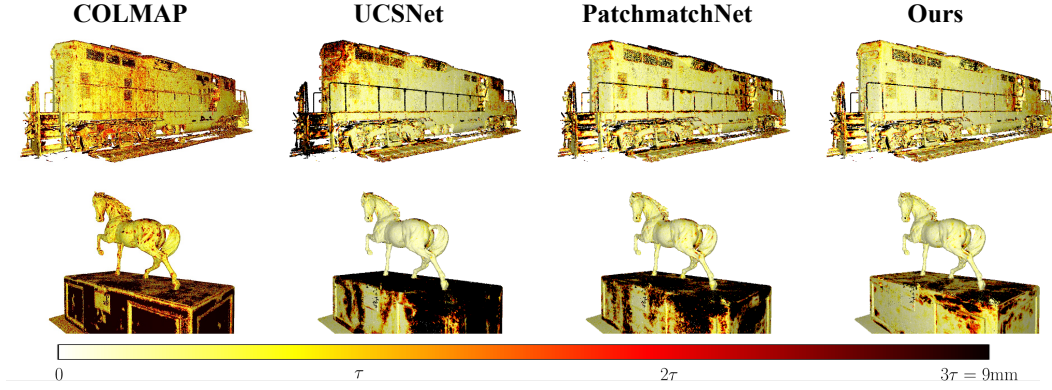| COLMAP	UCSNet	PatchmatchNet	Ours

Figure 4: Comparison of different methods on Tanks&Temples. The Recall reported by official benchmark is presented.

Extended from the 2D spatial window, an embedded cost volume patch $\in \mathbb{R}^{H^* \times W^* \times D^* \times G}$ with depth information is partitioned into non-overlapping 3D windows. These local windows are then transposed and reshaped to local cost tokens. Assuming the sizes of these windows are $hs \times ws \times ds$, the total number of tokens is $\lceil \frac{H^*}{h_s} \rceil \times \lceil \frac{W^*}{w_s} \rceil \times \lceil \frac{D^*}{d_s} \rceil$. These local tokens are further processed by the multi-head self-attention mechanism.

Depth-Aware Self-Attention Mechanism

For a cost window token $X \in \mathbb{R}^{h_s \times w_s \times d_s \times G}$, the query, key, and value matrices Q, K and $V \in \mathbb{R}^{h_s \times w_s \times d_s \times G}$ are computed as:

$$Q = XP_Q, K = XP_K, V = XP_V \qquad (7)$$

where P_Q, P_K, and $P_V \in \mathbb{R}^{G \times G}$ are projection matrices shared across different windows. By introducing depth and spatial aware relative position bias $B1 \in \mathbb{R}^{(h_s \times h_s) \times (w_s \times w_s) \times (d_s \times d_s)}$ for each head, the depth-aware self-attention(DA-SA1) matrix within a 3D local window is thus computed as:

$$\text{DA-SA1} = Attention1(Q1, K1, V1) = SoftMax(\frac{Q1K1^T}{\sqrt{G}} + B1)V1$$
$$\qquad (8)$$

Where $Q1$, $K1$ and $V1 \in \mathbb{R}^{h_s w_s d_s \times G}$ are reshaped from Q, K and $V \in \mathbb{R}^{h_s \times w_s \times d_s \times G}$. The process of DATL with LayerNorm(LN) and multi-head DA-SA1 at the current level is formulated as:

$$\widehat{X}^l = \text{DA-MSA1}((\text{LN}(X^{l-1})) + X^{l-1} \qquad (9)$$

By introducing depth-aware relative position bias $B2 \in \mathbb{R}^{d_s \times d_s}$ for each head, the depth-aware self-attention(DA-SA2) matrix along the depth dimension is an alternative module to DATL and thus computed as:

$$\text{DA-SA2} = Attention2(Q2, K2, V2) = SoftMax(\frac{Q2K2^T}{\sqrt{G}} + B2)V2$$
$$\qquad (10)$$

Where $Q2$, $K2$ and $V2 \in \mathbb{R}^{h_s w_s \times d_s \times G}$ are reshaped from Q, K and $V \in \mathbb{R}^{h_s \times w_s \times d_s \times G}$. $B1$ and $B2$ will be along the depth dimension and lie in the range of $[-d_s + 1, d_s - 1]$. Along the height and width dimension, $B1$ lies in the range

of $[-h_s + 1, h_s - 1]$ and $[-w_s + 1, w_s - 1]$. In practice, we parameterize a smaller-sized bias matrix $\overline{B1} \in \mathbb{R}^{(2h_s-1) \times (2w_s-1) \times (2d_s-1)}$ from $B1$ and perform the attention function for f times in parallel, and then concatenate the depth-aware multi-head self-attention (DA-MSA) outputs. The process of DATL with LayerNorm(LN), multi-head DA-SA1, and DA-SA2 at the current level is formulated as:

$$\widehat{X}^l = \text{DA-MSA1}(\text{LN}(\text{DA-MSA2}(\text{LN}(X^{l-1})))) + X^{l-1}$$
$$\qquad (11)$$

Then, an MLP module that has two fully-connected layers with GELU non-linearity between them is used for further feature transformations:

$$X^l = \text{MLP}(\text{LN}(\widehat{X}^l)) + \widehat{X}^l \qquad (12)$$

Compared with global attention, local attention makes it possible for computation in high resolution.

However, there is no connection across local windows with fixed partitions. Therefore, regular and shifted window partitions are used alternately to enable cross-window connections. So at the next level, the window partition configuration is shifted along the height, width, and depth axes by $\left(\frac{h_s}{2}, \frac{w_s}{2}, \frac{d_s}{2}\right)$. Depth-aware self-attention will be computed in these shifted windows(DAS-MSA); the whole process of DASTL can be formulated as:

$$\widehat{X}^{l+1} = \text{DAS-MSA1}(\text{LN}(\text{DAS-MSA2}(\text{LN}(X^l)))) + X^l$$
$$\qquad (13)$$
$$X^{l+1} = \text{MLP}(\text{LN}(\widehat{X}^{l+1})) + \widehat{X}^{l+1} \qquad (14)$$

DAS-MSA1 and DAS-MSA2 correspond to multi-head Attention1 and Attention2 within a shifted window, respectively. Assuming the number of stages is n, there are n RDACT blocks in CostFormer.

3.3 Residual Regression Transformer

After aggregation, the cost $\widetilde{\mathcal{C}} \in \mathbb{R}^{HXWXD}$ will be used for depth regression. To further explore the spatial relation under some depth, a Transformer block is applied to $\widetilde{\mathcal{C}}$ before softmax. Inspired by the RDACT, the whole process of Residual Regression Transformer(RRT) can be formulated as:

$$\widetilde{\mathcal{C}}_k = \text{RST}_k(\text{RT}_k(\widetilde{\mathcal{C}}_{k-1})), k = 1, 2, ..., L \qquad (15)$$

Methods	Intermediate Group (F-score ↑)							
	Mean	Fam.	Fra.	Hor.	Lig.	M60	Pan.	Pla.	Tra.
MVSNet	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
CasMVSNet	56.84	76.37	58.45	46.26	55.81	56.11	54.06	58.18	49.51
UCS-Net	54.83	76.09	53.16	43.03	54.00	55.60	51.49	57.38	47.89
CVP-MVSNet	54.03	76.50	47.74	36.34	55.12	57.28	54.28	57.43	47.54
PVA-MVSNet	54.46	69.36	46.80	46.01	55.74	57.23	54.75	56.70	49.06
AA-RMVSNet	61.51	77.77	59.53	51.53	64.02	64.05	59.47	60.85	54.90
PatchmatchNet	53.15	66.99	52.64	43.24	54.87	52.87	49.54	54.21	50.81
UniMVSNet	64.36	81.20	**66.34**	53.11	**63.46**	66.09	64.84	**62.23**	57.53
MVSTR	56.93	76.92	59.82	50.16	56.73	56.53	51.22	56.58	47.48
TransMVS	63.52	80.92	65.83	**56.94**	62.54	63.06	60.00	60.20	**58.67**
CostFormer-PM	$56.27^{\uparrow 3.12}$	72.46	52.59	54.27	55.83	56.80	50.88	55.05	52.32
CostFormer-PM*	$57.10^{\uparrow 3.95}$	74.22	56.27	54.41	56.65	54.46	51.45	57.65	51.70
CostFormer-Uni$^-$	$64.40^{\uparrow 0.04}$	**81.45**	66.22	53.88	62.94	66.12	65.35	61.31	57.90
CostFormer-Uni*	$64.51^{\uparrow 0.15}$	81.31	65.51	55.57	**63.46**	**66.24**	**65.39**	61.27	57.30

Table 1: Quantitative results on the Intermediate group of Tanks&Temples benchmark (higher is better). * is pretrained on DTU and fine-tuned on BlendedMVS. - is not pretrained on DTU and trained from scratch on BlendedMVS

Methods	Advanced Group (F-score ↑)					
	Mean	Aud.	Bal.	Cou.	Mus.	Pal.	Tem.
CasMVSNet	31.12	19.81	38.46	29.10	43.87	27.36	28.11
AA-RMVSNet	33.53	20.96	40.15	32.05	46.01	29.28	32.71
PatchmatchNet	32.31	23.69	37.73	30.04	41.80	28.31	32.29
UniMVSNet	38.96	28.33	44.36	39.74	52.89	33.80	34.63
MVSTR	32.85	22.83	39.04	33.87	45.46	27.95	27.97
TransMVS	37.00	24.84	44.59	34.77	46.49	**34.69**	36.62
MVSTER	37.53	26.68	42.14	35.65	49.37	32.16	**39.19**
CostFormer-PM	$34.07^{\uparrow 1.76}$	24.05	39.20	32.17	43.95	28.62	36.46
CostFormer-PM*	$34.31^{\uparrow 2.00}$	26.77	39.13	31.58	44.55	28.79	35.03
CostFormer-Uni$^-$	$\mathbf{39.55}^{\uparrow 0.59}$	28.61	**45.63**	**40.21**	52.81	34.40	35.62
CostFormer-Uni*	$39.43^{\uparrow 0.47}$	**29.18**	45.21	39.88	**53.38**	34.07	34.87

Table 2: Quantitative results on the Advanced group of Tanks&Temples benchmark (higher is better). * is pretrained on DTU and fine-tuned on BlendedMVS. - is not pretrained on DTU and trained from scratch on BlendedMVS

$$\widetilde{\mathcal{C}}_{out} = \text{RER}(\widetilde{\mathcal{C}}_L) + \widetilde{\mathcal{C}}_0 \qquad (16)$$

where RT_k is the k-th Regression Transformer layer with regular windows, RST_k is the k-th Regression Transformer layer with shifted windows, RER is the re-embedding layer to recover the depth dimension from $\widetilde{\mathcal{C}}_L$, and it can be a 2D convolution with D output channels.

RRT also computes self-attention in a local window. Compared with RDACT, RRT focuses more on spatial relations. Compared with regular Swin [Liu *et al.*, 2021] Transformer block, RRT treats the depth as a channel, the number of channels is actually 1 and this channel is squeezed before the Transformer. The embedding parameters are set to fit the cost aggregation of different iterations. If the embedding dimension number equals D, $\widetilde{\mathcal{C}}_{out}$ can be simply formulated as:

$$\widetilde{\mathcal{C}}_{out} = \widetilde{\mathcal{C}}_L + \widetilde{\mathcal{C}}_0 \qquad (17)$$

As a stage may iterate many times with different depth hypotheses, the number of RRT blocks should be set the same as the number of iterations. The whole RRT is shown in the yellow window in Figure 2.

3.4 Training

Loss Function

Final loss combines with the losses of all iterations at all stages and the loss from the final refinement module:

$$Loss = \sum_{k=1}^{s}\sum_{i=1}^{n} L_i^k + L_{ref} \qquad (18)$$

where L_i^k is the regression or unification loss of the i-th iteration at k-th stage. L_{ref} is the regression or unification loss from refinement module. If refinement module does not exist, the L_{ref} loss is set to zero.

Common Training Settings

CostFormer is implemented by Pytorch [Paszke *et al.*, 2019]. For RDACT, we set the depth number at stages 3, 2, 1 as 4, 2, 2; patch size at height, width and depth axes as 4, 4, 1; window size at height, width and depth axes as 7, 7, 2. If the backbone is set as PatchMatchNet, embedding dimension number at stages 3, 2, 1 are set as 8, 8, 4. For RRT, we set the depth number as 2 at all stages, patch size as 1 at all axes; window size as 8 at all axes. If the backbone is set as Patch-MatchNet, embedding dimension number at iteration 2, 2, 1

at stages 3, 2, 1 as 32, 64, 16, 16, 8. All models are trained on Nvidia GTX V100 GPUs. After depth estimation, we reconstruct point clouds similar to MVSNet [Yao *et al.*, 2018].

4 Experiments

4.1 Compared Methods

In the experiments, we compare our proposed method with following state-of-the-art MVS methods, including: 1) Traditional MVS methods, like Furu [Furukawa and Ponce, 2010], Tola [Tola *et al.*, 2012], Gipuma [Galliani *et al.*, 2015], Colmap [Schönberger and Frahm, 2016], MVE [Fuhrmann *et al.*, 2014], PMVS [Furukawa and Ponce, 2010]; 2) Learning-based MVS methods, like SurfaceNet [Ji *et al.*, 2017], MVS-Net [Yao *et al.*, 2018], R-MVSNet [Yao *et al.*, 2019], P-MVSNet [Luo *et al.*, 2019], Point-MVSNet [Chen *et al.*, 2019], Fast-MVSNet [Yu and Gao, 2020], CasMVSNet [Gu *et al.*, 2020], UCS-Net [Cheng *et al.*, 2020], CVP-MVSNet [Yang *et al.*, 2020b], PVA-MVSNet [Yi *et al.*, 2020], Patch-matchNet [Wang *et al.*, 2021], AA-RMVSNet [Wei *et al.*, 2021], UniMVSNet [Peng *et al.*, 2022], PVSNet [Xu and Tao, 2020], IterMVS [Wang *et al.*, 2022a], PatchMatch-RL [Lee *et al.*, 2021], MVSNet-s [Darmon *et al.*, 2021] , VisMVSNet [Zhang *et al.*, 2020], and EPPMVSNet [Ma *et al.*, 2021]. Furthermore, we also conduct the comparison with Transformer based MVS networks: MVSTR [Zhu *et al.*, 2021], Trans-MVSNet [Ding *et al.*, 2022], MVSTER [Wang *et al.*, 2022b], and WT-MVSNet [Liao *et al.*, 2022].

4.2 Main Settings

Our proposed CostFormer can utilize arbitrary MVS network as backbone. We name the framework as CostFormer-PM, CostFormer-Cas, and CostFormer-Uni, when PatchMatch-Net, CasMVSNet, and UniMVSNet are respectively used as backbone.

4.3 Datasets

The datasets used in the evaluation are DTU [Aanæs *et al.*, 2016], BlendedMVS [Yao *et al.*, 2020], ETH3D [Schöps *et al.*, 2017], Tanks&Temples [Knapitsch *et al.*, 2017], and YFCC-100M [Thomee *et al.*, 2016]. The DTU dataset is an indoor multi-view stereo dataset with 124 different scenes, there are 49 views under seven different lighting conditions in one scene. Tanks&Temples is collected in a more complex and realistic environment, and it's divided into the intermediate and advanced set. ETH3D benchmark consists of calibrated high-resolution images of scenes with strong viewpoint variations. It is divided into training and test datasets. While the training dataset contains 13 scenes, the test dataset contains 12 scenes. BlendedMVS dataset is a large-scale synthetic dataset, consisting of 113 indoor and outdoor scenes and split into 106 training scenes and 7 validation scenes.

4.4 Results on Tanks&Temples Benchmark

For the evaluation on Tanks&Temples, we use the DTU dataset and the Blended MVS dataset. We compare our method to those recent learning-based MVS methods, including PatchMatchNet and UniMVSNet which are also set as backbones of CostFormer. The quantitative results on

Methods	Acc.	Comp.	Overall
Furu	0.613	0.941	0.777
Tola	0.342	1.190	0.766
Gipuma	**0.283**	0.873	0.578
Colmap	0.400	0.644	0.532
SurfaceNet	0.450	1.040	0.745
MVSNet	0.396	0.527	0.462
R-MVSNet	0.383	0.452	0.417
P-MVSNet	0.406	0.434	0.420
Point-MVSNet	0.342	0.411	0.376
Fast-MVSNet	0.336	0.403	0.370
CasMVSNet	0.325	0.385	0.355
UCS-Net	0.338	0.349	0.344
CVP-MVSNet	0.296	0.406	0.351
PVA-MVSNet	0.379	0.336	0.357
PatchMatchNet	0.427	0.277	0.352
AA-RMVSNet	0.376	0.339	0.357
UniMVSNet	0.352	0.278	0.315
CostFormer-PM	0.424	**0.262**	0.343
CostFormer-Cas	0.378	0.313	0.345
CostFormer-Uni	0.301	0.322	**0.312**

Table 3: Quantitative results of different methods on DTU.

Method	GPU Mem.	Run-time	Overall
CasMVSNet	262.47%	221.24%	0.355
UCSNet	195.31%	202.84%	0.344
CVP-MVSNet	273.97%	440.53%	0.351
Ours	100.00%	100.00%	0.343

Table 4: Comparison with other SOTA learning-based MVS methods on DTU. Relationship between overall performance, GPU memory and run-time.

the Tanks&Temples set are summarized in Table 1 and 2, which indicate the robustness of CostFormer. Note that the UniMVSNet⁻ in the table only uses BlendedMVS for training which uses less data (no DTU) than the UniMVSNet baseline. The qualitative results are shown in Figure 4.

4.5 Results on DTU Benchmark

Comparisons with Other Methods

For the evaluation on the DTU evaluation set, we only use the DTU training set. During the training phase, we set the image resolution to 640×512. We compare our method to recent learning-based MVS methods, including CasMVSNet and PatchMatchNet which are also set as backbones of Cost-Former. We follow the evaluation metrics provided by the DTU dataset. The quantitative results on the DTU evaluation set are summarized in Table 3, which indicates that the plug-and-play CostFormer improves the cost aggregation. Partial visualization results of Table 3 are shown in Figure 3.

Complexity Analysis

For the complexity analysis of CostFormer, we plug it into PatchMatchNet and first compare the memory consumption and run-time with this backbone. For a fair comparison, a fixed input size of 1152×864 is used to evaluate the computational cost on a single GPU of NVIDIA Telsa V100. Memory consumption and run-time of PatchMatchNet are 2323MB and 0.169s. They are only increased to 2693MB and 0.231s

Method	Trans Improv.(mm)	Time(s)	Time(%)
MVSTR	+0.0140	+0.359s	+78.21%
TransMVSNet	+0.0160	+0.367s	+135.42%
WT-MVSNet	+0.0130	+0.265s	-
MVSTER	+0.0040	+0.016s	+13.34%
Ours	+0.0097	+0.062s	+36.69%

Table 5: Quantitative improvement of performance and incremental cost of run time of different Transformers on DTU evaluation set.

Methods	Training		Testing	
	F1 $\uparrow$	Time(s) $\downarrow$	F1 $\uparrow$	Time(s) $\downarrow$
MVE	20.47	1.32×10^4	30.37	1.06×10^4
Gipuma	36.38	5.87×10^2	45.18	6.90×10^2
PMVS	46.06	8.37×10^2	44.16	9.57×10^2
COLMAP	67.66	2.69×10^3	73.01	1.66×10^3
PVSNet	67.48	-	72.08	8.29×10^2
IterMVS	66.36	-	74.29	-
PatchMatchNet	64.21	4.52×10^2	73.12	4.92×10^2
PatchMatch-RL	67.78	-	72.38	-
CostFormer-PM	**68.92**	5.66×10^2	**75.24**	5.48×10^2

Table 6: Quantitative results of different methods on ETH3D.

Method	EPE	e1(%)	e3(%)
MVSNet	1.49	21.98	8.32
MVSNet-s	1.35	25.91	8.55
CVP-MVSNet	1.90	19.73	10.24
VisMVSNet	1.47	18.47	7.59
CasMVSNet	1.98	15.25	7.60
EPPMVSNet	1.17	12.66	6.20
TransMVSNet	0.73	8.32	3.62
CostFormer-PM	**0.84**	**12.37**	**4.59**
CostFormer-Uni	**0.43**	**7.05**	**2.70**

Table 7: Quantitative results of different methods on BlendedMVS.

Method	EPE	e1(%)	e3(%)
MVSNet	21.56	67.93	49.75
MVSNet-s	20.98	69.57	49.86
CVP-MVSNet	40.07	85.88	76.25
VisMVSNet	19.60	64.98	46.38
CasMVSNet	13.86	63.83	47.27
PatchMatchNet	9.97	62.51	43.62
CostFormer-PM	**7.82**	**54.10**	**34.71**

Table 8: Quantitative results of different methods on the YFCC test set.

by the plug-in.

Based on the reports of PatchMatchNet, we then get the comparison results of other state-of-the-art learning-based methods. Memory consumption and run-time are reduced by 61.9% and 54.8% compared to CasMVSNet, by 48.8% and 50.7% compared to UCSNet and by 63.5% and 77.3% compared to CVP-MVSNet. Combining the results(lower is better) are shown in Table 4 and Figure 1, GPU memory and run-time of CostFormer are set as 100%.

Comparison with Other Transformer-based Methods
We also compare CostFormer with other Transformers-based networks which are used in MVS methods and not plug-and-play. For a fair comparison, only direct improvements(higer is better) and incremental cost of run time(low is better) from pure Transformers under similar depth hypotheses are summarized in Table 5.

4.6 Results on ETH3D Benchmark

We use the PatchMatchNet as backbone and adopt the trained model used in the Tanks&Temples dataset to evaluate the ETH3D dataset. As shown in Table 6, our method outperforms others on both the training and particularly challenging test datasets(higher is better).

4.7 Results on BlendedMVS Benchmark

We use the model used in ETH3D. On BlendedMVS evaluation set, we set $N = 5$ and image resolution as 576×768. End point error (EPE), 1 pixel error (e1), and 3 pexels error (e3) are used as the evaluation metrics. Quantitative results(lower is better) of different methods are shown in Table 7.

4.8 Results on YFCC Dataset

Following the depth evaluation protocal in previous work, we utilize the YFCC-100M [Thomee *et al.*, 2016] to test the per-

formance. YFCC100M is the largest public multimedia collection that has ever been released. The dataset contains a total of 100 million media objects, of which approximately 99.2 million are photos and 0.8 million are videos. We set backbone as PatchMatchNet, $N = 5$, and image resolution as 512x512. End point error (EPE), 1 pixel error (e1), and 3 pixels error (e3) are also used as the evaluation metrics. Quantitative results of different methods are shown in Tabisle 8.

5 Conclusion

In this work, we explore whether cost Transformer can improve the cost aggregation and propose a novel CostFormer with the cascade RDACT and RRT modules. The experimental results on DTU , Tanks&Temples, ETH3D, BlendedMVS, and YFCC show that our method is competitive, efficient, and plug-and-play. Cost Transformer can be your need for better cost aggregation in multi-view stereo.

Acknowledgments

Contribution Statement

Weitao Chen and Hongbin Xu contribute equally to this work.

References

[Aanæs *et al.*, 2016] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *Int. J. Comput. Vis.*, 120(2):153–168, 2016.

[Chen *et al.*, 2019] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, pages 1538–1547. IEEE, 2019.

[Cheng *et al.*, 2020] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *CVPR*, pages 2521–2531. IEEE, 2020.

[Cho *et al.*, 2021] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34, 2021.

[Darmon *et al.*, 2021] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Deep multi-view stereo gone wild. *CoRR*, abs/2104.15119, 2021.

[Ding *et al.*, 2021] 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. *arXiv preprint arXiv:2111.14600*, 2021.

[Ding *et al.*, 2022] 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.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Fuhrmann *et al.*, 2014] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve - a multi-view reconstruction environment. In *GCH*, pages 11–18. Eurographics Association, 2014.

[Furukawa and Ponce, 2010] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010.

[Galliani *et al.*, 2015] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, pages 873–881. IEEE Computer Society, 2015.

[Gu *et al.*, 2020] 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.

[Hosni *et al.*, 2012] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2012.

[Ji *et al.*, 2017] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, pages 2326–2334. IEEE Computer Society, 2017.

[Knapitsch *et al.*, 2017] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017.

[Lee *et al.*, 2021] Jae Yong Lee, Joseph DeGol, Chuhang Zou, and Derek Hoiem. Patchmatch-rl: Deep mvs with pixelwise depth, normal, and visibility. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021.

[Li *et al.*, 2021] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6197–6206, 2021.

[Liao *et al.*, 2022] Jinli Liao, Yikang Ding, Yoli Shavit, Dihe Huang, Shihao Ren, Jia Guo, Wensen Feng, and Kai Zhang. Wt-mvsnet: Window-based transformers for multi-view stereo. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8564–8576. Curran Associates, Inc., 2022.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[Luo *et al.*, 2019] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *ICCV*, pages 10451–10460. IEEE, 2019.

[Ma *et al.*, 2021] Xinjun Ma, Yue Gong, Qirui Wang, Jingwei Huang, Lei Chen, and Fan Yu. Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5732–5740, 2021.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Peng *et al.*, 2022] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth esti-

mation for multi-view stereo: A unified representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[Scharstein and Szeliski, 2002] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[Schönberger and Frahm, 2016] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113. IEEE Computer Society, 2016.

[Schöps *et al.*, 2017] Thomas Schöps, Johannes L. Schönberger, 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 *CVPR*, pages 2538–2547. IEEE Computer Society, 2017.

[Sun *et al.*, 2021] 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.

[Thomee *et al.*, 2016] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016.

[Tola *et al.*, 2012] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Mach. Vis. Appl.*, 23(5):903–920, 2012.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2021] 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.

[Wang *et al.*, 2022a] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: iterative probability estimation for efficient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8606–8615, 2022.

[Wang *et al.*, 2022b] Xiaofeng Wang, Zheng Zhu, Guan Huang, Fangbo Qin, Yun Ye, Yijia He, Xu Chi, and Xingang Wang. Mvster: epipolar transformer for efficient multi-view stereo. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 573–591. Springer, 2022.

[Wei *et al.*, 2021] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6187–6196, 2021.

[Xu and Tao, 2020] Qingshan Xu and Wenbing Tao. Pvsnet: Pixelwise visibility-aware multi-view stereo network. *CoRR*, abs/2007.07714, 2020.

[Yan *et al.*, 2020] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision*, pages 674–689. Springer, 2020.

[Yang *et al.*, 2020a] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020.

[Yang *et al.*, 2020b] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *CVPR*, pages 4876–4885. IEEE, 2020.

[Yao *et al.*, 2018] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[Yao *et al.*, 2019] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, pages 5525–5534. Computer Vision Foundation / IEEE, 2019.

[Yao *et al.*, 2020] 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. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[Yi *et al.*, 2020] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV (9)*, volume 12354 of *Lecture Notes in Computer Science*, pages 766–782. Springer, 2020.

[Yu and Gao, 2020] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *CVPR*, pages 1946–1955. IEEE, 2020.

[Zhang *et al.*, 2020] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *British Machine Vision Conference (BMVC)*, 2020.

[Zhu *et al.*, 2021] Jie Zhu, Bo Peng, Wanqing Li, Haifeng Shen, Zhe Zhang, and Jianjun Lei. Multi-view stereo with transformer. *ArXiv*, abs/2112.00336, 2021.