Multimedia Systems (2023) 29:377–387
https://doi.org/10.1007/s00530-022-01009-2

REGULAR PAPER

Hierarchical MVSNet with cost volume separation and fusion based on U-shape feature extraction

Wanjun Liu[1] · Junkai Wang[1] · Haicheng Qu[1] · Lei Shen[1]

Received: 28 July 2021 / Accepted: 19 September 2022 / Published online: 27 September 2022

Abstract
Multi-view stereo (MVS) methods based on deep learning have developed rapidly in recent years, but inaccuracies in reconstruction due to the general effect of feature extraction and poor correlation between cost volumes are still present, opening possibilities for improvement in reconstruction accuracy and completeness. We therefore develop a hierarchical MVS network model with cost volume separation and fusion to mitigate these problems. First, to obtain a more complete and accurate feature information from the input images, a U-shape feature extraction module was designed that outputs feature information simultaneously according to a hierarchical structure composed of three different scales. Then, to enhance the learning ability of the network structure for features, we introduced attention mechanisms to the extracted features that focus on and learn the highlighted features. Finally, in the cost volume regularization stage, a cost volume separation and fusion module was designed in the structure of a hierarchical cascade. This module separates the information within the small-scale cost volume, passes it to the lower level cost volume for fusion, and performs a coarse-to-fine depth map estimation. This model results in substantial improvements in reconstruction accuracy and completeness. The results of extensive experiments on the DTU dataset show that our method performs better than Cascade-MVSNet by about 10.2% in accuracy error (acc.), 7.6% in completeness error (comp.), and 9.0% in overall error (overall), with similar performance in the reconstruction completeness, showing the validity of our module.

Keywords 3D reconstruction · Deep learning · Multi-view stereo · Cost volume · Separation and fusion

1 Introduction

Multi-view stereo (MVS) uses a set of images with known camera parameters to process a 3D model of the scene within the images. Conventional algorithms [1–6] usually use the projection relationships between multiple views to calculate the corresponding points and recover the scene information through a formula. Although these methods achieve good results in ideal Lambert scenes, satisfactory reconstructions are often not obtained in cases that include texture scarcity, texture repetition, or lighting variations. In recent years, the rapid development of deep learning techniques in the field of computer vision has stimulated interest in improving the reconstruction structure, resulting in various learning-based MVS methods [7–10]. These methods usually utilize convolutional neural networks (CNNs) to extract deep image features from the input images and combine global semantic information, such as specular reflection and reflection prior, to obtain robust feature matching. These methods not only improve the reconstruction speed but also perform better in reconstruction completeness compared to traditional methods.

The recent MVSNet [11] and subsequent works [12–16] map the depth features of all views into a cost volume, which is adjusted by a multi-scale 3D CNN to estimate the depth map of the reference image. However, as the image resolution increases, the cost volume regularization operation

Communicated by C. Yan.

✉ Junkai Wang
wangjunkailntu@163.com

Wanjun Liu
liuwanjun@lntu.edu.cn

Haicheng Qu
quhaicheng@lntu.edu.cn

Lei Shen
shenlei95821@163.com

[1] School of Software, Liaoning Technical University, Huludao 125105, People's Republic of China

causes the memory consumed to grow cubically. This problem is addressed in subsequent works via sequential regularization [12], which is achieved by reducing the depth dimension of the cost volume [15] or using sparse cost volumes [16]. Although the deep learning method can improve the quality of 3D reconstruction, further improvements in the completeness of feature extraction and the accuracy of depth map estimation are still achievable, indicating that 3D reconstruction technology based on deep learning is still in development.

In response to the above problems, we propose a hierarchical MVS network model for cost volume separation and fusion based on deep learning (hereafter referred to as HSF-MVSNet). The main contribution is summarized as follows:

(1) A U-shape feature extraction module is designed with reference to the network structure of the U-Net in the feature extraction stage. This module can extend the perceptual field to improve the accuracy of the depth map estimation results. Concurrently, the extracted feature map is divided into three scales for the output, forming a hierarchical structure with different scale sizes.
(2) A lightweight attention mechanism is introduced after the feature extract stage to the topmost layer containing the smallest scale. This mechanism focuses on important feature information and computes it for superior feature map information.
(3) A cost volume separation and fusion module for the cost volume regularization stage divides the cost volume generated by the previous layer into two; one layer continues to estimate the depth map, and the other is passed to the next layer in the fusion to generate a new cost volume. This process is iterated, with the new cost volume being split for depth map estimation and fusion to estimate the depth map of the original size.

The method was evaluated experimentally on the DTU dataset [17], with the evaluation results showing an approximate 21.0% improvement over MVSNet and 9.0% improvement over CascadeMVSNet. The method also outperformed most of the currently available MVS reconstruction methods.

2 Related work

2.1 Traditional MVS

Traditional MVS methods can usually be divided into four types: point cloud-based algorithms, variable polygon mesh based algorithms, stereo pixel based algorithms, and depth map based algorithms. Point cloud-based algorithms [4] start from a sparse set of matching key points and propagate them to low-texture regions for dense reconstruction by iteratively processing the 3D points. In algorithms based on variable polygon meshes [18, 19], an initial guess of the scene surface is required, followed by iterative operations to improve the multi-view photometric consistency and update the scene surface. Voxel-based algorithms [8, 9, 20] discretize the 3D space into a regular grid to find voxels near the scene surface, with the memory consumption of this method increasing with the resolution of the voxels. The depth map based algorithms [1, 3, 5, 6] decompose the reconstruction process into several single-view depth map estimation processes and then fuse all the depth maps in a point cloud under one coordinate system. This method can easily fuse the depth maps into a point cloud [21] or volume reconstruction [22], which is the optimal method for performing 3D reconstruction.

2.2 Learning-based MVS

Deep learning-based stereo matching methods [25, 26] have been widely used in recent times, achieving good results. To solve the MVS problem, Ji et al. [8] proposed a voxel-based approach inspired from these methods, which decomposed the entire space into smaller cubes and regresses their surfaces, improving the spatial resolution but also leading to a significant increase in time complexity. Huang et al. [10] took an arbitrary number of images of various poses as input and first generated a set of plane-sweep volumes and used the DeepMVS network to predict a high-quality depth map.

Yao et al. [11] proposed an end-to-end MVSNet that utilizes a differentiable homography for mapping multi-view image features, taking inspiration from the traditional planar scan method for constructing the reference image's matching cost. Meanwhile, MVSNet adopts a variance-based construction to match costs independently of the number of images. This achieves the depth estimation of the reference image through an hourglass-type 3D convolutional layer with a regression operation, with its performance surpassing that of traditional algorithms.

However, the number of depth samples requires MVSNet to occupy large amounts of computational resources. To address this problem, Yao et al. [12] avoided dense 3D CNNs using recurrent neural networks in the depth direction, effectively reducing the model's memory occupation. Chen et al. [13] proposed a direct point-based matching cost regularization method, which adopts a coarse-to-fine depth estimation strategy to generate a 3D point cloud directly based on the initial depth map. After constructing the matching cost for each 3D point, the method aggregates spatial neighborhood information for the 3D points via a K-neighborhood search before performing matching cost regularization for each point individually based on a multilayer perceptron. This method achieves better

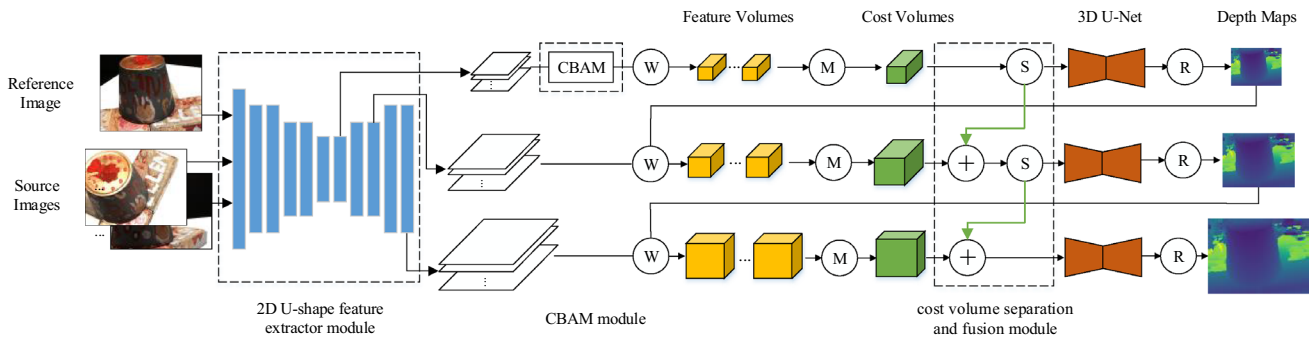


Fig. 1 Proposed network architecture for the HSF-MVSNet

accuracy and computational efficiency than the spatial cost matrix-based method for image depth estimation. Gu et al. [15] modified the MVSNet model to a cascading method by first predicting the depth map of the downsampled quarter, scaling down the range of depth assumptions at the current scale while continuing to predict the depth map of the downsampled half, and finally predicting the depth map of the original image size. Using this cascading method, larger and less frequent depth intervals can be used, thus allowing more data to be trained at a time. Overall, MVS contains the stereo geometric theoretical basis of stereo matching while effectively improving the impact of the occlusion problem via more image perspectives, and thus achieves improvements in both accuracy and generalization. However, the existing methods are still capable of improvements in the regularization of matching cost volumes, the optimization of depth estimation results, and algorithm efficiency.

3 Methods

The current deep learning-based MVS methods described above have average feature extraction performance and poor correlation between cost volumes. To improve on these, modifications were made in the feature extraction and cost volume regularization stage of a pre-existing [15]. The following sections discuss the main com-ponents of the method: Sect. 3.1 introduces the 2D U-shape feature extraction module; Sect. 3.2 introduces the lightweight attention mechanism (CBAM) [23] for feature refinement; Sect. 3.3 introduces the cost volume separation and fusion module that separates the cost volumes and passes them to the lower layer for fusion, improving results for cost volume regularization operations; and Sect. 3.4 describes the loss function of the model. Figure 1 shows the complete architecture of the network model.

3.1 Feature extraction

Similar to pre-existing work [11], the *N* input images are divided into two groups: one group has only one image, referred to as the reference image, and the other group contains the remaining *N-1* images, called the source images. In the feature extraction stage, most pre-existing works extract features using CNNs [11, 12] or multi-scale features [13, 15]. This is because multi-scale features can accomplish coarse-to-fine depth estimation while single features cannot, and the additional feature information can be retained for better depth estimation by gradually recovering the features from the original image size through small-scale features. To mitigate the poor closeness of association between feature maps at different scales, a U-Net inspired 2D U-shape feature extraction module based on multi-scale features was designed. Figure 2 details the specific structure.

The U-Net model was initially proposed in medical image processing. It uses a symmetric structure to extract deep features of the image by continuous convolution and pooling, but at the same time the size of the obtained feature map is decreasing compared to the original image, and the deeper the network is, the more information is lost. Therefore, for each down-sampling, an up-sampling is used to restore the original size, and the result of each up-sampling is connected to the feature map of the corresponding size in the previous period, so that the combination of high-level features and low-level features are achieved and the class of each pixel can be predicted more accurately. In the U-shape feature extraction module, for all input images with a resolution of 512×640, after convolution and ReLu, the original image of 3 channels is turned into 32 channels, and the feature maps of 1, 1/4 and 1/16 of the original image size are generated by two times of maximum pooling and constant convolution, respectively. In the upsampling stage, the feature information of different scales is stitched and fused in the channel dimension to form thicker features, which continue to convolve and upsample after the concat operation to finally

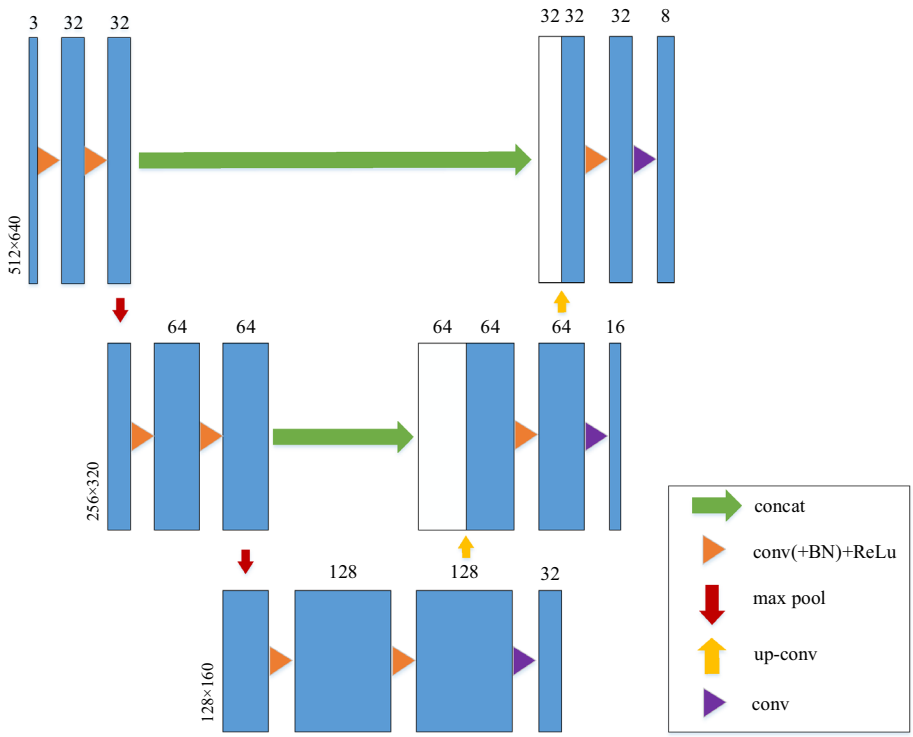


Fig. 2 Structure of the 2D U-shape feature extraction module

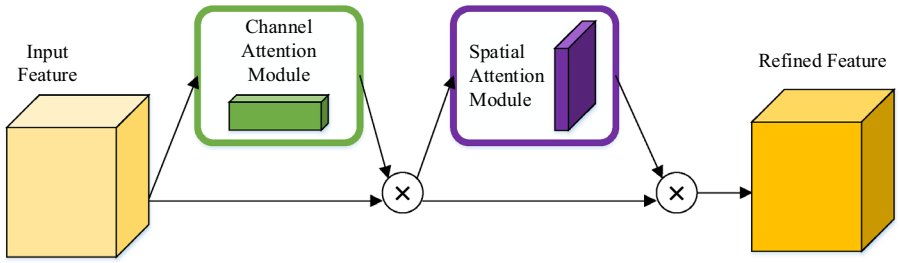


Fig. 3 CBAM structure

obtain three groups of feature maps of different sizes, corresponding to the number of channels 8, 16 and 32 separately in the order of feature maps from largest to smallest. Such a U-shaped feature extraction module can retain richer detail features. Downsampling can reduce the spatial dimension of the pooling layer, upsampling can repair the detail and spatial dimension of the object, while the concat operation can better repair the detail of the target, making the subsequent depth estimation results more accurate and complete.

3.2 Convolutional block attention module (CBAM)

In recent years, the CBAM [23] has played an important role in target detection, leading to improvements in detection accuracy. The module focuses on the important features while suppressing insignificant features, with the extracted features having the potential to be used in a feature enhancement role after secondary processing. Due to its flexibility and efficiency, we introduced a CBAM to further process the top-level feature maps extracted by the 2D U-shape module. This is because the topmost layer requires the most image scaling as it uses the smallest scale, and this increased scaling can result in a greater detail loss in the feature map, making it necessary to utilize the CBAM. Adding the CBAM will also decrease the time spent compared to performing the attention mechanism process in every layer.

The CBAM contains two submodules: the channel attention module (CAM) and the spatial attention module (SAM). For a given input feature map, the CBAM module computes the attention map sequentially from these two independent submodules and then multiplies the attention map with the feature map to output an adaptive and optimized feature map, whose structure is shown in Fig. 3. The process performs the following two main operations:

$$F_C = M_C(F) \otimes F, \tag{1}$$

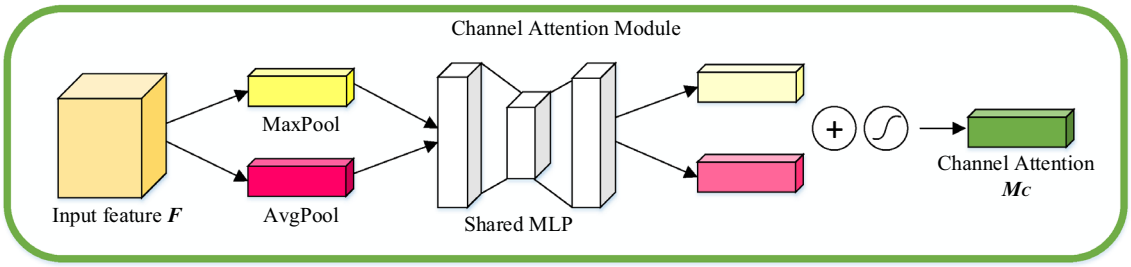


Fig. 4 The process of the CAM

$$F_S = M_S(F_C) \otimes F_C, \tag{2}$$

where F denotes the input features, F_C and F_S denote the features obtained in the channel and spatial dimensions, respectively, $\otimes$ denotes the dot product operation, and M_C and M_S denote the weight coefficients obtained by feature extraction in the channel and spatial dimensions, respectively. As shown in Fig. 4, the extracted features are inputted into the CAM. The feature map F, of dimension $H \times W \times C$, is first compressed in the spatial dimension. Following this, two $1 \times 1 \times C$ one-dimensional vectors are obtained by global maximum pooling and global average pooling, which are fed into a shared two-layer neural network. The output features are then summed element-by-element and combined using a sigmoid activation function to obtain the weight coefficient M_C. Finally, the new feature F_C is obtained by multiplying the weight coefficient with the original feature F. The calculation can be expressed by Eq. 3:

$$\begin{aligned} M_C(F) &= \sigma\left(\text{MLP}\left(\text{Avg Pool}\left(F\right)\right) + \text{MLP}\left(\text{Max Pool}\left(F\right)\right)\right) \\ &= \sigma\left(W_1(W_0(F^C_{avg})) + W_1(W_0(F^C_{max}))\right) \end{aligned} \tag{3}$$

where σ is the sigmoid operation, F^C_{avg} and F^C_{max} represent the features after global average and global maximum pooling computations, respectively, W_0 and W_1 represent the parameters of the two-layer neural network, and MLP stands for Multi-Layer Perceptron.

Similar to the CAM, the SAM (shown in Fig. 5) is first given the $H \times W \times C$ feature F_C, from which two $H \times W \times 1$ two-dimensional features are obtained via channel-based global maximum and global average pooling. The two results are combined based on channels, convolved using a hidden layer containing a convolution kernel to reduce the number of channels to 1, and then inputted into a sigmoid activation function to obtain the weight coefficient M_S. Finally, the new feature F_S is obtained by multiplying the weight coefficient with the original feature F_C. The module's entire operation can be represented by Eq. 4:

$$\begin{aligned} M_S(F_C) &= \sigma\left(f^{7\times7}([\text{Avg Pool}\left(F_C\right); \text{Max Pool}\left(F_C\right)])\right) \\ &= \sigma\left(f^{7\times7}[F^S_{C\,avg}; F^S_{C\,max}]\right) \end{aligned} \tag{4}$$

where σ is the sigmoid operation, 7×7 is the size of the convolution kernel, $F^S_{C\,avg}$ and $F^S_{C\,max}$ denote the features after global average and global maximum pooling calculations, respectively.

3.3 Cost volume separation and fusion module

In the cost volume regularization stage, there is a lack of corroboration between the cost volumes generated at each scale, relying only on the 2D U-shape module's up-sampling to maintain the only connection; this results in the information of the cost volumes in each layer not being passed on. Inspired by [24], the increased inter-correlation between cost volumes makes the depth estimation results more accurate. So, we designed a cost volume separation fusion module based on a multi-scale hierarchical structure from coarse-to-fine to separate and fuse the cost volumes generated in each layer into the next layer, with the small-scale cost volume information fused into the cost volume of the next layer to improve the estimation quality of the depth map. The specific operation is divided into two steps: separation and fusion.

As shown in Fig. 6, the input cost volume of this layer is divided into two by the separator, with one of the cost volumes entering the 3D U-Net module for regularization and to estimate its depth map. The second cost volume is

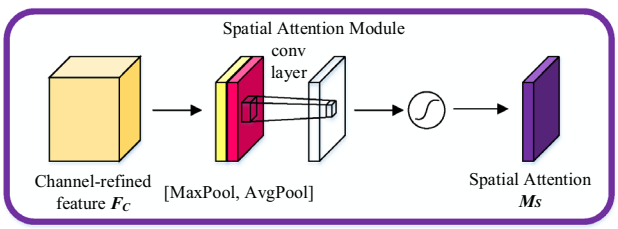


Fig. 5 The process of the SAM

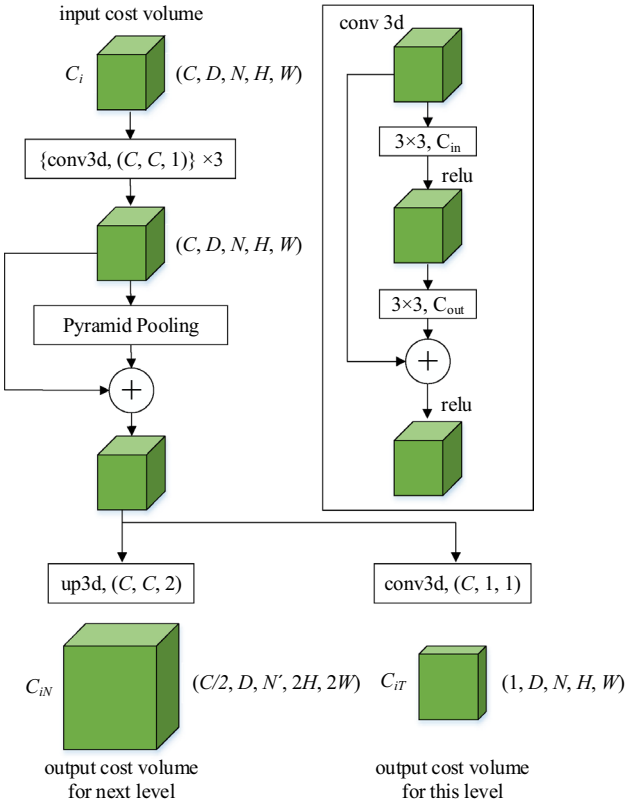


Fig. 6 Structure of the cost volume separator

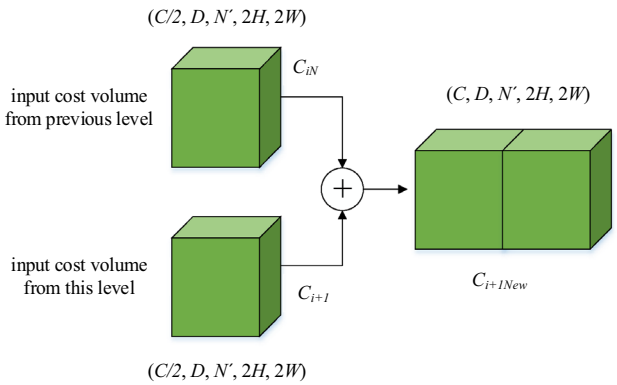


Fig. 7 Structure of the cost volume fuser

passed into the next layer to be fused with the cost volume generated in the next layer, as shown in Fig. 7. This new cost volume is then also divided into two, and this process is iterated to estimate the depth map of the original image size. At the top layer, the generated $1\times32\times48\times128\times160$ cost volume C_1 is used in the separation operation to generate the $1\times1\times48\times128\times160$ cost volume C_{1T} and $1\times16\times32\times256\times320$ cost volume C_{1N}. C_{1T} continues into the 3D U-Net module for processing and estimating the layer's depth map. Meanwhile, C_{1N} is passed to the middle layer and fused with the $1\times16\times32\times256\times320$ cost volume C_2, which is generated by the middle layer, to create the $16\times1\times256\times320$ cost volume C_{2New}. Continuing this process, the fused cost volume C_{3New} can then be fed into the 3D U-Net module and used to estimate the final depth map.

3.4 Loss functions

In the three-level cascade model, we monitor losses on every level, and the total losses are defined as follows:

$$Loss_{\text{total}} = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3, \tag{5}$$

where λ_1, λ_2, and λ_3 are the loss weights of the three stages, which are the same as CasMVSNet [15], are set as 0.5, 1 and 2, respectively; L_1, L_2 and L_3 are the losses at the small, medium, and large scales, respectively. We define L_i as *SmoothL1Loss*, as follows:

$$L_{\text{i}} = Loss(x, y) = \frac{1}{n}\sum_i z_{\text{i}}, \tag{6}$$

$$z_i = \begin{cases} 0.5(x_{\text{i}} - y_{\text{i}})^2, & |x_{\text{i}} - y_{\text{i}}| < 1 \\ |x_{\text{i}} - y_{\text{i}}| - 0.5, & \text{otherwise} \end{cases}, \tag{7}$$

where n is the number of images, and x_{i} and y_{i} are the estimated depth and true depth, respectively.

4 Experiments

4.1 Dataset

The DTU dataset[1] [17] was used for the experiments. The DTU dataset is an indoor dataset specially shot and processed for MVS, using an industrial robot arm equipped with adjustable lights to shoot multiple views of an object. The viewpoint of each object is strictly controlled, with the internal and external parameters of the camera directly obtainable for each view. The DTU dataset consists of 124 different objects or scenes, with 49 views taken for each object and 7 different brightness levels for each view. There are 343 images inside each object or scene folder, each with a resolution of 1600×1200.

[1] http://roboimagedata.compute.dtu.dk/.

Table 1 Environment configuration of the experiments

Operating system	Ubuntu 18.04.5
CPU	Intel Core i9-10920X
GPU	GeForce RTX 3090
RAM	64GB
Python version	3.7.0
CUDA version	11.0
Pytorch version	1.7.1

4.2 Training

The model was trained using the DTU training set, with an initial input image resolution size of 512×640 and $N = 3$ input images. The three layers were divided into three stages with depth assumptions of 48, 32, and 8, which corresponded to 4, 2, and 1 times the depth intervals of MVSNet [11], respectively. The feature map resolutions at the three scales were 1/16, 1/4, and 1 times the original image's resolution size. In the training, the Adam optimizer was used, with the parameters β_1=0.9 and $\beta_2 = 0.999$ [15]. The initial learning rate of the training was set to 0.001, and 16 epochs were performed, with the learning rate being halved after the 10, 12, and 14th epochs. The method was trained on a single NVIDIA 3090 graphics card with the batch size set to 1. Table 1 shows the specific environment configuration.

4.3 Evaluation and comparison experiments

For the DTU dataset, the accuracy and completeness of the generated files were experimentally evaluated on the MATLAB code provided with the DTU dataset. There are three main evaluation metrics: accuracy error (Acc.), completeness error (Comp.), and overall error (Overall). The accuracy error is calculated as the average distance between the reconstructed and real point cloud, which reflects the accuracy of the reconstructed point cloud in terms of spatial location. The completeness error is calculated as the average distance between the real and reconstructed point cloud, which reflects the completeness of the reconstructed point cloud. Finally, the overall error was calculated as the average of the accuracy error and the completeness error.

Table 2 shows the evaluation results of the DTU dataset after adjusting the model for optimization. It can be seen that the designed model achieved good performance in all three error types. Figure 8 compares reconstructed models of scenes Scan1 and Scan24, with the red frames indicating HSF-MVSNet's improved results over pre-existing methods.

Table 2 Comparison of HSF-MVSNet and other methods for evaluating metrics

Methods	Acc. (mm)	Comp. (mm)	Overall (mm)
Camp [3]	0.835	0.554	0.695
Furu [4]	0.613	0.941	0.777
Tola [6]	0.342	1.190	0.766
Gipuma [5]	**0.283**	0.873	0.578
SurfaceNet [8]	0.450	1.040	0.745
MVSNet [11]	0.396	0.527	0.462
R-MVSNet [12]	0.383	0.452	0.417
P-MVSNet [14]	0.406	0.434	0.420
Point-MVSNet [13]	0.342	0.411	0.376
MVSCRF [27]	0.371	0.426	0.398
Fast-MVSNet [28]	0.336	0.403	0.370
CVP-MVSNet [29]	0.296	0.406	**0.351**
CIDER [30]	0.417	0.437	0.427
D2HC-RMVSNet [31]	0.395	0.378	0.386
CascadeMVSNet [15]	0.421	0.382	0.401
HSF-MVSNet	0.378	**0.353**	0.365

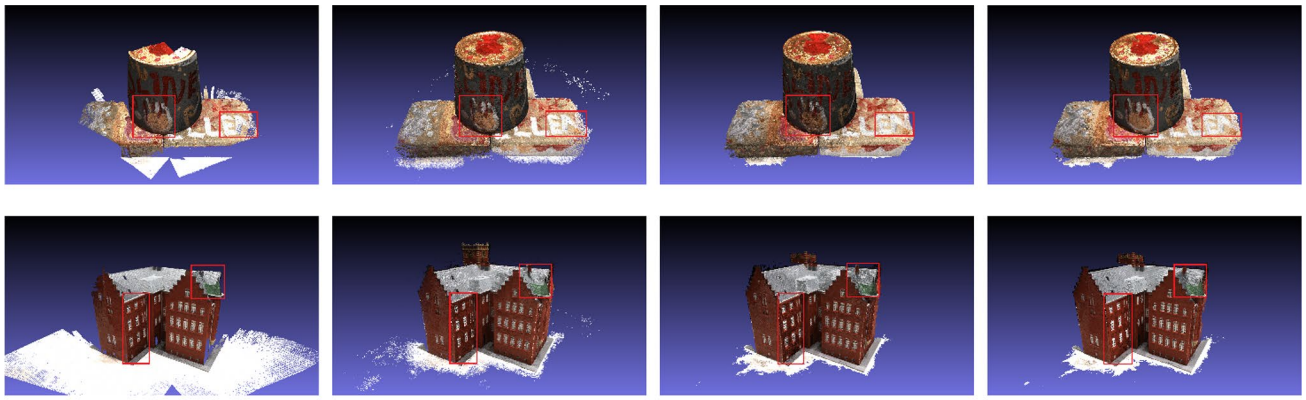
Bold values indicate the best result of each column

Table 3 shows the comparison of our method with the Cascade-MVSNet method in terms of secondary metrics. It can also be seen that our method is to be due to the original method.

Figure 9 Displays the reconstruction results of other scenes.

4.4 Ablations

Table 4 shows the results of the individual tests for the three improvements proposed in this paper. The first row shows the metrics of the initial model. The second, third, and fourth rows are the evaluation metrics for adding the U-shape feature extraction module, CBAM module, and cost volume separation and fusion module to the original model, respectively. The fifth row shows the evaluation metrics for the overall HSF-MVSNet. All three modules resulted in improved metrics, and the whole HSF-MVSNet structure improved the accuracy of the reconstruction. Figure 10 presents a comparison of the depth maps estimated for each experiment, with the red frames indicating that each of the



(a) ground truth (b) MVSNet (c) CascadeMVSNet (d) HSF-MVSNet

Fig. 8 Comparison of the reconstructed models of scan1 and scan24

Table 3 Comparison of HSF-MVSNet and CascadeMVSNet for Secondary Evaluating Metrics

Methods	Absolute depth error	2 mm absolute error	4 mm absolue error	2 mm accuracy	4 mm accuracy
CascadeMVSNet [15]	9.152	0.733	2.725	75.05%	84.50%
HSF-MVSNet	8.658	0.592	2.740	79.03%	86.49%

modules led to improvements over the original method; the 3D model estimated from the depth maps also improved.

In this paper, we focus on improving the accuracy metrics of the reconstruction to obtain higher accuracy at the cost of memory consumption and running time. The experiments in this paper are shown in Table 5 for the comparison of their corresponding memory consumption and occupied time after each improvement is made.

The proposed method in this paper aims to extract a more accurate and complete feature map when improving the U-shape feature extraction network. This network iteratively processes the extracted features, which adds a large amount of computation and, at the same time, requires caching a large amount of intermediate data. In the attention mechanism processing, it also requires a large number of additional convolutions. Similarly, in the pre-processing stage of cost volume regularization, the multi-scale cost volume information fusion network also needs to cache a large number of small-scale cost volume computation results for fusion with large-scale customers. The corresponding memory consumption and running time of the experiments in this paper increase after each improvement is made, and therefore, how to effectively reduce the memory consumption and running time while maintaining the same accuracy also becomes the main goal of the subsequent work.

5 Conclusion

In response to problems arising from the general effect of feature extraction, and poor correlation between cost volumes that appear in deep learning-based MVS matching methods, we propose a new MVS network model HSF-MVSNet. This method introduces improvements to the feature extraction, feature enhancement, and cost volume regularization stages. To improve the completeness and accuracy of the features, the following modifications were made: a U-shape network structure for feature extraction, a CBAM module for feature enhancement of the extracted features, and a cost volume separation and fusion module designed to improve the estimation quality of the depth map by increasing the interconnection between different layers of cost volume so that information can be passed layer by layer. The method achieves better performance on the DTU dataset than pre-existing methods, validating its effectiveness. However, due to the increased convolution size, the computation between cost volumes is also increased. As reconstruction accuracy increases, the GPU's memory consumption and the training time also increase. So in the future, we will further adjust the network structure to reduce the memory consumption and shorten the training time.

Fig. 9 Reconstructed models of HSF-MVSNet in other scenes

Table 4 Comparison of evaluation metrics for ablations

Methods	Acc. (mm)	Comp. (mm)	Overall (mm)	Absolute depth error	2 mm absolute error	4 mm absolute error	2 mm accuracy (%)	4 mm accuracy (%)
Cascade MVSNet [15]	0.421	0.382	0.401	9.152	0.733	**2.725**	75.05	84.50
CasMVSNet + U-shape	0.397	**0.349**	0.373	8.661	0.676	2.737	77.30	85.12
CasMVSNet + CBAM	0.378	0.364	0.371	9.337	0.665	2.750	77.02	85.15
CasMVSNet + Sep&Fus	0.405	0.378	0.392	8.943	0.659	2.731	76.32	84.80
HSF-MVSNet	**0.378**	0.353	**0.365**	**8.658**	**0.592**	2.740	**79.03**	**86.49**

Bold values indicate the best result of each column

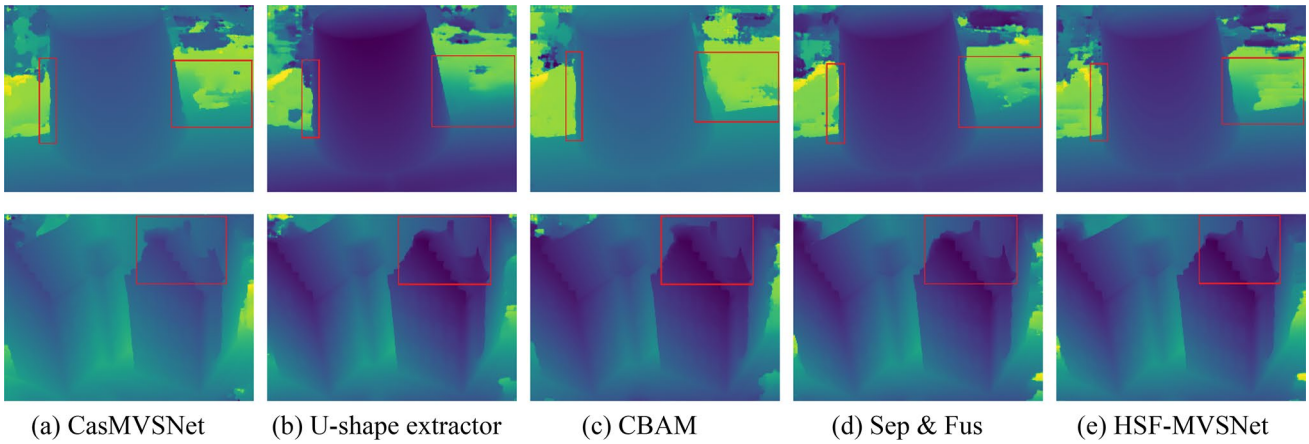


Fig. 10 Comparison of depth estimation maps for ablations

Table 5 Comparison of memory consumption and occupied time for various improvement sections

Methods	Memory consumption (GB)	Time occupied (seconds)
MVSNet [11]	7.1	0.4
CascadeMVSNet [15]	6.2	0.8
CasMVSNet + U-shape	9.7	1.0
CasMVSNet + CBAM	7.6	0.8
CasMVSNet + Sep&Fus	14.8	2.6
HSF-MVSNet	21.2	3.1

Acknowledgements The DTU datasets used in this paper are from the addresses provided by MVSNet and CascadeMVSNet, and we also thank Y. Yao and X. Gu for sharing their contributions.

Funding This work was funded by National Natural Science Foundation of China (Grant Number: 42071351) and Liaoning Natural Fund General Project (Grant Number: LJ2019JL010).

Declarations

Conflict of interest The authors declare they have no conflict of interest.

References

1. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer vision–ECCV 2016. Lecture notes in computer science, vol. 9907. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31
2. Schönberger, J. L., Frahm, J.: Structure-from-Motion Revisited. 2016 IEEE conference on computer vis-ion and pattern recognition (CVPR), pp. 4104–4113, (2016). https://doi.org/10.1109/CVPR.2016.445
3. Campbell, N.D.F., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) Computer vision–ECCV 2008 lecture notes in computer science, vol. 5302. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_58
4. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010). https://doi.org/10.1109/TPAMI.2009.161
5. Galliani, S., Lasinger, K., Schindler, K.: Massi-vely parallel multiview stereopsis by surface norm-al diffusion. 2015 IEEE International Confer-ence on Computer Vision (ICCV), pp. 873–881. (2015). https://doi.org/10.1109/ICCV.2015.106
6. Tola, E., Strecha, C., Fua, P.: Efficient large-scale multi-view stereo for ultra high-resolution image sets. Mach. Vis. Appl. **23**, 903–920 (2012). https://doi.org/10.1007/s00138-011-0346-8
7. Hartmann, W., Galliani, S., Havlena, M., Van Gool, L., Schindler, K.: Learned Multi-patch Similarity. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1595–1603, (2017) https://doi.org/10.1109/ICCV.2017.176
8. Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: SurfaceNet: an end-to-end 3d neural network for multiview stereopsis. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2326–2334, (2017). https://doi.org/10.1109/ICCV.2017.253
9. Kar, A., Hane, C., Malik, J.: Learning a multi-view stereo machine. Adv Neural Inf Process Syst (NIPS) (2017).https://arxiv.org/pdf/1708.05375.pdf
10. Huang, P. H., Matzen, K., Kopf, J., Ahuja, N., Hu-ang, J. B.: DeepMVS: learning multi-view stereo-psis. In Proceedings-2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition). IEEE Computer Society. (2018). pp. 2821–2830. https://doi.org/10.1109/CVPR.2018.00298
11. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: MVSNet: depth inference for unstructured multi-view Stereo. ArXiv, abs/1804.02505. (2018). https://doi.org/10.1007/978-3-030-01237-3_47.
12. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent MVSNet for high-resolution multi-view stereo depth inference. 2019 IEEE/CVF Conference on Computer Vision and Pattern

Recognition (CVPR), pp. 5520–5529, (2019). https://doi.org/10.1109/CVPR.2019.00567.
13. Chen, R., Han, S., Xu, J., Su, H.: Point-Based Multi-View Stereo Network. 2019 IEEE/CVF Inter-national Conference on Computer Vision (ICCV), pp. 1538–1547. (2019). https://doi.org/10.1109/ICCV.2019.00162
14. Luo, K., Guan, T., Ju, L., Huang, H, Luo, Y.: P-MVSNet: learning patch-wise matching confi-dence aggregation for multi-view stereo. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10451–10460, (2019). https://doi.org/10.1109/ICCV.2019.01055
15. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. 2020 IEEE/CVF Conference on Computer Vision and Pattern Reco-gnition (CVPR), pp. 2492–2501. (2020). https://doi.org/10.1109/CVPR42600.2020.00257
16. Yu, Z., Gao, S.: Fast-MVSNet: sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. 2020 IEEE/CVF Confer-ence on Computer Vision and Pattern Recognition (CVPR), pp. 1946–1955. (2020). https://doi.org/10.1109/cvpr42600.2020.00202
17. Aanæs, H., Jensen, R.R., Vogiatzis, G., et al.: Large-scale data for multiple-view stereopsis. Int. J. Comput. Vis. **120**, 153–168 (2016). https://doi.org/10.1007/s11263-016-0902-9
18. Furukawa, Y., Ponce, J.: Carved visual hulls for image-based modeling. Int. J. Comput. Vision **81**(1), 53–67 (2009). https://doi.org/10.1007/s11263-008-0134-8
19. Li, Z., Wang, K., Zuo, W., Meng, D., Zhang, L.: Detail-preserving and content-aware variational multi-view stereo reconstruction. IEEE Trans Image Process **25**(2), 864–877 (2016). https://doi.org/10.1109/TIP.2015.2507400
20. Seitz, S.M., Dyer, C.R.: Photorealistic scene recons-truction by voxel coloring. Int J Computer Vision (IJCV) (1999). https://doi.org/10.1109/CVPR.1997.609462
21. Merrell, P.C., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J., Yang, R., Nistér, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. 2007 IEEE 11th International Conference on Computer Vision, 1–8. (2007). https://doi.org/10.1109/ICCV.2007.4408984
22. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: real-time dense surface mapping and tracking. 2011 10th IEEE International Symposium on Mixed and Augmented Reality, 127–136. (2011). https://doi.org/10.1109/ISMAR.2011.6092378
23. Woo, S., Park, J., Lee, J., Kweon, I.: CBAM: convolutional block attention module. (ECCV) (2018). https://doi.org/10.1007/978-3-030-01234-2_1.
24. Yang, G., Manela, J., Happold, M., Ramanan, D.: Hierarchical deep stereo matching on high-resolution images. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5510–5519, (2019). https://doi.org/10.1109/CVPR.2019.00566
25. Luo, W., Schwing, A. G., Urtasun, R.: Efficient deep learning for stereo matching. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5695–5703, (2016). https://doi.org/10.1109/CVPR.2016.614
26. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. J Mach Learn Res **17**(65), 1–65 (2016)
27. Xue Y, Chen J, Wan W, et al.: MVSCRF: learning multi-view stereo with conditional random fields[C]. International Conference on Computer Vision, 2019: 4311–4320
28. Yu Z, Gao S.: Fast-MVSNet: sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement[C]. Conference on Computer Vision and Pattern Recognition, 2020: 1946–1955
29. Yang J, Mao W, Álvarez J, et al.: cost volume pyramid based depth inference for multi-view stereo[C]. Conference on Computer Vision and Pattern Recognition, 2020: 4876–4885
30. Xu Q, Tao W.: Learning inverse depth regression for multi-view stereo with correlation cost volume[C]. AAAI, 2020: 12508–12515
31. Yan J, Wei Z, Yi H, et al.: Dense hybrid recurrent multi-view stereo net with dynamic consistency checking[C]. ECCV, 2020: 674–689