Complex & Intelligent Systems
https://doi.org/10.1007/s40747-023-01106-3

ORIGINAL ARTICLE

DSC-MVSNet: attention aware cost volume regularization based on depthwise separable convolution for multi-view stereo

Song Zhang[1,2,3,4] · Zhiwei Wei[1,2] · Wenjia Xu[5] · Lili Zhang[1,2] · Yang Wang[1,2] · Xin Zhou[1,2,3,4] · Junyi Liu[1,2]

Received: 15 November 2022 / Accepted: 8 May 2023

Abstract

Deep learning has recently been proven to deliver excellent performance in multi-view stereo (MVS). However, it is difficult for deep learning-based MVS approaches to balance their efficiency and effectiveness. Towards this end, we propose the DSC-MVSNet, a novel coarse-to-fine and end-to-end framework for more efficient and more accurate depth estimation in MVS. In particular, we propose an attention aware 3D UNet-shape network, which first uses the depthwise separable convolutions for cost volume regularization. This mechanism enables effective aggregation of information and significantly reduces the model parameters and computation by transforming the ordinary convolution on cost volume as depthwise convolution and pointwise convolution. Besides, a 3D-Attention module is proposed to alleviate the feature mismatching problem in cost volume regularization and aggregate the important information of cost volume in three dimensions (i.e. channel, space, and depth). Moreover, we propose an efficient Feature Transfer Module to upsample the low-resolution (LR) depth map to a high-resolution (HR) depth map to achieve higher accuracy. With extensive experiments on two benchmark datasets, i.e. DTU and Tanks & Temples, we demonstrate that the parameters of our model are significantly reduced to 25% of the state-of-the-art model MVSNet. Besides, our method outperforms or maintains on par accuracy with the state-of-the-art models. Our source code is available at https://github.com/zs670980918/DSC-MVSNet.

Keywords Multi-view stereo · Depth estimation · DSC-MVSNet

Introduction

Multi-view stereo (MVS) has been extensively studied and widely applied in augmented reality and 3D reconstruction [1–6]. The goal of MVS is to reconstruct 3D scenes using a series of camera-calibrated 2D images by establishing dense correspondences, which can be formulated as an optimization problem. Thus, the optimization methods such as Markov discrete optimization [7] and spatial patch diffusion [8] are applied to solve this problem. However, the above methods

Zhiwei Wei and Wenjia Xu have contributed equally to this work.

✉ Junyi Liu
zs670980918@gmail.com

Song Zhang
zhangsong20@mails.ucas.ac.cn

Zhiwei Wei
2011301130108@whu.edu.cn

Wenjia Xu
xuwenjia@bupt.edu.cn

Lili Zhang
zhanglili86@126.com

Yang Wang
primular@163.com

Xin Zhou
zhouxin191@mails.ucas.ac.cn

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China

[2] Key Laboratory of Network Information System Technology (NIST), Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

[3] University of Chinese Academy of Sciences, Beijing 100190, China

[4] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China

[5] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Published online: 07 June 2023

may result in incomplete surfaces in scenes with weak textures or non-Lambertian surfaces [1, 9].

With the development of deep learning in recent years, Yao et al. [10] show promising results by achieving MVS with a cost volume regularization process and using deep learning to solve the optimization problem. The cost volume is composed of the confidence between matched features at different depths, and the regularization process optimizes the cost volume to obtain the depth probability distribution which is then used to obtain the depth map. Attention that the regularization here is not a widely used strategy to avoid overfitting in machine learning, but a terminology denoting the optimization process of cost volume in MVS domain. Because the accuracy of regularized cost volume directly determines the quality of the final depth map obtained from the regression, the improvement of cost volume regularization network is the major part of the later research studies, such as the P-MVSNet [11] and Cascade-MVSNet [12]. These methods can achieve better reconstruction results for fully exploiting the information of the images in multiple dimensions. But they still suffer from the problem of low efficiency. Therefore, some efficient frameworks are then proposed to improve the efficiency, such as the Fast-MVSNet [13], UCS-MVSNet [14]. They propose some lightweight strategies to reduce computation (e.g. Sparse High-Resolution Representation, Adaptive Thin Volume). Nevertheless, all of the aforementioned methods will inevitably lead to higher computation cost for their used 3D cost volume regularization structure. Some other methods such as the R-MVSNet [15] and D^2 HC-RMVSNet [16] are proposed by dividing the original 3D cost volume into smaller or lower dimension pieces, i.e. channel sliced-based cost maps or depth sliced-based cost maps. Additionally, RNN [17] and LSTM [18] are used to establish the connections between these cost maps. These methods can effectively reduce the computation cost by converting the cost volume regularization into the regularization of a group of cost maps. However, these methods can not incorporate enough context information over the cost volume like 3D-UNet and their effectiveness might need further improvement.

Therefore, how to significantly reduce the computation with effectiveness maintenance is our main research problem. First, 2D depthwise separable convolution has been a standard module in current mainstream vision tasks to improve efficiency [19–21]. It converts the ordinary convolution into depthwise convolution on channel-independent feature maps, and pointwise convolution to establish the relations between these feature maps, which seems as a RNN-like mechanism in MVS. Due to its full consideration of the information and the relations of the feature maps, 2D depthwise separable convolution can achieve a similar performance as the ordinary convolution with a much lower computation cost. However, for the MVS domain [8, 10, 22–25], the usage of 3D depth separable convolution has not been explored and needs to be explored in combination with specific application scenarios. Inspired by the above thoughts, we try to use the depthwise separable convolution working on the regularization to construct our 3D UNet-shape network, which extends the 2D depthwise separable convolution into a 3D task. Second, features of the different positions may be easily mismatched in cost volume regularization because they share a visual similarity, and it will cause similar confidence in the same position at different depths. This feature mismatching problem will seriously affect the quality of the depth map by depth regression. Attention is a practical mechanism that can achieve the above-mentioned capabilities, but conventional attention (can only perform convolution on 3D volumes, without a dimension-wise process) does not effectively consider the mutuality of information between different dimensions, e.g. depth, space. Thus, we propose a 3D-Attention module to aggregate the more important multi-dimension matching information (channel, space, and depth) of cost volume and alleviate the above problem of feature mismatching. Third, the quality of the depth map directly affects the final reconstruction. Therefore, to achieve better performance, we propose a feature transfer module to upsample the low-resolution (LR) depth map to a high-resolution (HR) depth map. In addition, the feature extraction module can obtain multi-level feature information by simultaneously incorporating low-level and high-level information learned from CNN, which can achieve accurate 3D points localization. We term our effective and efficient coarse-to-fine framework as DSC-MVSNet.

The remainder of this paper is organized as follows. In Sect. "Related work", we introduce related work, followed by an overview of our method in Sect. "Methodology". Section "Experiments" presents our experimental results for two challenging datasets. Section "Limitation analysis" discusses the limitation of our proposed method, followed by some concluding remarks in Sect. "Conclusion".

In summary, our main contributions are as follows:

- We propose a 3D UNet-shape network and firstly use the depthwise separable convolution for 3D cost volume regularization, which can effectively improve the model efficiency with performance maintained.
- We propose a 3D-Attention module to enhance the ability in cost volume regularization to fully aggregate the valuable information of cost volume and alleviate the problem of feature mismatching.
- We proposed an effective and efficient feature transfer module to upsample the LR depth map to obtain the HR depth map to achieve higher quality reconstruction.
- With extensive experiments on two benchmarks, our method demonstrates comparable or even better reconstruction results than the state-of-the-art methods with

much lower computation cost. For instance, compared to state-of-the-art methods MVSNet, our model reduces the memory by 49% while improving the accuracy by 20%.

Related work

Traditional MVS reconstruction

The MVS is achieved by creating dense correspondences from multiple images of calibrated camera poses, which can be considered an optimization problem [24]. Many optimization methods are then proposed. Due to the different presentations of scenes in MVS, these methods can be divided into three categories: voxel-based [7, 26–28], patch-based [8, 24, 29, 30] or depth map-based [10, 13, 15, 22, 23, 25]. For example, the Markov discrete optimization is applied [7] by updating the state of chain voxels with constraints including luminosity consistency, smoothness, and visibility optimized; The spatial patch diffusion [8] considers each pixel in space as a patch and optimally expands the number of patches. The non-linear optimization [24] optimizes the depth and normal vector of given seed points by using stochastic gradient descent and least squares to estimate the depth map of the image. The depth map-based methods use the depth map as an intermediate, which decouples complex dense reconstruction problems into multiple simple subproblems and enables a more flexible scene reconstruction. Many recent deep learning-based MVS methods [10, 15, 31] are also performed based on the depth map.

Deep learning-based MVS

Recently, to overcome the blemish of traditional MVS methods, many deep learning-based methods [10–13, 15, 31, 32] are also introduced. For example, MVSNet [10] proposes an end-to-end MVS framework that extracts features from multiple views by CNNs to construct the matching cost volume. Then it uses 3D CNNs to regularize the cost volume to obtain a final depth map estimate. P-MVSNet [11] proposes a hybrid 3D U-Net to infer a probability volume from the cost volume and estimate the depth maps. These methods can achieve good results for their full consideration of the multi-dimensional information of images, but they are not efficient enough. To improve the efficiency, Fast-MVSNet [13] based on Point-MVSNet [32] is proposed to solve this problem by a sparse high-resolution depth map representation and some efficient modules. However, using 3D CNNs to regularize the cost volume inevitably results in a high computation cost. Thus, some methods try to slice the cost volume into cost maps, and higher efficiency can then be achieved for they convert the cost volume regularization into the regularization of the group of cost maps. For example, R-MVSNet [15] uses convolutional GRUs instead of 3D CNNs to regularize the 2D cost maps. D^2HC-RMVSNet [16] slices cost volume into cost maps along the direction of depth, and uses a hybrid architecture DHU-LSTM which absorbs both the merits of LSTM [18] and U-Net to reduce the consumption cost. However, the structures such as RNN [17], or LSTM [18] inherently suffer a forgetfulness problem. They cannot fully consider the correlation of the cost maps and do not aggregate the multi-dimensional information of cost volume well. Also intending to improve the quality of the final reconstructed point cloud, DeepFusion [31] proposes a novel fusion strategy that accurately fuses all depth maps to obtain high quality point cloud results by balancing the geometric consistency and the predicted confidence.

Depthwise separable convolutions

The depthwise separable convolution is a useful lightweight strategy to build light and efficient networks. It is first proposed and applied in an AlexNet for image classification [19, 33] by Laurent Sifre and achieves similar performance as ordinary convolution with lower computation cost. Then a similar idea is widely applied in other frameworks for object detection [34, 35] and semantic segmentation [36, 37], such as the MobileNetV1 [20] and the MobileNetV2 [21]. Unlike ordinary convolution, the depthwise separable convolution transforms it into a depthwise convolution and a pointwise convolution. It computes each feature map independently by a channel-independent depthwise convolution and then uses a pointwise convolution to correlate each channel of feature maps to obtain the final feature map. This mechanism helps reduce the computation cost with a similar performance as the ordinary convolution. It is very similar to the strategies used in the above light MVS methods [15, 16] which slice the cost volume into cost maps and use networks such as RNN [17] and LSTM [18] to correlate the maps.

Attention mechanism

It is well known that the attention mechanism plays an important role in deep learning. Except for natural language processing [38], the attention mechanism has been widely explored in many visual problems including scene segmentation [39–41], panoptic segmentation [42], and image classification [43]. As the research progresses, some attention mechanisms incorporating convolution operations have been proposed. SE Block [44] adds a residual connection between different convolutions that assigns weights to different channels. CBAM [45] adds a spatial attention block based on SE Block [44] to achieve fine-grained allocation and processing of spatial information. However, these attention mechanisms only focus on the channel and spatial information. While for 3D cost volume, it also contains depth information. And

the value of cost volume indicates the similarity between features, so there may be similarity confidence between different depths in the same spatial location of the same channel due to similar features. And just using the above attention mechanisms can not pay more attention to the more important depth information of cost volume. Therefore, we propose a depth attention mechanism combined with the original attention mechanisms, so that the regularization network can better optimize the matching information of cost volume, which allows us to obtain better depth maps and thus higher-quality point cloud reconstruction results.

Methodology

Our proposed DSC-MVSNet framework is a coarse-to-fine and end-to-end framework for estimating a goal depth map $\tilde{D}_r$ of the reference image I_0 from $N+1$ input images $\{I_i\}_{i=0}^{N}$ of size $H \times W \times 3$. We achieve this task with four subprocesses: Feature Extraction, Cost Volume Regularization, Depth Map Upsampling and Depth Map Refinement. The overall architecture of DSC-MVSNet is shown in Fig. 1.

In the cost volume regularization, we propose a DSC-Attention 3D UNet network based on depthwise separable convolution to significantly reduce the time and memory consumption while maintaining the performance. Moreover, to obtain high quality depth map, we also propose a feature transfer module to upsample the LR depth map.

Pipeline description

(1) *Feature extraction* (in Sect. "Informative feature extraction network"): we use an informative feature extraction network to extract the corresponding feature $F_i \in \mathbb{R}^{C\times\frac{H}{4}\times\frac{W}{4}}$ for each image I_i, where I_0 and $\{I_i\}_{i=1}^{N}$ denote the reference image and source images, respectively.

(2) *Cost volume regularization* (in Sects. "3D depthwise separable convolution (3D-DSC)", and "3D-attention module (3DA)"): we propose a DSC-Attention 3D UNet to regularize the coarse cost volume $V \in \mathbb{R}^{C\times D\times\frac{1}{8}H\times\frac{1}{8}W}$, which is constructed by the reference feature F_0 and other source features $\{F_i\}_{i=1}^{N}$.

(3) *Depth map upsampling* (in Sect. "Feature transfer module"): we propose a feature transfer module to upsample the LR depth map $\tilde{D}_s \in \mathbb{R}^{1\times\frac{1}{8}H\times\frac{1}{8}W}$ to a HR depth map $\tilde{D}_d \in \mathbb{R}^{1\times\frac{1}{4}H\times\frac{1}{4}W}$.

(4) *Depth map refinement* (in Sect. "Depth map refinement"): a Gauss–Netwon Layer is utilized to obtain the refined depth map $\tilde{D}_r \in \mathbb{R}^{1\times\frac{1}{4}H\times\frac{1}{4}W}$ by using input images $\{I_i\}_{i=0}^{N}$ and HR depth map $\tilde{D}_d$. Finally, we fuse the refined depth maps to obtain point clouds as the result.

3D depthwise separable convolution (3D-DSC)

Inspired by the mechanism of 2D depthwise separable convolution, we try to decrease the computation of 3D cost volume regularization by proposing 3D-DSC to replace ordinary 3D CNNs. We may have different dividing strategies for the applied 3D convolution due to it is a 3D task. But the cost volume regularization is constructed by matching similarities between feature points at different spatial positions in different views at different depths. Thus, we divide 3D CNN into 3D depthwise convolution (depthwise is depth-dimension and can perform cost aggregation for cost volume information in depth dimension) and 3D pointwise convolution (pointwise is space-dimension and perform cost aggregation for cost volume information in spatial dimension), which is consistent with the form of cost volume. The schematic of 3D-DSC is shown in the lower left part of Fig. 1.

(1) *3D depthwise convolution* The 3D depthwise convolution is performed over the cost volume in each channel independently to obtain the channel-independent intermediate feature maps, as defined in Eq.(1):

$$Conv_{\text{Depth}}(V)_{(i,j,u)} = \sum_{k,l,m}^{K,L,M} W_{1(k,l,m)} \odot V_{(i+k,j+l,u+m)} \tag{1}$$

where W_1 represent the weight of 3D depthwise convolution, $V \in \mathbb{R}^{C\times D\times H\times W}$ represent the cost volume, i, j, u represent the position index, K, L, M denote the kernel size of convolution, and $\odot$ denotes the element-wise product.

(2) *3D pointwise convolution* The 3D pointwise convolution acts on these channel-independent feature maps to aggregate the channel-wise information, as defined in Eq.(2):

$$Conv_{\text{Point}}(\hat{V})_{(i,j,u)} = \sum_{n}^{N} W_{2(n)} \cdot \hat{V}_{(i,j,u,n)} \tag{2}$$

where W_2 represent the weight of 3D pointwise convolution, $\hat{V} \in \mathbb{R}^{C\times D\times H\times W}$ represent the intermediate feature maps, N denotes the kernel size of convolution.

The two convolutions are performed sequentially to form a complete convolution. And the mathematical formulations are defined as Eq. (3):

$$Conv_{\text{SepConv}}(V) = Conv_{\text{Point}}\left(Conv_{\text{Depth}}(V)\right) \tag{3}$$

Here we compare our 3D-DSC regularization scheme with other mainstream regularization schemes theoretically, to demonstrate the effectiveness of our scheme. We display the

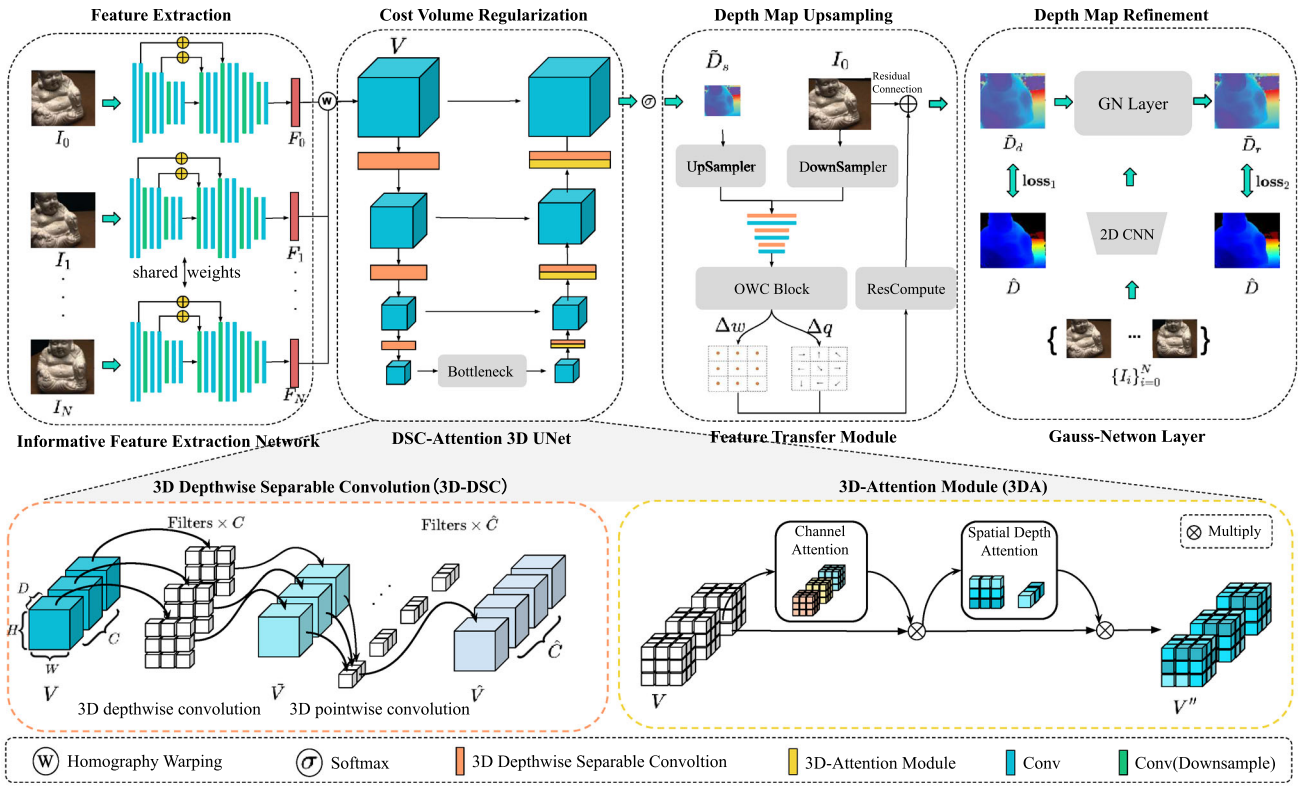
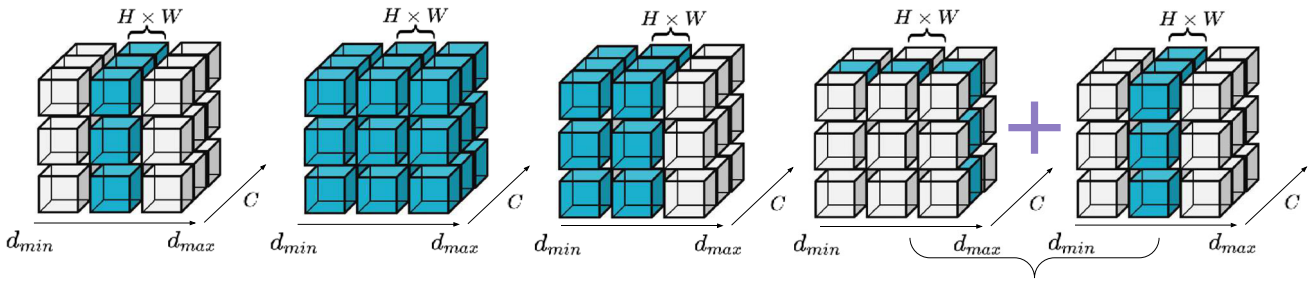


Fig. 1 The architecture of the DSC-MVSNet. In the first part, we use an informative feature extraction network to extract features to build the coarse cost volume. In the second part, we use our DSC-Attention 3D UNet to regularize the cost volume. In the third part, we use the FTM to upsample the LR depth map. In the forth part, we use the Gauss-Newton layer [13] to further refine the depth map. The two bottom parts are used for cost volume regularization. The lower left part is the schematic of our 3D depthwise separable convolution. The lower right part is the schematic of our 3D-Attention module



(a) Spatial Regularization (b) 3D CNN Regularization (c) Recurrent Regularization (d) 3D DSC Regularization

Fig. 2 Illustration of different regularization schemes. We denote the receptive field of voxels in cyan during the regularization. Horizontal is the depth dimension and vertical is the channel dimension. H and W denote the height and width respectively. In this figure, we set H and W as one dimension

four regularization schemes in Fig. 2: (a) spatial Regularization (SR) [46] is a cost aggregation method, it filters cost volume at different depths. However, due to the small receptive field, the regularization results of SR are highly affected; (b) 3D CNN Regularization (3D-CNN) [10] is a CNN-based method, it uses 3D CNNs to obtain a larger receptive field for cost volume regularization. But it causes much more computation cost; (c) recurrent Regularization [15] is an RNN-based method, it proposes sequential processing to divide the cost volume into depth-independent cost maps to reduce computation cost; (d) our 3D-DSC Regularization is a DSC-based method, we split the cost volume into intermediate feature maps, then apply a point-wise convolution to establish the relations between these intermediate feature maps to maintain the performance of the model. Our method can obtain a larger receptive field when compared to SR. While 3D CNN

Table 1 Comparison of the ordinary 3D convolution (3D-CNN), depthwise convolution (Depthwise-Conv), pointwise convolution (Pointwise-Conv) and 3D depthwise separable convolution (3D-DSC)

Convolution	Computation
3D-CNN	$M \times K^3 \times C \times \hat{C}$
Depthwise-Conv	$M \times K^3 \times C$
Pointwise-Conv	$M \times C \times \hat{C}$
3D-DSC	$M \times K^3 \times C + M \times C \times \hat{C}$

Set $H \times W \times D = M$

regularization can obtain better performance, it also incurs higher computational cost. On the other hand, our scheme can achieve similar performance with lower cost. Moreover, the recurrent regularization scheme and our regularization scheme are two different but similar ideas, both of us split cost volume into intermediate feature maps to reduce the computation cost. Therefore, we conclude that adopting the 3D-DSC as our regularization scheme is both feasible and effective.

Then we compare the efficiency of our 3D-DSC and 3D-CNN. Assuming the cost volume is $V \in \mathbb{R}^{C\times D\times H\times W}$ and the goal cost volume is $\hat{V} \in \mathbb{R}^{\hat{C}\times D\times H\times W}$, and the convolution kernel size is K, the computation cost of the ordinary 3D convolution and our proposed 3D depthwise separable convolution is shown in Table 1. We can see from the results that the computation cost of ordinary 3D convolution is $(K^3 \times \hat{C})/(K^3 + \hat{C})$ times that of 3D depthwise separable convolution. For instance, when $K = 3$ and $\hat{C} = 32$, the computation cost of our 3D-DSC convolution is around $\frac{1}{14}$ of 3D-CNN. Thus, our regularization scheme 3D-DSC will be more efficient than 3D-CNN based models. In summary, we have analyzed the effectiveness and efficiency separately, which demonstrates the feasibility of our 3D-DSC as a regularization scheme.

3D-attention module (3DA)

Although the cost volume information can be effectively aggregated after the 3D-DSC, there is still a feature mismatching problem affecting the cost volume quality. The feature mismatching problem happens when features from different key points are mistakenly matched, which will cause similarity confidence at different depths of the cost volume, and finally results in inaccurate depth estimation. Specifically, as shown in Fig. 3, a reference feature matches two similar source features at different depths (the two hands from the Buddha statue), and the confidences of different depths are similar in the cost volume. These similar confidences will affect the quality of the depth map regressed by Eq. 8.

Since attention mechanisms can highlight important information by calculating different weights, we here use an attention mechanism to address the feature mismatching problem. We propose a 3D-Attention module, which alleviates this problem by computing an attention weight using the information of the whole cost volume to enhance or weaken similar confidence in different depths. The schematic of the module is depicted in the lower right part of Fig. 1, and it consists of two blocks.

(1) **Channel attention block**. A channel attention block performs attention for channel wise information. It is constructed by a multi-layer perceptron (MLP) which acts on the channel of cost volume $V \in \mathbb{R}^{C\times D\times H\times W}$ to obtain the channel attention enhancement weights $\hat{W}$. We multiply the channel weights W' with the cost volume V to obtain the channel-refined cost volume $V' \in \mathbb{R}^{C\times D\times H\times W}$. The formula of channel attention block is defined as Eq. 4:

$$\hat{W} = \sigma(MLP(MaxPool(V)) + MLP(AvgPool(V))) \tag{4}$$

where $MaxPool$ is max pooling, $AvgPool$ is avg pooling. $\hat{W} \in \mathbb{R}^C$ denotes the channel attention enhancement weights, and both of two parts share weights of MLP.

(2) **Spatial depth attention block**. A spatial depth attention block is proposed to alleviate the problem of similarity confidence. Different from the ordinary attention, which uses full perception (without distinguishing between space and depth), the spatial depth attention block perceives cost information according to the composition of the cost volume in two different dimensions, e.g. space and depth, respectively. First, we use a spatial-oriented anisotropic [11] convolutions with kernel sizes of $1\times7\times7$ (different positions at same depth) to filter cost volume along the spatial direction to reduce noise while maintaining useful matching information at the same depth. It provides more accurate spatial information for next depth-oriented convolution. Then a depth-oriented anisotropic convolution with kernel sizes of $7\times1\times1$ (different depths at same position) acts on depth dimension, it effectively enhances or weakens matching information at different depths at the same spatial location (illustration shown in Fig. 3). Finally, we use an isotropic [11] convolution with kernel sizes of $7 \times 7 \times 7$ acts on multi-dimension (space, depth) to fully aggregate information from above processes. The formula of spatial depth attention block is defined as Eq. (5):

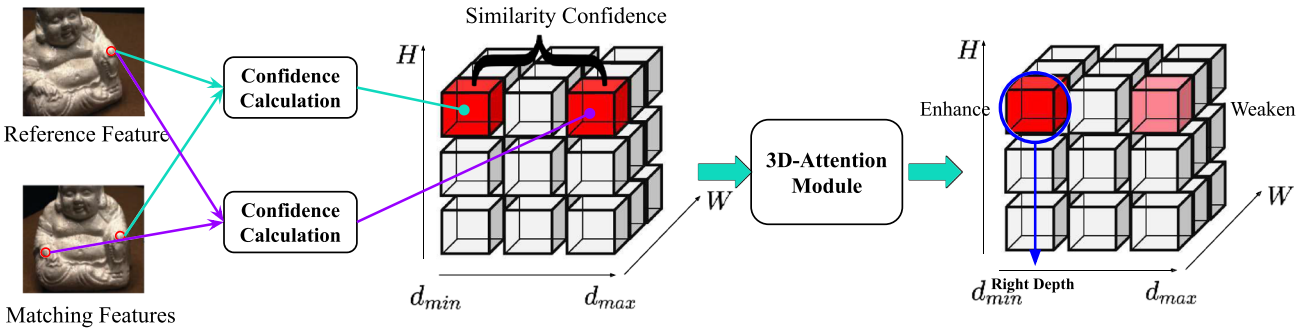


Fig. 3 Illustration of the problem of similarity confidence at different depth and use 3DA to alleviate it. Red voxels represent the similarity confidence; For representation of cost volume, we have excluded the channel dimension; The light red indicates that the confidence is weakened

$$\tilde{W} = \sigma\left(f^{7\times7\times7}\left(\left[f^{1\times7\times7}\left(\left[MaxPool(V'),\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\times AvgPool(V')\right]\right), f^{7\times1\times1}\left(\left[MaxPool(V'),\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\times AvgPool(V')\right]\right)\right]\right)\right) \tag{5}$$

where σ is the activation function; $\tilde{W} \in \mathbb{R}^{1\times D\times H\times W}$ is the spatial depth weight; $f^{1\times7\times7}$ is the spatial oriented convolution, $f^{7\times1\times1}$ is the depth oriented convolution and $f^{7\times7\times7}$ is the overall convolution.

We form a 3D-Attention module by cascading these two blocks. As shown in Fig. 3, the confidence of right depth is enhanced by using our module. The formula of 3D-Attention module is defined as Eq. 6:

$$\begin{aligned} V' &= V \times \hat{W} \\ V'' &= V' \times \tilde{W} \end{aligned} \tag{6}$$

where $V'' \in R^{C\times D\times H\times W}$ is the attention-weighted cost volume.

After regularization, we use a softmax operation (Eq. 7) in the depth direction to regress all the values between [0, 1] to form our probability volume P for depth estimation. Finally, we multiply different depth hypothesis plane values with the probability volume P to obtain the LR depth map $\tilde{D}_s$. The formula is defined as Eq. 8:

$$P = \text{softmax}(V'') \tag{7}$$

$$\tilde{D}_s = \sum_{d=d_{min}}^{d_{\max}} d \times P(d) \tag{8}$$

Feature transfer module

The high-resolution (HR) depth map obtained by upsampling directly affects the quality of the point cloud results. To obtain a high resolution and precise depth map, we propose a Feature Transfer Module (FTM) for the low-resolution (LR) depth map upsampling. The third part of Fig. 1 shows the framework of our FTM module.

The inputs of FTM are a three-channel reference image $I_0 \in \mathbb{R}^{3\times H\times W}$ and a single-channel LR depth map $\tilde{D}_s \in \mathbb{R}^{1\times\frac{1}{8}H\times\frac{1}{8}W}$. To unify the scale of inputs, we first use the bicubic interpolation algorithm [47] to upsample the LR depth map $\tilde{D}_s$ to obtain a larger scale depth map $\tilde{D}'_s \in \mathbb{R}^{1\times\frac{1}{4}H\times\frac{1}{4}W}$. And we downsample the reference image into a 16-channel image $I'_0 \in \mathbb{R}^{16\times\frac{1}{4}H\times\frac{1}{4}W}$ by a downsample layer. After unification, we propose a common offset and weight extraction backbone to obtain the offset $\Delta p_{I'_0} \in \mathbb{R}^{k^2\times\frac{1}{16}H\times\frac{1}{16}W}$ and weight $\Delta w_{I'_0} \in \mathbb{R}^{k\times\frac{1}{16}H\times\frac{1}{16}W}$ of reference image and the offset $\Delta p_{\tilde{D}'_s} \in \mathbb{R}^{k^2\times\frac{1}{16}H\times\frac{1}{16}W}$ and weight $\Delta w_{\tilde{D}'_s} \in \mathbb{R}^{k\times\frac{1}{16}H\times\frac{1}{16}W}$ of LR depth map, respectively. This backbone contains a seven convolutional feature extraction network, a offset convolution, a weight convolution, and a sigmoid layer. The equation of this backbone is defined as Eq. (9):

$$\begin{aligned} \Delta q_{input} &= f_{oc}(f_{FE}(input)),\ \ input \in [I'_0, \tilde{D}'_s] \\ \Delta w_{input} &= sigmoid(f_{wc}(f_{FE}(input))) \end{aligned} \tag{9}$$

where f_{FE} represents the extraction network, f_{oc} represents the offset convolution, f_{wc} represents weight convolution, and the $sigmoid$ represent the sigmoid layer.

Then we use the OWC Block to compute the weight $\Delta w \in \mathbb{R}^{\frac{k^2}{16}\times\frac{1}{4}H\times\frac{1}{4}W}$ and offset $\Delta q \in \mathbb{R}^{\frac{k^2}{8}\times\frac{1}{4}H\times\frac{1}{4}W}$ for guiding depth map upsampling, where k is a hyperparameter and we set $k = 12$. In detail, we multiply the corresponding offsets $\Delta p_{I'_0}$, $\Delta p_{\tilde{D}'_s}$ and weights $\Delta w_{I'_0}$, $\Delta w_{\tilde{D}'_s}$, and then pass the result through PixelShuffle to get the goal offset Δq and weightΔw. Then we use the offset to guide feature sampling and multiply the sampled features with the weight to obtain the final result. Finally, we obtain the HR depth map by a residual addition block. The equation of above process is defined as Eq. (10):

$$\begin{aligned}\Delta q &= f_{ps}(\Delta p_{I_0'} \odot \Delta p_{\tilde{D}_s'})\\ \Delta w &= f_{ps}(\Delta w_{I_0'} \odot \Delta w_{\tilde{D}_s'})\\ D_{res} &= \Delta w \odot f_{gs}(\Delta q, \tilde{D}_s')\\ \tilde{D}_d &= D_{res} + \tilde{D}_s'\end{aligned} \tag{10}$$

where f_{ps} represents the PixelShuffle [48] operation of PyTorch, f_{gs} represents the grid_sample function of PyTorch, D_{res} represents the depth residual, $\odot$ denotes the element-wise product.

Other modules

Informative feature extraction network

In the feature extraction process, many previous methods [10, 11, 13, 15, 49] only use sequential convolution operations to extract the feature map from input images $\{I_i\}_{i=0}^{N}$, which only contain the high level semantic information. And the loss of low level spatial information will affect the quality of reconstruction results. Thus, we propose an informative feature extraction network using the skip connection to propagate low level spatial information to aggregate the multi-level feature information. This network has three components (Encoder, Decoder, Adjuster), and the architecture details is provided in Table 2.

Cost volume construction

Following the previous methods [12, 13, 15, 32, 50], to build the cost volume V, we use the same differentiable homography to warp all feature maps into different fronto-parallel planes of the reference camera to construct N feature volumes $\{V_i^f\}_{i=1}^{N}$. Then we adopt the same cost metric [15] to aggregate them into the cost volume V. The equation of cost metric is defined as Eq. (11):

$$V = \frac{\sum_{i=1}^{N}\left(V_i - \overline{V_i}\right)^2}{N} \tag{11}$$

$\overline{V_i}$ is the average volume of all feature volumes.

Depth map refinement

The quality of the HR depth map $\tilde{D}_d$ and obtain the refined depth map $\tilde{D}_r \in \mathbb{R}^{\frac{1}{4}H \times \frac{1}{4}W}$ obtained in previous step is insufficient and needs to be refined. And the Gauss–Netwon Layer is an effective and efficient module for depth map refinement in Fast-MVSNet [13]. Therefore, we use a Gauss–Netwon Layer to refine the HR depth map $\tilde{D}_d$ and obtain the refined depth map $\tilde{D}_r \in \mathbb{R}^{1 \times \frac{1}{4}H \times \frac{1}{4}W}$ for MVS reconstruction.

Training loss

Following the previous methods [10, 32], we compute the average absolute value error between the predicted depth map and ground truth depth map as our training loss as Eq. (12):

$$\text{Loss} = \sum_{p \in \mathbf{p}_{\text{valid}}} \|\tilde{D}_d(p) - \hat{D}(p)\|_2 + \lambda \cdot \left\|\tilde{D}_r(p) - \hat{D}(p)\right\|_2 \tag{12}$$

where $\tilde{D}_d$ denotes the HR depth map, $\tilde{D}_r$ denotes the refined depth map, $\hat{D}$ denotes the Ground Truth Depth Map, $\mathbf{p}_{\text{valid}}$ denotes the valid point set of the Ground Truth Depth Map, λ is used to balance $\text{loss}_1(p)$ and $\text{loss}_2(p)$. In the training process, we usually set λ to 1.0.

Experiments

In this section, we first introduce the experimental settings in this paper, then quantitatively and qualitatively demonstrate the performance on the DTU dataset, and finally verify the generalization ability of the proposed work on the TnT dataset.

Experimental settings

Dataset

The DTU dataset [51] is a large-scale dataset that is captured with precise camera pose and lighting conditions using robot arm control in the laboratory. The dataset consists of the images, real point clouds, and their obtained camera parameters of 128 scenes with 7 different lighting conditions. Each scene has 49 or 64 images with a resolution of 1600 × 1200 and corresponding internal and external camera parameters for training. The dataset provides calibrated images and real point clouds, and Yao et al. [10] divide it into training set, validation set and test set.

The Tanks & Temples (TnT) [9] is captured from real outdoor sensors, which is different from DTU [51]. These outdoor scenes contain a variety of different lighting conditions, reflection conditions, and other outdoor factors that make the TnT dataset more complex than obtaining a DTU dataset under specific conditions. The intermediate set used for evaluation contains eight different scenes, namely Family, Francis, Horse, Lighthouse, M60, Panther, Playground, and Train.

Table 2 Summary of the informative feature extraction network

Input images size: $3 \times H \times W$		
Name	Layer description	Output size
Encoder		
conv0	3×3 conv, stride 1	$8 \times H \times W$
conv1	3×3 conv, stride 1	$8 \times H \times W$
conv2	5×5 conv, stride 2	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv3	3×3 conv, stride 1	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv4	3×3 conv, stride 1	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv5	5×5 conv, stride 2	$32 \times \frac{1}{4}H \times \frac{1}{4}W$
conv6	3×3 conv, stride 1	$32 \times \frac{1}{4}H \times \frac{1}{4}W$
conv7	3×3 conv, stride 1	$32 \times \frac{1}{4}H \times \frac{1}{4}W$
Decoder		
conv8	3×3 transposed conv, stride 2	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv9	3×3 conv, stride 1	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv10	3×3 conv, stride 1	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
sp	Add conv4 & conv10 features	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv11	3×3 transposed conv, stride 2	$8 \times H \times W$
conv12	3×3 conv, stride 1	$8 \times H \times W$
conv13	3×3 conv, stride 1	$8 \times H \times W$
sp	Add conv7 & conv13 features	$8 \times H \times W$
Adjuster		
conv14	5×5 conv, stride 2	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv15	3×3 conv, stride 1	$16 \times \frac{1}{2}H \times \frac{1}{2}W$
conv16	5×5 conv, stride 2	$32 \times \frac{1}{4}H \times \frac{1}{4}W$
conv17	3×3 conv, stride 1 (no BN&ReLU)	$32 \times \frac{1}{4}H \times \frac{1}{4}W$

Each convolutional layer represents a block of convolution, batch normalization (BN) and ReLU. 'sp' means skip connection

Implement details

Training The proposed DSC-MVSNet is implemented using PyTorch and trained on the DTU training set. The ground truths for evaluation in DTU are represented as real point clouds. The depth maps for training our framework are obtained using the screened Poisson surface reconstruction algorithm (SPSR) [52]. In the training process, the input image resolution is set as 640×512, and the number of training views is set as $\mathbf{N} = 3$. The selection of reference images and source images is the same as MVSNet [10]. The virtual hypothetical depth plane value is set as $\mathbf{D} = 48$ and $\mathbf{D} = 96$ for training, and the depth values are sampled within the range $[425\,mm, 921\,mm]$. The learning rate is set using the RMSProp optimizer and the initial learning rate is set to 0.0008, and the decay weight is set as 0.002 every epoch. The batch size is set as 16 and trained on $6\times$ NVIDIA GTX 2080ti GPU devices. Our best model is trained with two stages: (1) We use a virtual hypothetical depth plane of 48 for training, set 6 epochs for end-to-end training with the DSC-Attention 3D UNet and Feature Transfer Module, and use 12 epochs for overall training. (2) We retrain our network based on the best model obtained in the first stage with 10 epochs in the hypothetical depth plane of 96. The best model for the second stage is selected as our evaluation model.

Testing The model obtained in the training process is tested on DTU test dataset [51]. We use 5 adjacent images of 1280×960 as the input. The hypothetical depth plane for testing is set as $\mathbf{D} = 128$. The evaluation of the DTU dataset [51] is performed by converting the output depth map into a predicted point cloud using the method according to Yao [10], and then comparing it with the ground truth point cloud by official Matlab code.

Evaluation metrics

To obtain comprehensive conclusions, we use three metrics for evaluating performance and three metrics for evaluating efficiency of our model. The performance evaluation metrics (Acc, Comp, and Overall) are all mentioned in DTU [51]. Acc is measured as the distance from the MVS reconstruction to the structured light reference, encapsulating the quality of the reconstructed MVS points. A lower Acc value indicates more accurate positioning of the points in the point clouds. Com-

Table 3 Quantitative results of different methods on DTU's evaluation set [51](lower is better)

Methods	Acc. (mm) ↓	Comp. (mm) ↓	Overall (mm) ↓
Camp [55]	0.835	0.554	0.695
Furu [8]	0.613	0.941	0.777
Tola [25]	0.342	1.19	0.766
Gipuma [23]	**0.283**	0.873	0.578
MVSNet [10]	0.396	0.527	0.462
R-MVSNet [15]	0.383	0.452	0.417
P-MVSNet [11]	0.406	0.434	0.420
MVSCRF [56]	0.371	0.426	0.398
PointMVSNet [32]	0.342	0.411	0.376
Fast-MVSNet [13]	0.336	0.403	0.370
Cascade-MVSNet [12]	0.325	0.385	0.355
CVP-MVSNet [49]	0.296	0.406	0.351
PVA-MVSNet [53]	0.372	0.350	0.361
Vis-Net [57]	0.369	0.361	0.365
MVSNet++ [54]	0.407	0.345	0.376
UCS-Net [14]	0.338	0.349	**0.344**
D^2HC-RMVSNet [16]	0.395	0.378	0.386
DeepFusion [31]	0.357	0.502	0.429
AA-RMVSNet [58]	0.376	0.339	0.357
PatchmatchNet [59]	0.427	**0.277**	0.352
Our	<u>0.316</u>	0.372	**0.344**

Bold values means the best values compared to all list values of each colume
Underline value means the second lowest values compared to all listed Acc values
Our method DSC-MVSNet outperforms all deep learning-based MVS methods in terms of reconstruction accuracy, and has a better result in terms of overall. The top group of methods are traditional MVS methods, and the bottom group exhibits the deep learning based-methods

pleteness is measured as the distance from the reference to the MVS reconstruction, encapsulating how much of the surface is captured by the MVS reconstruction. A lower Comp value means that we reconstruct more point cloud surfaces. Acc and Comp are calculated using the official Matlab code provided by DTU [51]. Overall is calculated as the average of Acc and Comp to evaluate overall reconstruction quality. The metrics used to evaluate efficiency are Parameters, Memory, and Time, which are widely adopted in previous methods [13, 53, 54].

Evaluation on DTU dataset

Comparison of the models performance

We compare our DSC-MVSNet with two groups of state-of-the-art methods: traditional MVS methods e.g. Camp [55], Furu [8], Toal [25], Gipuma [23]; and deep learning-based MVS methods e.g. MVSNet [10], R-MVSNet [15], Fast-MVSNet [13], CVP-MVSNet [53], UCS-Net [14], DeepFusion [31], PatchmatchNet [59]. Table 3 shows the results of the DTU [51] dataset. We have the following observations: our method establishes the state-of-the-art overall performance by comparing two groups of methods. For instance, DSC-MVSNet achieves significant improvement in Overall performance: 50.5% (Camp), 55.7% (Furu), 55.1% (Tola), 40.4% (Gipuma), 25.5% (MVSNet), 17.5% (R-MVSNet), 7.0% (Fast-MVSNet), 2.1% (CVP-MVSNet), 19.8% (DeepFusion) and 2.3% (PatchmatchNet). It indicates that our model can reconstruct a sufficient number of surfaces and the spatial locations of the points on these surfaces are accurate enough. In the generalized Acc metric that is more challenging, our method achieves notably gains over state-of-the-art methods: we achieve 0.316 on Acc. Although the Gipuma [23] method has the highest Acc, its Comp is much higher than our proposed method (0.873 vs 0.372). And compared to deep learning-based methods, our method achieves comparable results to CVP-MVSNet [53] (0.316 vs 0.296). This shows that our network is accurate in estimating the position of each point obtained from the reconstruction. Our DSC-MVSNet is comparable to or better than SOTA methods in terms of Comp. However, the PatchmatchNet [59] has the lowest Comp, its Acc and Overall are higher than our proposed method (0.427 vs 0.316; 0.352 vs 0.344). It indicates that our method can reconstruct more of the target surfaces to meet the low Comp. Thus, these results demonstrate that our proposed method has a better or comparable performance compared to the majority of state-of-the-art methods.

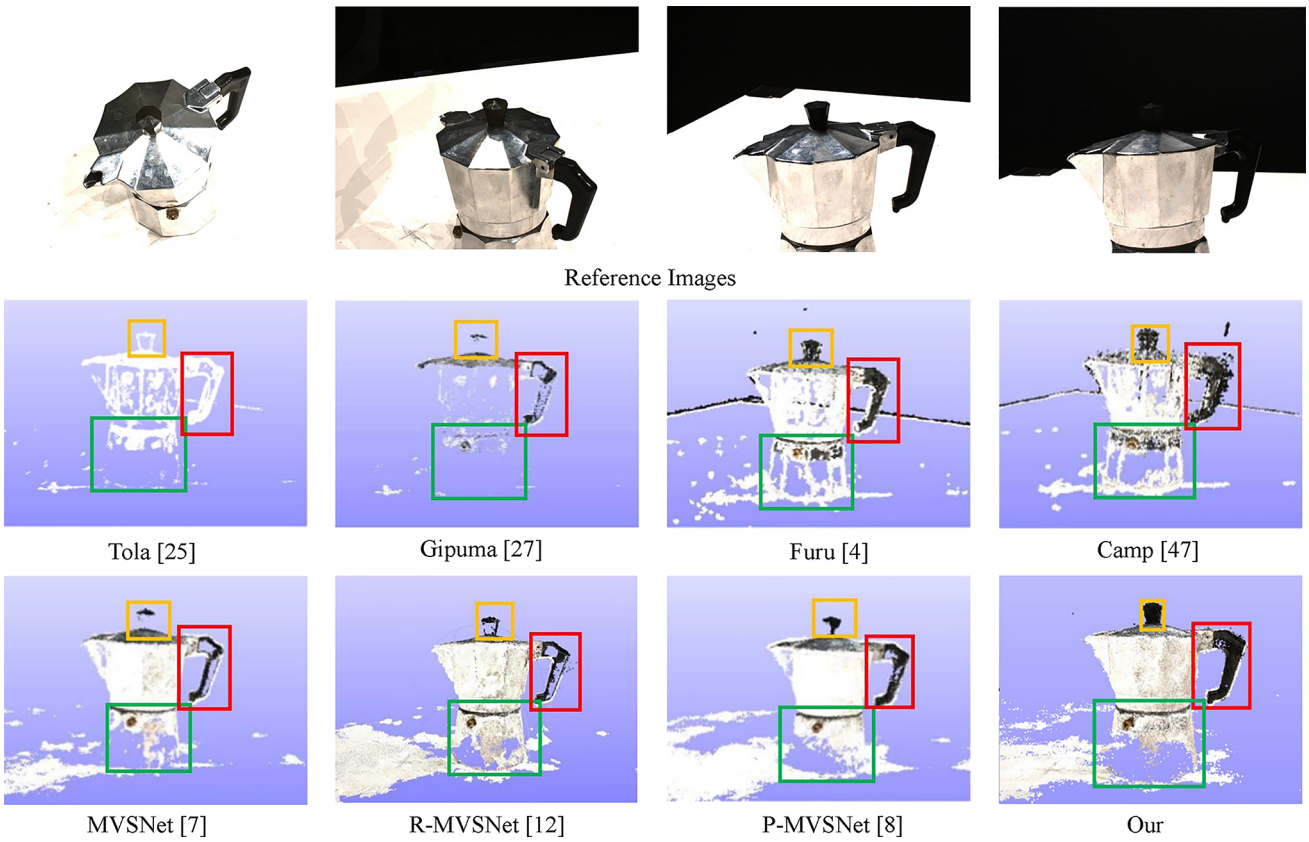


Fig. 4 Visualization of the reconstructed point cloud models for scan77 in DTU dataset by different methods. The results are directly cited from the paper P-MVSNet [11]. Three important parts: cover (yellow), handle (red) and base (green) are highlighted. Although the reference image sequences contain many reflective regions which is hard for 3D model reconstruction, our DSC-MVSNet reconstructs a more complete and more accuracy point clouds compared to the most of exist methods

Figure 4 shows the qualitative comparison results (Scan 77 in DTU [51]) between DSC-MVSNet and most of state-of-the-arts methods (Tola [25], Gipuma [23], Furu [8], Camp [55], MVSNet [10], R-MVSNet [15], P-MVSNet [11]). The colored boxes (red, yellow, green) shown in the figure, our method DSC-MVSNet reconstructs a more complete point cloud, which corresponds to the Comp value in Table 3. We think the improvement of completeness benefits from the introduction of the 3DA, which can alleviate the feature mismatch problem to improve the quality of depth map.

We further compare our DSC-MVSNet with R-MVSNet [15] on some scenes (Scan 1, Scan 75, Scan 110, Scan 114) of DTU [51]. Because R-MVSNet can handle large-scale scenarios for 3D model reconstruction [54]. Figure 5 shows the visualization of various reconstructed point cloud models of DTU dataset. The comparisons reveal that our DSC-MVSNet reduces a considerable number of outliers compared to R-MVSNet. That shows our DSC-MVSNet estimates the position of each point to be reconstructed accurately, and the conclusion corresponds to the ACC value in Table 3. Furthermore, it is worth mentioning that our network occupies less memory and runs faster than R-MVSNet. We think the above improvements benefit from the introduction of the 3D-DSC.

Comparison of the models efficiency

We compared the efficiency of different methods by reporting their model parameters, memory consumption, and runtime (some results are obtained from official reports). Table 3 and Table 4 show that our framework has lower model parameters, memory consumption, and runtime than most state-of-the-art deep learning methods, with very competitive performance. Although our method runs with slower runtime, it uses smaller memory consumption and parameters (5.5 GB, 253,585). We also compared our network with various state-of-the-art methods, such as Fast-MVSNet [13], Cascade-MVSNet [12], PVA-MVSNet [53], UCS-Net [14], and D^2HC-RMVSNet [16]. Table 4 shows that DSC-MVSNet achieves lower or comparable efficiency results compared to SOTA methods. Memory consumption directly affects the environment setting for model training. In terms of

Fig. 5 Visualization of several scenes on DTU dataset between R-MVSNet [15] (left) and our DSC-MVSNet (right). The point cloud results clearly show that our method DSC-MVSNet achieve better reconstruction results even with much lower parameters

Table 4 Comparison on the parameters, memory and time consumption on the evaluation DTU [51] dataset

Methods	H, W	Parameters	Memory (GB)	Time (s)
MVSNet [10]	1152, 864	1084304	10.8	1.21
R-MVSNet [15]	1600, 1184	799365	6.7	2.35
PointMVSNet [32]	1600, 1152	698936	8.7	5.44
Fast-MVSNet [13]	1280, 960	455472	5.3	0.6
Cascade-MVSNet [12]	1152, 864	934304	5.3	0.49
CVP-MVSNet [49]	1600, 1152	551585	8.7	1.72
PVA-MVSNet [53]	1600, 1184	338129	17.3	0.95
UCS-Net [14]	1152, 864	938496	5.4	0.76
D^2HC-RMVSNet [16]	1600, 1200	338257	6.6	29.15
Our	1600, 1152	253585	5.5	0.74

Some results are obtained from PVA-MVSNet [53]

Table 5 VRAM and time consumption of the inference on DTU [51] dataset

#Source	2	3	4	5	6	7	8	9
VRAM (MB)	3810	4302	4966	5539	6289	7044	7732	8546
TIME (s)	0.45	0.55	0.64	0.74	0.87	0.99	1.11	1.23

memory consumption, Fast-MVSNet and Cascade-MVSNet achieve the lowest memory among SOTA methods. Our method also has similar memory consumption to the above methods (5.3 GB vs 5.5 GB), and reduces parameters by 72% over Cascade-MVSNet and 44% over Fast-MVSNet. Although PVA-MVSNet and D^2HC-RMVSNet are similar to DSC-MVSNet in terms of model parameters, we reduce memory consumption by 68% over PVA-MVSNet [53] and achieve faster runtime than D^2HC-RMVSNet [16] (5.5 GB vs 17.3 GB; 0.74 s vs 29.15 s). Similarly, UCS-Net [14] is comparable to our method in terms of memory and time, but we reduce parameters by 73% compared to UCS-Net [14] on the DTU [51] dataset. In conclusion, our proposed method has better or comparable efficiency than most state-of-the-art methods.

Then we discuss the memory and the time consumption of the inference phase. The size of the inputs is $H \times W = 1600 \times 1152$, and the hypothetical depth plane is set as $\mathbf{D} = 96$. Table 5 shows the results of the inference on the DTU [51] dataset w.r.t. the number of sources. It demonstrates that the memory occupied by inference and the inference time is linearly increasing with the number of sources.

Ablation experiments

The ablation experiments are also conducted on the DTU dataset to illustrate our method's efficiency and effectiveness. The network only with the 3D UNet-shape network for cost volume regularization is taken as a baseline for ablation experiments. The results are shown in Table 6.

Table 6 Ablation study on the DTU evaluation dataset [51], which demonstrates the effectiveness of different modules of our method, where model parameters, memory, and time are recorded during training

Method	Acc. (mm) ↓	Comp. (mm) ↓	Overall (mm) ↓	Parameters ↓	Memory (MB) ↓	Time (s) ↓
Baseline (+ 3D CNNs)	0.391	0.482	0.437	169024	9034	0.523
Baseline + DSC	0.398	0.470	0.434	**892713**	**3408**	**0.274**
Baseline + DSC + 3DA	0.358	0.453	0.406	170785	3422	0.324
Baseline + DSC + IFEN	0.376	0.467	0.422	208432	3488	0.338
Baseline + DSC + FTM	0.364	0.441	0.403	214177	3572	0.335
Baseline + DSC + 3DA + IFEN + FTM	**0.316**	**0.372**	**0.344**	253585	3766	0.354

Bold values means the best values compared to all list values of each colume
We use standard 3D UNet as a baseline and add different modules to the baseline separately to compare the improvement of each module to the network. i.e. 3D-Attention (3DA), Informative Feature Extraction Network (IFEN), Feature Transfer Module (FTM)

Effectiveness of DSC: Our novelty contribution is to explore the feasibility of 3D depth separable convolution as a cost volume regularization scheme in the MVS domain. As shown in Table 6, compared to Row 2 (Baseline + 3D CNNs) and Row 3 (Baseline + DSC), we can observe that replacing 3D CNN with 3D DSC in 3D UNet, which not cause a sharp decline in model performance, e.g. Acc from 0.391 to 0.398. Meanwhile, our model can greatly reduce the number of parameters, memory consumption and time. Therefore, it is feasible to use 3D DSC in the MVS domain. Based on the above phenomenon, we think that the regularization scheme we designed for cost volume plays a key role in the model. We divide 3D DSC into 3D pointwise convolution and 3D depthwise convolution, which perceives multi-dimensional cost information and aggregates in depth dimension and spatial dimension. This mechanism is similar to 3D CNN-based mechanism (as shown in Fig. 2b and d), so our model can still maintain an impressive performance, which proves the feasibility of using 3D DSC in the MVS domain.

Effectiveness of DSC 3D UNet: As shown in Table 6, compared to the baseline (+3D CNNs), the baseline using the DSC 3D UNet can effectively reduce the model parameters, memory consumption, training time, and the Acc, Comp, and Overall can also be maintained to some extent. It means a significant reduction in parameters without much accuracy loss can be achieved using the 3D depthwise separable convolution.

Effectiveness of 3D-Attention module: As shown in Table 6, the Acc, Comp, and Overall metrics can all be improved with only a slight increase in computation and memory consumption by adding the 3D-Attention module to the baseline + DSC. This means that adding the attention layer is effective and it helps to improve the information extraction of our proposed separable convolution.

As the problem of similarity confidence mentioned in Sect. "3D-Attention module (3DA)", we discuss the effectiveness of the 3DA module in solving the above problems. We illustrate separately the confidence line charts for different depths at a spatial location with 3DA (red line chart) and without 3DA (blue line chart) in Fig. 7. We can see from the charts that the confidence of the GT depth in the blue dash is very similar to the confidence of the error depth, which can lead to incorrect depth estimates when the predicted depth value (the blue dashed line) is calculated via Eq. (8), to obtain depth values that are far from the GT depth. After adding the 3DA module, we can see from the red line chart that the confidence of the GT depth has been enhanced and the confidence of the error depth has been weakened, so that we obtain a value similar to the GT depth value when calculating the predicted depth value (the red dashed line). This is also reflected in the higher Accuracy of ablation experiments with baseline + DSC + 3DA in Table 6.

Effectiveness of Informative Feature Extraction Network: As shown in Table 6, our baseline + DSC combines Informative Feature Extraction Network can achieve better performance with a small increase in the number of model parameters, memory, and time.

Effectiveness of Feature Transfer Module: We use a Feature Transfer Module in the baseline + DSC to upsample the LR depth map. Table 6 shows that the FTM can further improve the performance of our network with a small increase in model parameters, memory, and time.

The ablation reconstruction results of scan 118 of the DTU [51] when adding different modules of our method are shown in Fig. 6. As the areas identified by rectangles in Fig. 6, our baseline has higher completeness and richer detail information by combining different modules.

Generalization on TnT dataset

The Tanks & Temples (TnT) dataset [9] is widely used in previous methods [10, 12, 13, 15, 31, 32] as a benchmark. Therefore, to evaluate the generalization of our DSC-MVSNet, we perform a test on TnT and evaluate the results by uploading the point cloud to the official website. We use the best model of training on DTU without fine-tuning to evaluate the TnT dataset [9], and we set 5 adjacent images with a resolution

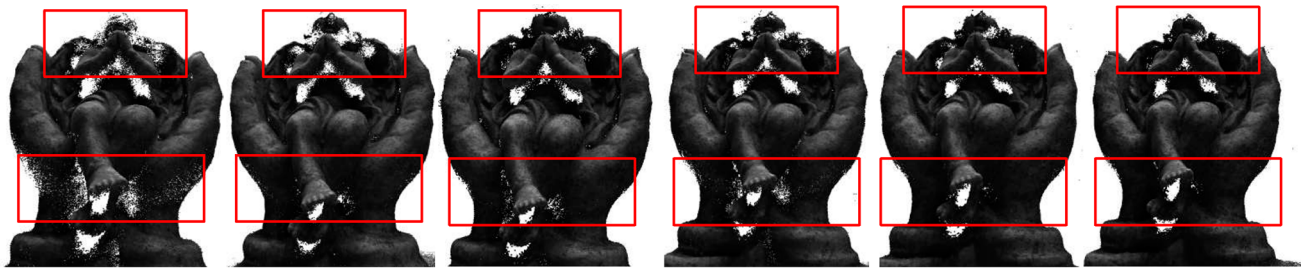


Fig. 6 Ablation reconstruction results of scan118 of the DTU dataset [51]. Two important parts: top (red) and bottom (red) are highlighted. The point cloud results show the effectiveness of each modules

Table 7 Generalization results on the Tanks & Temples benchmark [9]

Methods	Family ↑	Francis ↑	Horse ↑	Lighthouse ↑	M60 ↑	Panther ↑	Playground ↑	Train ↑	Intermediate mean ↑
Colmap [22]	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04	42.14
Pix4D [60]	64.45	31.91	26.43	54.41	50.58	35.37	47.78	34.96	43.24
OpenMVG [61] + OpenMVS [62]	58.86	32.59	26.25	43.12	44.73	46.85	45.97	35.27	41.71
MVSNet [10]	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69	43.48
R-MVSNet [15]	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38	48.40
MVSCRF [56]	59.83	30.60	29.93	51.15	50.61	51.45	52.60	39.68	45.73
PointMVSNet [32]	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06	48.27
CIDER [63]	56.79	32.39	29.89	54.67	53.46	53.51	50.48	42.85	46.76
Fast-MVSNet [13]	65.18	39.59	34.98	47.81	49.16	46.20	53.27	42.91	47.39
PatchmatchNet [59]	66.99	**52.64**	**43.24**	54.87	52.87	49.54	**54.21**	50.81	53.15
DSC-MVSNet	**68.06**	47.43	41.60	**54.96**	**56.73**	**53.86**	53.46	**51.71**	**53.48**

Bold values means the best values compared to all list values of each colume
We achieve comparable F-score results with many state-of-the-art methods. The top part of the table shows the comparison results with traditional MVS methods, and the bottom part exhibits the comparison results with deep learning based-methods

1920×1080 as the input. Meanwhile, the depth hypothesis plane is set as $\mathbf{D} = 128$.

As shown in Table 7, our model exhibits comparable results with lower consumption. Compare to traditional multi-view stereo methods (Colmap, Pix4D, OpenMVG+OpenMVS), our DSC-MVSNet obtains better reconstruction scores on all scenes. Besides, our DSC-MVSNet outperforms all listed learning-based MVS methods with a 53.48 mean F-score on Tanks and Temples intermediate [9]. And we achieve a comparable generalization performance with the state-of-the-art methods e.g. DSC-MVSNet achieves the highest accuracy on several scenes, i.e., Family, Lighthouse, M60, Panther, and Train. Figure 8 shows the error visualization calculated according to the corresponding ground truth point clouds. Our DSC-MVSNet significantly improves the precision of reconstructions compared to the recent work PatchmatchNet [59]. For example, as shown in the red boxes in Fig. 8, PatchmatchNet has more incorrect points and noise. Our method is able to obtain more accurate point positions while reducing noise, which is benefited from our proposed 3DA and FTM methods.

Limitation analysis

Although our model exhibits better or comparable performance than most of the state-of-the-art methods on the two benchmarks [9, 51], we still have some limitations. (1) For complex environmental factors (i.e. lighting conditions, reflection conditions, etc) that have never been obtained before, there are still some limitations in the accuracy of the reconstruction. Therefore, we consider improving the generalization ability of the model in future works. (2) As we use several images as input, our model is still higher than the best method in memory consumption as shown in Table 4. This motivates us to explore high-quality reconstruction with limited input images.

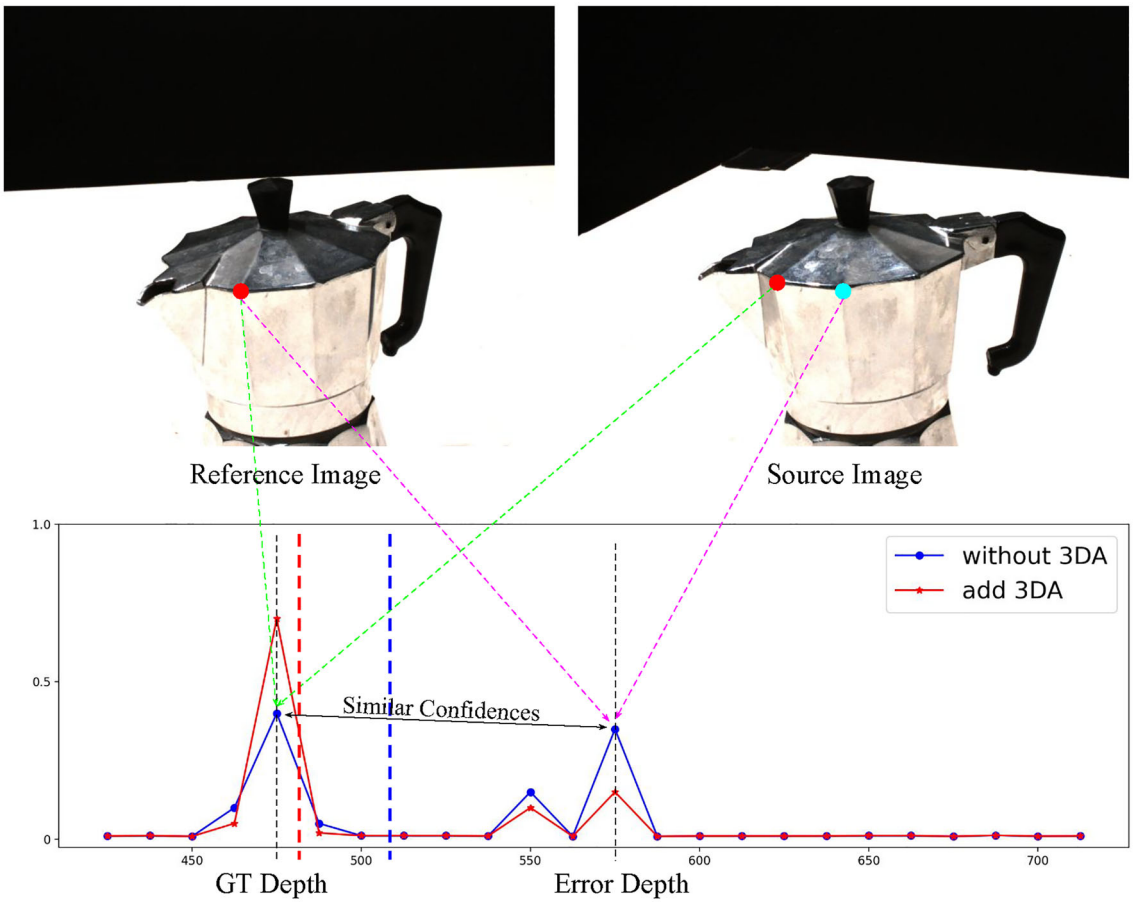


Fig. 7 We illustrate the similar confidences of an example of scan 77. On the top, we show an RGB reference image, and an RGB source image. The red point of right image is the matching point, and the green point is the mismatching point. On the bottom, we show the corresponding confidence line charts for the two examples with 3DA (red line chart) and without 3DA (blue line chart). The red dashed line represents the predicted depth value of red line chart, and the blue dashed line is the predicted depth value of blue line chart

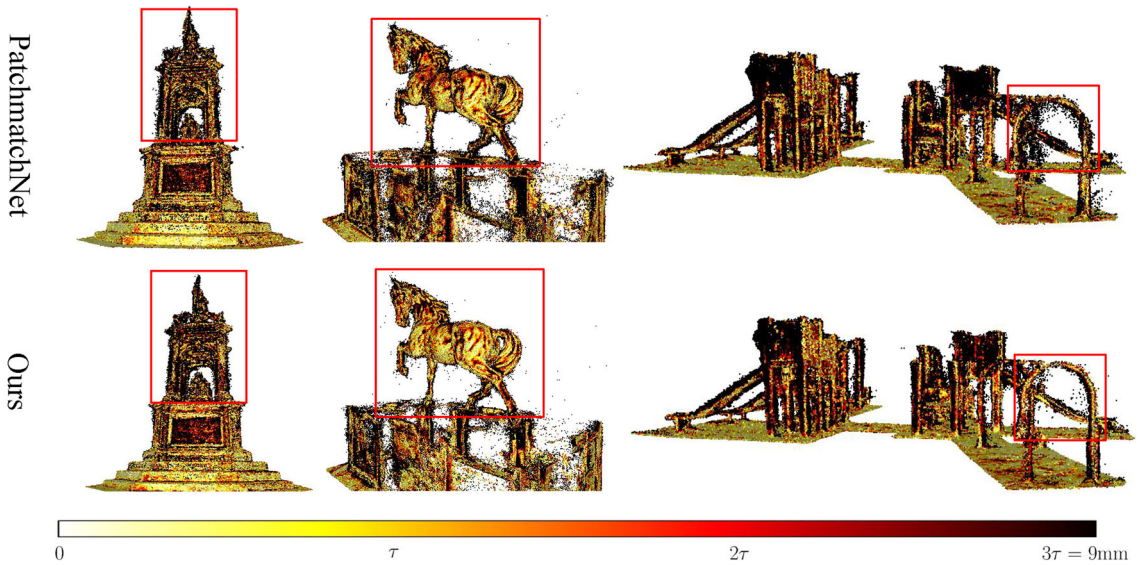


Fig. 8 Error Visualization of Francis, Horse and Playground in the Tanks and Temples intermediate dataset [9], compared with PatchmatchNet [59]

Conclusion

Our proposed DSC-MVSNet is a novel coarse-to-fine and end-to-end framework for efficient and accurate depth estimation in MVS. Firstly, we use depthwise separable convolution to construct our attention-aware 3D UNet-shaped network for cost volume regularization with lower parameters and memory cost. Additionally, we introduce a 3D-Attention module to focus on more critical information and alleviate the feature-mismatching problem. Furthermore, we propose an efficient and effective Feature Transfer Module to upsample the LR depth map. The experimental results verify the effectiveness and efficiency of our method.

References

1. Aanæs H, Jensen RR, Vogiatzis G, Tola E, Dahl AB (2016) Large-scale data for multiple-view stereopsis. Int J Comput Vision 120:153–168
2. Furukawa Y, Hernández C (2015) Multi-view stereo: a tutorial. Found Trends Comput Graph Vis 9:1–148
3. Kim H, Guillemaut J-Y, Takai T, Sarim M, Hilton A (2012) Outdoor dynamic 3-d scene reconstruction. IEEE Trans Circuits Syst Video Technol 22(11):1611–1622. https://doi.org/10.1109/TCSVT.2012.2202185
4. Michailidis G-T, Pajarola R, Andreadis I (2014) High performance stereo system for dense 3-d reconstruction. IEEE Trans Circuits Syst Video Technol 24(6):929–941. https://doi.org/10.1109/TCSVT.2013.2290575
5. Yu Q, Yang C, Wei H (2022) Part-wise atlasnet for 3d point cloud reconstruction from a single image. Knowl-Based Syst 242:108395
6. Wang P, Liu L, Zhang H, Wang T (2021) Cgnet: a cascaded generative network for dense point cloud reconstruction from a single image. Knowl-Based Syst 223:107057
7. Seitz SM (2002) Photorealistic scene reconstruction by voxel coloring. In: Proceedings., 1997 IEEE Computer Society Conference On Computer Vision and Pattern Recognition, 1997
8. Furukawa Y, Ponce J (2010) Accurate, dense, and robust multiview stereopsis. IEEE Trans Pattern Anal Mach Intell 32:1362–1376
9. Knapitsch A, Park J, Zhou Q, Koltun V (2017) Tanks and temples: benchmarking large-scale scene reconstruction. ACM Trans Graph 36(4):78–17813
10. Yao Y, Luo Z, Li S, Fang T, Quan L (2018) Mvsnet: depth inference for unstructured multi-view stereo. arXiv:1804.02505
11. Luo K, Guan T, Ju L, Huang H, Luo Y (2019) P-mvsnet: learning patch-wise matching confidence aggregation for multi-view stereo. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 10451–10460
12. Gu X, Fan Z, Zhu S, Dai Z, Tan F, Tan P (2020) Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 2492–2501
13. Yu Z, Gao S (2020) Fast-mvsnet: sparse-to-dense multi-view stereo with learned propagation and Gauss–Newton refinement. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 1946–1955
14. Cheng S, Xu Z, Zhu S, Li Z, Li EL, Ramamoorthi R, Su H (2020) Deep stereo using adaptive thin volume representation with uncertainty awareness. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 2521–2531
15. Yao Y, Luo Z, Li S, Shen T, Fang T, Quan L (2019) Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 5520–5529
16. Yan J, Wei Z, Yi H, Ding M, Zhang R, Chen Y, Wang G, Tai Y-W (2020) Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. arXiv:2007.10872
17. Mikolov T, Karafiát M, Burget L, Cernocký JH, Khudanpur S (2010) Recurrent neural network based language model. In: INTERSPEECH
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780
19. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. CoRR arXiv:1409.1556
20. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017)Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
21. Sandler M, Howard AG, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4510–4520
22. Schönberger JL, Zheng E, Frahm J-M, Pollefeys M (2016) Pixelwise view selection for unstructured multi-view stereo. In: ECCV
23. Galliani S, Lasinger K, Schindler K (2015) Massively parallel multiview stereopsis by surface normal diffusion. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 873–881
24. Goesele M, Snavely N, Curless B, Hoppe H, Seitz SM (2007) Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision, pp 1–8
25. Tola E, Strecha C, Fua P (2011) Efficient large-scale multi-view stereo for ultra high-resolution image sets. Mach Vis Appl 23:903–920
26. Ji M, Gall J, Zheng H, Liu Y, Fang L (2017) Surfacenet: an end-to-end 3d neural network for multiview stereopsis. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 2326–2334
27. Kutulakos KN, Seitz SM (2004) A theory of shape by space carving. Int J Comput Vis 38:199–218
28. Kar A, Häne C, Malik J (2017) Learning a multi-view stereo machine. In: NIPS
29. Lhuillier M, Quan L (2005) A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE Trans Pattern Anal Mach Intell 27:418–433
30. Rezaee Kaviani H, Shirani S (2018) An adaptive patch-based reconstruction scheme for view synthesis by disparity estimation using optical flow. IEEE Trans Circuits Syst Video Technol 28(7):1540–1552. https://doi.org/10.1109/TCSVT.2017.2682887
31. Wang Y, Luo K, Chen Z, Ju L, Guan T (2021) Deepfusion: a simple way to improve traditional multi-view stereo methods using deep learning. Knowl-Based Syst 221(3):106968

32. Chen R, Han S, Xu J, Su H (2019) Point-based multi-view stereo network. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 1538–1547
33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778
34. Girshick RB (2015) Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 1440–1448
35. Ren S, He K, Girshick RB, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39:1137–1149
36. Chen L-C, Papandreou G, Kokkinos I, Murphy KP, Yuille AL (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40:834–848
37. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39:640–651
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp 5998–6008
39. Dai T, Cai J, Zhang Y, Xia ST, Zhang L (2019) Second-order attention network for single image super-resolution. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
40. Zhang H, Dana K, Shi J, Zhang Z, Wang X, Tyagi A, Agrawal A (2018) Context encoding for semantic segmentation. IEEE
41. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Learning a discriminative feature network for semantic segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
42. Li Y, Chen X, Zhu Z, Xie L, Wang X (2019) Attention-guided unified network for panoptic segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
43. FeI W, Jiang M, Chen Q, Yang S, Tang X (2017) Residual attention network for image classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
44. Jie H, Li S, Gang S, Albanie S (2017) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell, p 99
45. Woo S, Park J, Lee J-Y, Kweon I-S (2018) Cbam: convolutional block attention module. In: ECCV
46. Yang Q (2012) A non-local cost aggregation method for stereo matching. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
47. Keys R (1981) Cubic convolution interpolation for digital image processing. IEEE Trans Acoust Speech Signal Process 29:1153–1160
48. Shi W, Caballero J, Huszár F, Totz J, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. IEEE
49. Yang J, Mao W, Álvarez JM, Liu M (2020) Cost volume pyramid based depth inference for multi-view stereo. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 4876–4885
50. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. ArXiv arXiv:1502.03167
51. Jensen RR, Dahl A, Vogiatzis G, Tola E, Aanæs H (2014) Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp 406–413
52. Kazhdan MM, Hoppe H (2013) Screened Poisson surface reconstruction. ACM Trans Graph 32:29–12913
53. Yi H, Wei Z, Ding M, Zhang R, Chen Y, Wang G, Tai Y-W (2020) Pyramid multi-view stereo net with self-adaptive view aggregation. arXiv:1912.03001
54. Chen P-H, Yang H-C, Chen K-W, Chen Y-S (2020) Mvsnet++: learning depth-based attention pyramid features for multi-view stereo. IEEE Trans Image Process 29:7261–7273
55. Campbell NDF, Vogiatzis G, Hernández C, Cipolla R (2008) Using multiple hypotheses to improve depth-maps for multi-view stereo. In: ECCV
56. Xue Y, Chen J, Wan W, Huang Y, Yu C, Li T, Bao J (2019) Mvscrf: learning multi-view stereo with conditional random fields. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 4311–4320
57. Zhang J, Yao Y, Li S, Luo Z, Fang T (2020) Visibility-aware multi-view stereo network. In: 31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7–10, 2020. BMVA Press
58. Wei Z, Zhu Q, Min C, Chen Y, Wang G (2021) Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. arXiv:2108.03824
59. Wang F, Galliani S, Vogel C, Speciale P, Pollefeys M (2021) Patchmatchnet: learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 14194–14203
60. Pix4d: https://pix4d.com/
61. Moulon P, Monasse RMP (2014) Openmvg. An open multiple view geometry library. https://github.com/openMVG/openMVG
62. Cernea D (2015) Openmvs: open multiple view stereovision. https://github.com/cdcseacave/openMVS
63. Xu Q, Tao W (2019) Learning inverse depth regression for multi-view stereo with correlation cost volume