IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 May 2023, accepted 21 May 2023, date of publication 29 May 2023, date of current version 6 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280929

METHODS

Multi-View Stereo Network With Gaussian Distribution Iteration

XIAOHAN ZHANG AND SHIKUN LI
School of Mathematics and Statistics, Shandong University, Weihai 264209, China
Corresponding author: Xiaohan Zhang (202017778@mail.sdu.edu.cn)

ABSTRACT Multi-view stereo estimates the depth maps of multiple perspective images in a scene and then fuses them to generate a 3D point cloud of the scene, which is an essential technology of 3D reconstruction. In this paper, we propose a deep learning method GDINet, applying probabilistic methods to the pyramid framework, which can significantly improve reconstruction quality. In detail, we first establish a Gaussian distribution for each image's pixel and iterate it in the pyramid framework. The mean value is the estimated depth, and the variance represents the depth estimation error. In addition, we design a novel loss function with excellent convergence to train our network. Finally, we present an initialization module to generate the coarse Gaussian distribution, controlling the parameters in a reasonable range. Our results rank 2*nd* on both DTU and Tanks & Temples datasets, showing that our network has high accuracy, completeness, and robustness. We also make a visualization comparison on the BlendedMVS dataset (containing many aerial scene images) to demonstrate the generalization ability of our model.

INDEX TERMS Multi-view stereo, 3D reconstruction, deep learning, Gaussian distribution.

I. INTRODUCTION

Multi-view stereo (MVS) calculates the depth information of images through overlapping views and restores the 3D architecture, which is a meaningful branch of 3D reconstruction [1], [2], [3], [4], [5]. As one of the critical technologies of environment perception, it has been widely used in various applications, such as remote sensing [6], augmented reality [7] and image-based rendering [8], [9], [10], [11]. Furthermore, MVS is a significant methodology in many areas of computer vision and photogrammetry [12], [13], [14], [15], [16].

The early traditional MVS methods [17], [18], [19], [20] use the matching pixels of adjacent photos to construct the energy function calculating the depth information. However, the depth estimation is inaccurate in some reflective and texture-less areas [21], [22], [23] due to the large difference of matching pixels. So many hand-crafted operators are needed to alleviate these areas' influence, which takes much time.

Compared to traditional methods, deep learning methods [24], [25], [26], [27] make it better to reconstruct scenes.

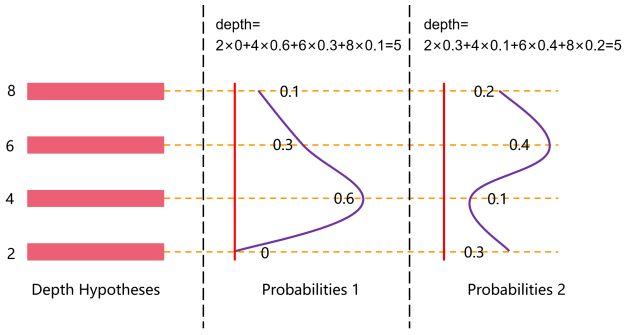


FIGURE 1. Illustration of regression method. The depth hypotheses are some possible depth hypotheses of one pixel; the probabilities are generated by networks. Regressing depth hypotheses and corresponding probabilities to calculate the final estimated depth. Many probabilities can calculate the same result, which may result in error depth estimation.

The deep CNNs [28] extract multi-channel features from images, which can reduce the impact of environmental factors such as light intensity, thus improving the reconstruction effect. The first end-to-end multi-view stereo network is MVSNet [16]. And the subsequent learning-based MVS approaches [29], [30], [31], [32], [33], [34], [35], [36], [37]

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

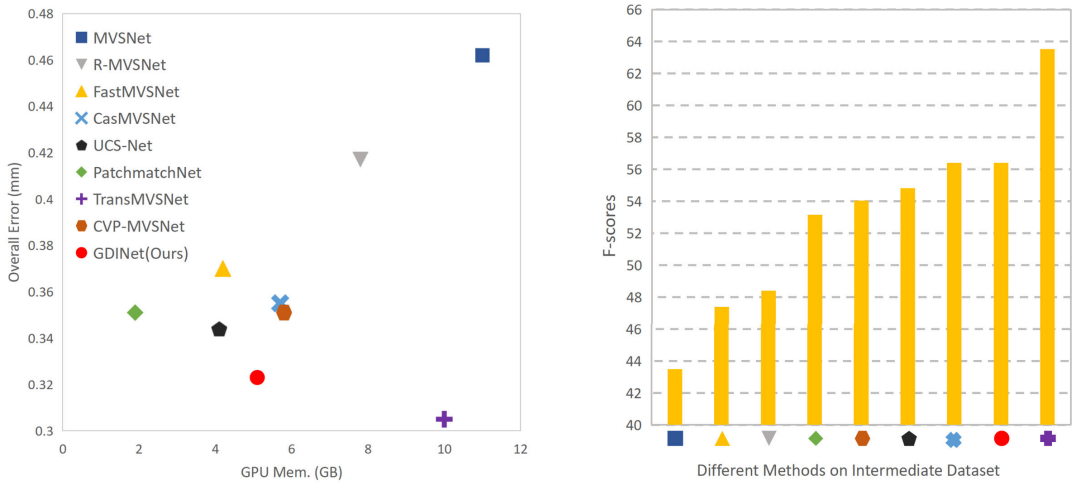


FIGURE 2. Comparison with other deep learning methods on the DTU and tanks & temples dataset. The left figure compares different methods on the DTU dataset (lower is better). The right figure is the result generated through tanks & temples dataset (higher is better). Our model ranks 2*nd* on both datasets. However, the TransMVSNet ranking first consumes twice as much memory as we do.

emulate MVSNet. They all use the regression architecture. These networks select a series of possible depth hypotheses for each pixel and train the corresponding probabilities through multi-view images. Then regressing the depth hypotheses and probabilities to calculate each pixel's depth. However, this regression method may lead to incorrect convergence of neural network parameters because different probabilities may get the same depth map, as shown in Figure 1, which results in bad reconstruction in some areas. Moreover, these methods all take the probabilities sum over four depth hypotheses nearest the estimated depth to calculate the confidence map for filtering out pixels with high depth estimation errors. However, the simple filtering method is more vulnerable to environmental impact.

CVP-MVSNet [32] proposes a pyramid framework to promote the accuracy and completeness of reconstruction. This method designs a pyramid structure, which takes the previous layer's depth map as the next layer's initial depth map to conduct depth sampling. This coarse-to-fine architecture is widely used in MVS [33], [34], [37].

Some novel probabilistic methods [38], [39] can significantly improve the reconstruction effect. They assume that each pixel's depth hypotheses follow a Gaussian distribution. The probabilistic methods propose a method for adaptive sampling depth hypotheses on a Gaussian distribution, which only requires a few depth hypotheses to achieve good reconstruction results, greatly reducing memory and computational time. The other advantage is that this method limits the shape of depth hypotheses' probabilities, dramatically reducing the non-robustness of the regression method. LANet [38] builds a Gaussian distribution of depth hypotheses through the ground truth depth and trains the probabilities converging to the distribution. However, many hyperparameters in the loss function affect the reconstruction performance. MaGNet [39] is a depth estimation network, but it provides noval methods for estimating distribution parameters. This article establishes a Gaussian distribution for each pixel and designs an NLL loss function without super parameters to train the network. However, when it is greater than the extreme point of standard deviation, the loss function is too flat, which makes the network difficult to converge.

To sum up, there are three ways can enhance the reconstruction results. First, the regression method, which could lead to error probabilities, should be modified. Second, the pyramid framework significantly improving reconstruction results should be added. Third, the probabilistic method can increase the reconstruction robustness, but the loss function should be revised.

To this end, we first create a Gaussian distribution for each pixel. The mean value is the estimated depth, and the variance is used to filter points with excessive error. We use it iterating in a pyramid framework to solve the problem of regression methods. In addition, We propose a novel loss function, called Gaussian loss, to estimate the parameters of the Gaussian distribution. Finally, we design an initialization module to limit the coarse distribution's parameters in a reasonable range. The comparison between our network and others is shown in Figure 2. It is worth noting that MaGNet [39] did not generate point clouds from depth maps through filtering and fusion algorithms, so this method is not added to the comparison on the point cloud datasets.

In summary, our main contributions are listed as follows:

(1) We propose a novel probabilistic pyramid architecture for reconstruction. We give each pixel a Gaussian distribution. The mean value is the estimated depth instead of obtaining it from the regression method and the variance is used to calculate confidence map for filtering, significantly improving the robustness and effect of reconstruction.

(2) We introduce a novel loss function named Gaussian loss. The loss function has an excellent convergence property, enhancing pixel depth estimation accuracy.

(3) We design a Gaussian distribution initialization module, controlling the parameters within an appropriate range to ensure the accuracy and robustness of the reconstruction.

II. RELATED WORKS

A. TRADITIONAL MVS METHODS

MVS methods mainly include direct point cloud methods [1], [17], volumetric methods [40], [41] and depth map methods [13], [19], [20], [42], [43]. The point cloud methods directly process point clouds, consuming much time. The volumetric methods divide the space into voxels which take up much memory. The depth map methods calculate the pixels' depth of each image and then project all the depth maps into space through filtering and fusion algorithms [44] to generate point clouds. This method is carried out in 2D space. Compared with the above two models in 3D dimension, it can save a lot of time and memory. Patchmatch [45] and SGM [46], [47] are basic algorithms in multi-view stereo algorithms. Patchmatch initializes a depth and normal vector for each pixel and optimizes them through the propagation algorithm. SGM establishes the energy function for the matched pixels in the image pair and obtains the image depth map after optimization. These works significantly improve the accuracy and speed of 3D reconstruction. However, for texture-less regions and highly reflective areas, the depth estimation of traditional methods may cause a high error, which needs some hand-crafted features.

B. LEARNING-BASED MVS METHODS

Compared with traditional methods, deep learning methods are more accurate and robust. The first deep learning method to fully implement end-to-end MVS is MVSNet [16]. It generates cost volume through differentiable homography and regulars the volume in 3D CNN to infer a probability volume. The depth map is calculated by regressing the probability volume and corresponding depth hypotheses. This paper also gives the fusion and filtering algorithms for projecting depth maps to space. However, the regression method is not robust enough, and the large 3D CNN consumes a lot of memory. R-MVSNet [48] uses 2D CNN to get a probability for each depth hypothesis and uses a classification loss function to train the model. This method saves much memory compared with MVSNet. Nevertheless, the receptive field of 2D CNN is much smaller than that of 3D CNN, affecting depth estimation accuracy. Based on these works, CVP-MVSNet [32] proposes a pyramid framework for MVS. This architecture iterates the depth map and optimizes it coarse-to-fine, dramatically improving the reconstruction effect. However, this method also uses the regression method to calculate the depth map. CasMVSNet [33] achieves the training with low-resolution images, which significantly improves training speed. However, this method increases the overall error in depth estimation. PatchmatchNet [34] uses adaptive neighborhoods to improve the reconstruction effect and reduce memory. However, this method is susceptible to the environment.

C. PROBABILISTIC METHODS

Introducing probabilistic methods into deep learning methods can significantly improve reconstruction performance. LANet [38] trains the probabilities of hypotheses converging to a Gaussian distribution, enhancing the robustness of reconstruction. However, too many hyperparameters in the loss function.

MaGNet [39] gives each pixel a Gaussian distribution. This work samples depth hypotheses adaptively using the 3σ principle of Gaussian distribution. The advantage is that only selecting a few depth hypotheses can achieve a good reconstruction effect, significantly reducing memory. This method also designs the NLL loss function without hyperparameters to train the network. However, this loss function is too flat when the standard deviation exceeds the extreme point, harming the convergence. Furthermore, the initial parameters are learned through a 2D CNN, which may lead to unreasonable parameters.

Based on CVP-MVSNet, giving each pixel a Gaussian distribution, modifying the NLL loss, and designing an initialization module, we achieve an excellent reconstruction effect by iterating the Gaussian map.

III. METHOD

This section describes the details of GDINet. Based on CVP-MVSNet [32], we use the iteration of the distribution map to enhance the reconstruction effect. Our network is illustrated in Figure 3.

We aim to estimate each reference image's depth map utilizing the multi-view stereo algorithm. Inputs of the GDINet are the reference image $\boldsymbol{I}_0$ and N source images $\{\boldsymbol{I}_i\}_{i=1}^{N}$ with the known camera parameters. After the multi-scale feature extraction, the images are converted into feature maps of different sizes.

Beginning at the coarsest stage, we design an initialization module to generate an initial Gaussian distribution for each pixel. The mean value represents each pixel's estimated depth, and the variance reflects the reconstruction quality and is used for filtering. This module's advantage is limiting the mean and the variance in a reasonable range. After this is a pyramid framework divided into four modules. First, the sampling and warping module warps the source feature maps to the reference view and calculates the similarity with the reference feature map to generate N similarity volumes. Second, the cost aggregation module generates cost volume by weighted aggregating those similarity volumes. Third, the regularization module calculates the probability volume through a regular 3D CNN module. Finally, the optimization module estimates the residuals of distribution's parameters to refine the initial Gaussian map. The refined Gaussian map is upsampled as the initial value of the next stage, using the

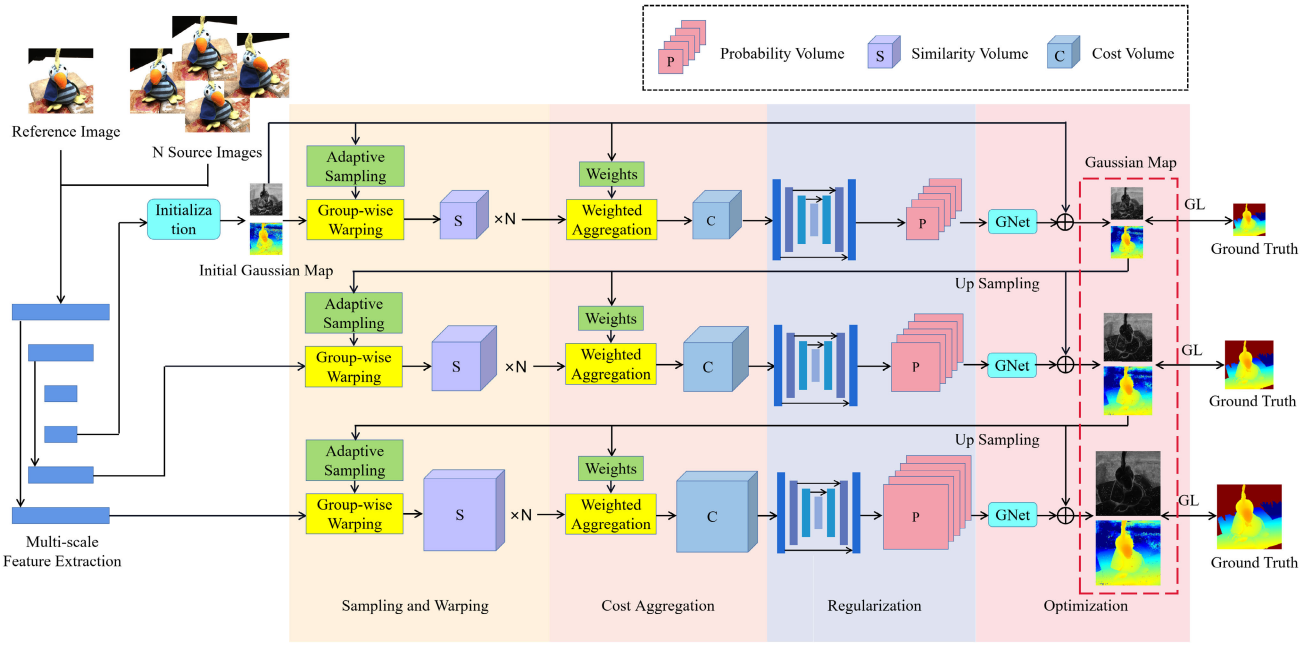


FIGURE 3. Illustration of GDINet. The iteration starts at the coarsest stage. For each different perspective image, the shared 2D CNN first extracts multi-scale features. Then, the initialization module estimates an initial Gaussian map for the reference view. Through adaptive sampling, we sample D ($D = 5$ in the network) depth hypotheses for each reference image's pixel. The group-wise warping calculates the similarity between the source and reference features on each depth hypothesis. After that, weighted aggregating the similarity volumes to obtain the cost volume. The probability volume (D channels) is generated by a regularization network. GNet can learn the residual of mean and variance to update the initial Gaussian map. Finally, We design a novel Gaussian loss function (GL) to train the GDINet.

similar method for depth inference. We design the Gaussian loss function with a high convergence to train our network.

We first introduce the multi-scale feature extraction in Section III-A; then we give the details of initialization module in Section III-B; next, we will describe the iteration of Gaussian distribution in the pyramid framework in Section III-C; in addition, we design a novel loss function to train our network in Section III-D; Finally, we give a new filtering method in Section III-E.

A. MULTI-SCALE FEATURE EXTRACTION

We use the Feature Pyramid Network (FPN) [49] (detailed presentation in Appendix Table 12) to hierarchically extract the features of the input images. The FPN makes the feature maps in three scales. On stage l ($l = 0, 1, 2$, stage 2 is the coarsest stage), the resolution of the feature maps are $\frac{H}{2^l} \times \frac{W}{2^l}$, with the input images of size $H \times W$. FPN uses the small receptive field to extract rich texture information and the large receptive field to extract the information of the texture-less areas and fuses them, which is very helpful for reconstruction.

B. INITIALIZATION MODULE

MaGNet [39] applies a large 2D CNN module to learn the initial Gaussian distribution. However, the problem is that the learned value may be too large or too small when datasets are insufficient. So we design an initialization module to limit these parameters to a reasonable range. The structure of this module is shown in Figure 4. We can obtain the initial Gaussian map from the probability volume.

1) GROUP-WISE WARPING

This module converts the source feature maps to the reference view according to the depth hypotheses, and it performs the similarity calculation with the reference feature map to generate the similarity volumes. This module contains differentiable warping and group-wise correlation.

a: DIFFERENTIABLE WARPING

This method warps the source feature maps to the reference view. With images' intrinsic matrices $\{\boldsymbol{K}_i\}_{i=0}^{N}$, relative transformations $\left\{\left[\boldsymbol{R}_{0,i}|\boldsymbol{t}_{0,i}\right]\right\}_{i=1}^{N}$ ($\boldsymbol{R}_{0,i}$ and $t_{0,i}$ represent the rotation and translation from the reference view 0 to the source view i) and the jth depth for a pixel $\boldsymbol{p}$ in the reference image, defined as $d_j := d_j(\boldsymbol{p})$, we can calculate the corresponding pixel $\boldsymbol{p}_{i,j}$ in the view i:

$$\boldsymbol{p}_{i,j} = \boldsymbol{K}_i \cdot \left(\boldsymbol{R}_{0,i} \cdot \left(\boldsymbol{K}_0^{-1} \cdot \boldsymbol{p} \cdot d_j\right) + \boldsymbol{t}_{0,i}\right) \tag{1}$$

Through the differentiable warping, in the feature maps extracted by FPN, we can get one reference feature $\boldsymbol{F}_0$'s corresponding future $\boldsymbol{F}_{i,j}$ in the source view i on the jth depth hypothesis. Then we calculate the similarity of these to generate the similarity volume.

b: GROUP-WISE CORRELATION

In some previous methods [16], [32], the similarity volumes contain all feature channels, increasing the computing time and memory. We use the group-wise correlation method [50] to alleviate this problem. The reference feature $\boldsymbol{F}_0$ and corresponding source feature $\boldsymbol{F}_{i,j}$ with C channels are evenly

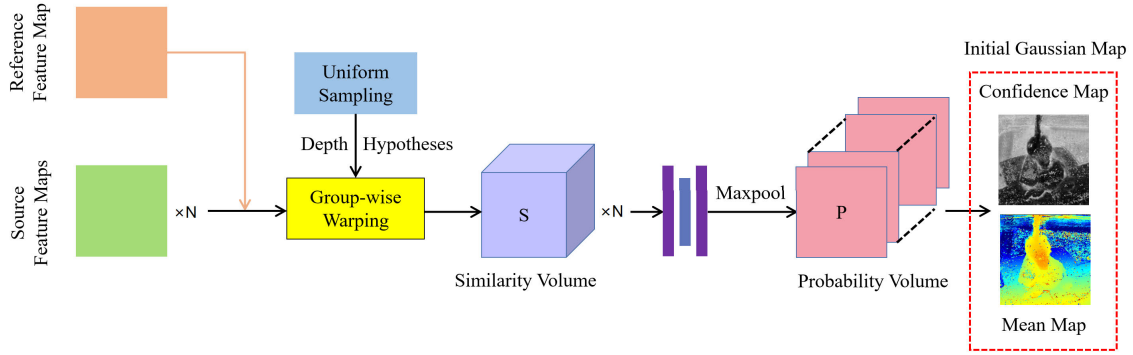


FIGURE 4. Illustration of the initialization module. The uniform sampling module evenly divides the space from the minimum to maximum depth to obtain the depth hypotheses. The group-wise warping module converts the source feature maps to the reference view and calculates similarity with the reference feature map. A similarity network regresses the similarity volume to obtain the probability volume. The Gaussian map can be obtained by processing the probability volume.

divided into G groups. The similarity of the gth group features $\boldsymbol{F}_0^g, \boldsymbol{F}_{i,j}^g$ is calculated as:

$$S\left(\boldsymbol{F}_0^g, \boldsymbol{F}_{i,j}^g\right) = \frac{G}{C}\left\langle \boldsymbol{F}_0^g, \boldsymbol{F}_{i,j}^g \right\rangle \tag{2}$$

where $\langle , \rangle$ represents the inner product. We compute the similarity of all source view's features on all depth hypotheses with the corresponding reference features to obtain the similarity volumes, reducing a large number of feature channels.

2) OBTAINING INITIAL GAUSSIAN MAP

This module fuses similarity volumes into a probability volume and initializes the Gaussian map within a reasonable range.

The N similarity volumes are regressed by a simple 3D CNN similarity network (detailed presentation in Appendix Table 9) with a sigmoid activation function to eliminate environmental factors such as light intensity and reflection. Due to lack of view weights, for each pixel of each depth hypothesis, We choose the perspective that best matches the reference view (the perspective with the highest similarity). So we take the maximum value of all regressed similarity volumes to generate the probability volume.

In the probability volume $\boldsymbol{P}$, $\boldsymbol{P}_k(x)$ represents the possibility that the kth depth hypothesis is the pixel x's true depth. Therefore, the pixel should have the highest probability at the ground truth depth and a gradually decreasing probability further away from the true depth. We define pixels along the depth hypotheses (D layers) dimension are $\{\boldsymbol{P}_k(x)\}_{k=1}^D$. The depth index closest to the ground truth depth is:

$$d_e = \arg\max_k \{\boldsymbol{P}_k(x)\}_{k=1}^D \tag{3}$$

We take the d_eth depth hypothesis as the mean value μ of the pixel's Gaussian distribution so that we can obtain a mean map (depth map). We hope the standard deviation σ of the Gaussian map should have the property: Each pixel has a different variance, those matched confidently should have a low variance, and those matched ambiguously should have a relatively high variance. This can expand the sampling range for pixels with inaccurate depth estimation and narrow the sampling range with confident depth estimation, resulting in more accurate depth estimation. LANet [38] proposes a method to calculate the σ, but it has many hyperparameters that do not easily converge to the best value. We revise it as follows:

$$\sigma = \alpha\left(1 - \max\{\boldsymbol{P}_k(x)\}_{k=1}^D\right) \tag{4}$$

We denote the distance between the two depth hypotheses as dis, and the range of sampling in the adaptive sampling (in Figure 3) as $[\mu - dis, \mu + dis]$ (sampling between two adjacent layers). We use the 3σ principle to sample. Therefore, it can be deduced that the value of α is $\frac{dis}{3}$. So the variance map can be calculated. Because the variance can measure the quality of reconstruction results, we also refer the variance map as confidence map. Finally, we splice the mean map and confidence map to generate the initial Gaussian map. The dimensional analysis is shown in Table 1.

TABLE 1. The dimensional analysis in the initialization module. We denote the number of source images as N, the number of depth hypotheses as D, the resolution of images as H×W, the number of feature channels as C, the number of groups for dividing feature channels as G.

Feature Structure	Dimension
Reference Feature Map& Source Feature Map	H×W×C
Similarity Volumes	N×D×H×W×G
Cost Volume	D×H×W×G
Probability Volume	D×H×W×1
Initial Gaussian Map	H×W×2

C. PYRAMID FRAMEWORK

The pyramid method is proposed by CVP-MVSNet [32]. This paper uses multi-scale feature extraction to generate multiple resolution feature maps. Firstly, the depth map corresponding to the minimum resolution feature map is obtained. After upsampling, it serves as each pixel's initial depth of the next stage. Then sampling depth hypotheses near the depth to calculate a more accurate depth. This coarse-to-fine method dramatically improves the reconstruction effect. However,

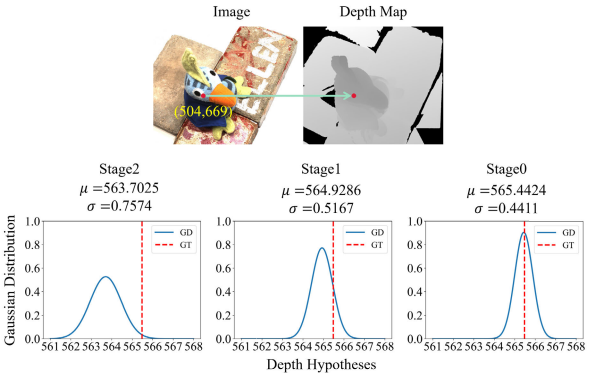


FIGURE 5. Illustration of Gaussian distribution iteration. GD refers to the estimated Gaussian distribution of the point at (504, 669) position. GT refers to the ground truth depth 565.47626 of this point. The coarsest stage 2 with a large depth estimation error has a higher variance, which makes a broader range of depth sampling. As the iteration progresses, the estimated mean value is closer to the ground truth depth and the variance is lower.

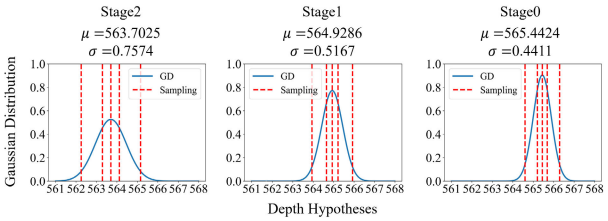


FIGURE 6. The adaptive sampling of Figure 5's Gaussian distribution. Adaptive sampling is densely near the mean value (estimated depth) and sparsely far from the mean value. As the iteration progresses, the depth estimation becomes more accurate and the sampling range becomes more smaller.

this method also uses the regression method to calculate depth maps resulting in non-robust depth estimation. In our method, we use the iteration of the Gaussian map instead of the depth map to solve this problem. The Gaussian distribution iteration of a pixel on different stages is shown in Figure 5. We use the nearest neighbor interpolation to upsample the refined Gaussian map as the initial Gaussian map to the next stage.

1) ADAPTIVE SAMPLING

One of the advantages of Gaussian distribution is that each pixel's depth hypotheses can be sampled adaptively. In some previous pyramid framework [32], [33], [34], the interval of per pixel's hypotheses is a hyperparameter that needs enough intensive sampling. Compared with these methods, adaptive sampling requires only a few layers to achieve state-of-the-art results, which can save much memory. Figure 6 illustrates the adaptive sampling of Figure 5's Gaussian distribution.

As for MaGNet [39], for each pixel (u, v), we use the mean value $\mu_{u,v}$ and variance $\sigma^2_{u,v}$ to sample depth hypotheses. We define the search space is $[\mu_{u,v} - \beta\sigma_{u,v}, \mu_{u,v} + \beta\sigma_{u,v}]$. In our work, we use the Gaussian distribution's 3σ ($\beta = 3$) principle for sampling, which contains 99.74% parts of the distribution. For the standard Gaussian distribution, when the sampling range is $[\beta, -\beta]$, the probability mass $P^*_{u,v}$ covered

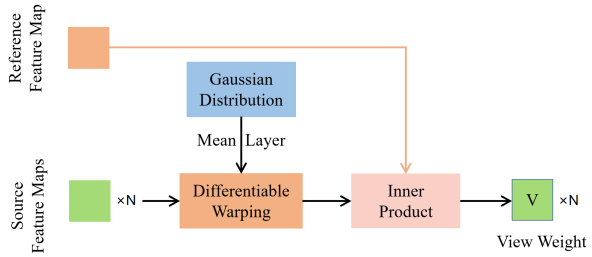


FIGURE 7. Illustration of the view weights module. The differential warping method is described in Section III-B. We only select the mean layer and warp the source feature maps to the reference view on this depth. Then we calculate the similarity with reference feature map (inner product) as the weights of different view images.

by this search space is:

$$\begin{aligned} P^*_{u,v} &= \Phi_{u,v}(\beta) - \Phi_{u,v}(-\beta) \\ &= \text{erf}\left(\frac{\beta}{\sqrt{2}}\right) \end{aligned} \tag{5}$$

where $\Phi_{u,v}$ and $\text{erf}(\cdot)$ are the cumulative distribution function and the error function of standard Gaussian distribution. We sample D layers for pixel (u, v), the kth depth hypothesis $d_{u,v,k}$ is calculated as:

$$\begin{aligned} d_{u,v,k} &= \mu_{u,v} + b_k\sigma_{u,v} \\ b_k &= \frac{1}{2}\left[\Phi^{-1}\left(\frac{k-1}{D}P^*_{u,v} + \frac{1-P^*_{u,v}}{2}\right)\right. \\ &\quad \left.+\Phi^{-1}\left(\frac{k}{D}P^*_{u,v} + \frac{1-P^*_{u,v}}{2}\right)\right] \end{aligned} \tag{6}$$

This adaptive sampling method can achieve a dense sampling near the mean value (current depth of a pixel), and a sparse sampling at a distance from the mean value, which makes the depth hypotheses more conducive.

2) VIEW WEIGHTS AND WEIGHTED AGGREGATION

We can calculate similarity volumes through adaptive sampling and group-wise warping (described in Section III-B). Next, we calculate each perspective's weight according to the initial Gaussian map, then aggregate the similarity volumes to generate the cost volume. The view weights module is shown in Figure 7.

This module is similar to the initialization module. However, this module only selects the mean layer of the Gaussian map as the depth hypothesis. The other difference is that each pixel's feature channels are not divided into G groups. In contrast, whole channels do the inner product to calculate the similarity between reference and source features. So the dimension of the view weights is $N \times H \times W$. We can calculate the cost volume by weighted aggregating similarity volumes.

3) GNet

Then, the probability volume is obtained by regressing the cost volume through a regularization network (detailed presentation in Appendix Table 11). The probability volume represents the probability of each pixel at each depth hypothesis.

TABLE 2. The dimensional analysis in the pyramid framework. On the current stage, we denote the number of source images as N, the number of depth hypotheses as D, the resolution of feature maps extracted by FPN as H×W, the number of feature channels as C, the number of groups for dividing feature channels as G.

Feature Structure	Dimension
Initial Gaussian Map	H×W×2
Similarity Volumes	N×D×H×W×G
View Weights	N×H×W
Cost Volume	D×H×W×G
Probability Volume	D×H×W×1
Gaussian Map	H×W×2

Regression method regresses probability volume and depth hypotheses to calculate each pixel's depth, resulting in poor robustness of depth estimation. Here, we use the probability volume to calculate a refined Gaussian distribution for depth estimation. The module adopts the ResNet [51] to optimize the Gaussian map. The advantage is that this method can alleviate the lack of information and the problem of gradient disappearance, thus significantly improving training accuracy.

The GNet (detailed presentation in Appendix Table 10) is a simple 2D CNN. For each pixel (u, v), the GNet estimates the normalized residual $\left(\mu_{u,v}^{new} - \mu_{u,v}\right)/\sigma_{u,v}$ to update mean value and elu $\left(\sigma_{u,v}^{new}/\sigma_{u,v}\right)$ to update the variance. The elu $(\cdot)$ function is:

$$\text{elu}\,(x) = \begin{cases} x, & x \geq 0 \\ e^x, & x < 0 \end{cases} \tag{7}$$

The elu $(\cdot)$ function can ensure the residual of the standard deviation is greater than 0. So we can get the refined Gaussian map from this module. Moreover, the dimensional analysis in the pyramid framework is shown in Table 2.

D. GAUSSIAN LOSS FUNCTION

In this section, we introduce the loss function NLL (negative log-likelihood) of MaGNet [39] and design the Gaussian loss function GL with excellent convergence. For a pixel (u, v), with the ground truth depth $d_{u,v}^{gt}$, the $\text{NLL}_{u,v}$ is defined as:

$$\begin{aligned} \text{NLL}_{u,v} &= \frac{1}{2}\log\sigma_{u,v}^2 + \frac{L_2}{2\sigma_{u,v}^2} \\ L_2 &= \left(d_{u,v}^{gt} - \mu_{u,v}\right)^2 \end{aligned} \tag{8}$$

This loss function has only one minimum point in the domain of $\mu_{u,v}$ and $\sigma_{u,v}$. The loss function has the minimum value where the $\sigma_{u,v}$ equals $\sqrt{L_2}$ when we fix the $\left|d_{u,v}^{gt} - \mu_{u,v}\right|$. Figure 8 shows loss function $\text{NLL}_{u,v}^0$ where $L_2 = 10$ and $\text{NLL}_{u,v}^1$ where $L_2 = 50$.

We can observe that the $\sigma_{u,v}$ is smaller when the $\mu_{u,v}$ is closer to the $d_{u,v}^{gt}$. It means that when the depth estimation error is small, the $\sigma_{u,v}$ is low, reducing the sampling range to obtain more accurate depth layers and when it is challenging to limit the error L_2, the pixel haves a relative high $\sigma_{u,v}$, which can increase the range of depth sampling.

However, this loss function has two disadvantages. One is that when $\sigma_{u,v} > \sqrt{L_2}$, the trend of the NLL is too flat,

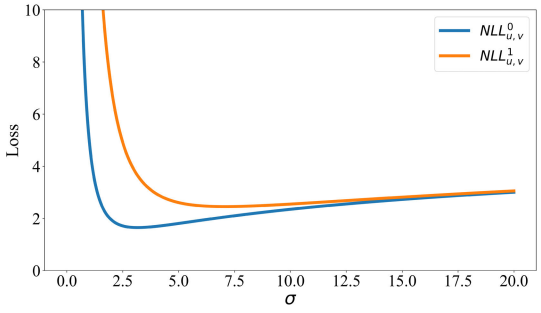


FIGURE 8. Illustration of $\text{NLL}_{u,v}^0$ and $\text{NLL}_{u,v}^1$. The minimum point of $\text{NLL}_{u,v}^0$ is lower than that of $\text{NLL}_{u,v}^1$.

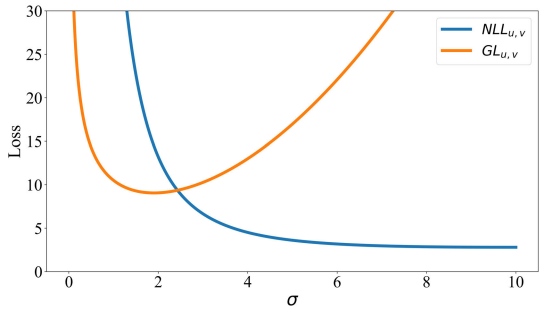


FIGURE 9. Illustration of $\text{GL}_{u,v}$ and $\text{NLL}_{u,v}$ when $\left|d_{u,v}^{gt} - \mu_{u,v}\right| = 10$. The $\text{GL}_{u,v}$ has a strong convergence and a more reasonable variance.

which may lead to poor convergence. The other is when the value of $\left|d_{u,v}^{gt} - \mu_{u,v}\right|$ is high, the $(\cdot)^2$ in the L_2 will lead to excessive variance, resulting in an unreasonable range of depth hypotheses. We design a novel loss function named Gaussian loss to solve these problems as follows:

$$\begin{aligned} \text{GL}_{u,v} &= \frac{1}{2}\sigma_{u,v}^2 + \frac{L_s}{2\sigma_{u,v}^2} \\ L_s &= \begin{cases} \frac{1}{2}\left(\mu_{u,v} - d_{u,v}^{gt}\right)^2, & \left|\mu_{u,v} - d_{u,v}^{gt}\right| < 1 \\ \left|\mu_{u,v} - d_{u,v}^{gt}\right| - \frac{1}{2}, & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

The Gaussian loss function is shown in Figure 9. This function has the only minimum point $\sigma = \sqrt[4]{L_s}$ when fixing the $\left|d_{u,v}^{gt} - \mu_{u,v}\right|$, and has a strong convergence. The variance calculation is more reasonable, significantly promoting the reconstruction effect.

E. POST-PROCESSING

We use photometric consistency and geometric consistency to filter the pixels with large error in depth maps and then project all depth maps to generate 3D point clouds as shown in Figure 10.

1) PHOTOMETRIC CONSISTENCY FILTERING

The regions with a high depth estimated error also have a high variance. So we set a threshold to generate a photometric consistency from the confidence map. When the standard deviation of one pixel is higher than this threshold, we filter

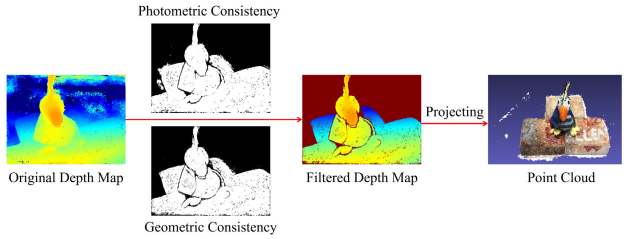


FIGURE 10. The process of post-processing. The photometric consistency and geometric consistency filter the error depth. Then projecting all depth maps to generate the point cloud.

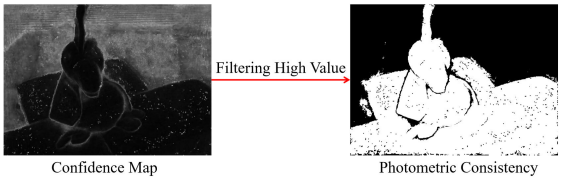


FIGURE 11. Illustration of calculating photometric consistency. The white areas of the photo consistency are the reserved regions.

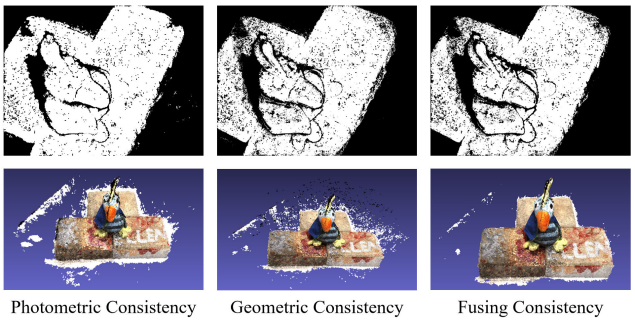


FIGURE 12. Consistencies and 3D reconstruction results filtered by corresponding consistencies on the 4*th* scene of DTU dataset.

this depth, as shown in Figure 11. In our network, we set this threshold as 1.4. The left column of Figure 12 shows the photometric consistency of a image and the 3D point cloud only filtered by photometric consistency. The photometric consistency is mainly used to filter out the inaccurate points outside the object.

2) GEOMETRIC CONSISTENCY FILTERING

This method is followed by [34]. Geometric consistency aims to filter out pixels with high re-projection error. The middle column of Figure 12 presents the geometric consistency of a image and the 3D point cloud only filtered by geometric consistency. The geometric consistency is mainly used to filter out the inaccurate points on the object.

Combining the photometric consistency and the geometric consistency, the final consistency and the final 3D point cloud is shown in the right column of Figure 12.

IV. EXPERIMENTS AND RESULTS

A. DATASETS

We train our network on the DTU dataset and evaluate it on the DTU and Tanks & Temples datasets.

1) DTU DATASET

DTU dataset [21] contains 124 different scenes for MVS. Each scene is photographed from different directions and different light intensities. The camera is fixed on the mechanical arm, and the rotation angle of the mechanical arm is strictly controlled, so the internal and external parameters of the camera can be accurately obtained. Each scene has 49 views, and each view has 7 different brightness. Therefore, there are 343 pictures in each scene folder. The resolution of each image is 1600×1200. This dataset contains each perspective's photos, camera parameters, masks, and depth maps. We select 79 scenes to train our network and 22 to evaluate as the same as [32].

2) TANKS & TEMPLES DATASET

Tanks & Temples [22] dataset is mainly utilized to verify the networks' robustness and whether they still have a relatively accurate reconstruction ability for scenes with large light changes and dynamic targets. This dataset divides scenes into advanced scenes (6 scenes) and intermediate scenes (8 scenes). The advanced scenes include more uncertain conditions compared with the intermediate scenes. The resolution of the images in the advanced scenes is 1920×1080, while the intermediate datasets include 2048×1080 sized images and the 1920×1080 sized images. The lidar scans objects to obtain ground truth point clouds. The generated point clouds are uploaded to the official website for evaluation and published on the leaderboard.

We also make a visual comparison on the BlendedMVS dataset [52] (including many large-scale scenes) to illustrate the generalization ability of our network.

B. TRAINING AND EVALUATING CONFIGURATIONS

We present the details of the parameters on the DTU and Tanks & Temples datasets.

1) TRAINING ON DTU DATASET

The original size of images in the dataset is 1600×1200 which will take a lot of time and memory during training. For preprocessing, we resize the images to 864×512 and process the camera intrinsic matrices, depth maps, and masks accordingly. 5 different perspective images of a scene (one reference view and four source views) are the inputs of our network. We build the Gaussian map pyramid with 3 stages, Beginning at the coarsest stage 2. In the FPN, the number of extracted feature channels on stage 2, 1, 0 are seted as 64, 32, 16. In the initialization module, we set the number of depth hypotheses as 48. For adaptive sampling, we set the number of depth hypotheses on each stage as 5 and the sampling range of Gaussian distribution as 3 covering 99.73% of the region. For group-wise correlation, we set the number of groups on stage 2, 1, 0 as 8, 8, 4. The number of iterations on all stages is 2. We set the weight of loss function on stage 2, 1, 0 as 0.8^2, 0.8, 1. Our network is trained with Adam [53] ($\beta_1 = 0.9, \beta_2 = 0.999$) for 16 epochs. We set the learning

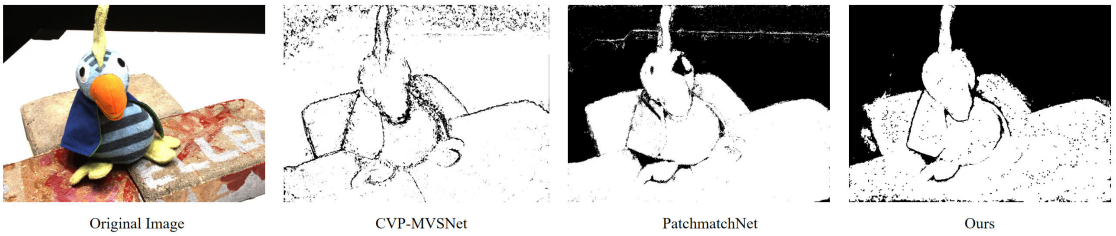


FIGURE 13. The comparison of different methods' photometric consistencies. Our result preserves relatively complete object information.

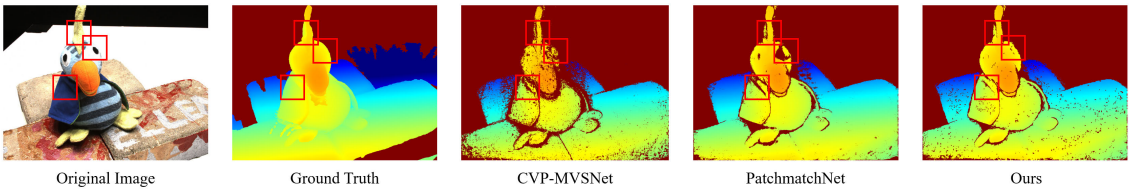


FIGURE 14. The comparison of different methods' filtered depth maps. Our result is more explicit in detail.

rate as 0.0004 multiplied by 0.5 at the 10*th*, 12*th*, 14*th* epoch. Here, we use 4 batches and train on 4 Nvidia TeslaV100-PCIE GPUs.

2) EVALUATING ON DTU DATASET

We resize the original images to 1152 × 864 and choose 5 views of a scene as the inputs with the corresponding camera matrices. We use the 1 batch to evaluate the depth map of each image. We use photometric and geometric consistency to filter the depth map. Finally, we fuse depth maps of all perspectives to generate 3D point clouds [34]. Other parameters are the same as training.

3) EVALUATING ON TANKS & TEMPLES DATASET

We use this dataset to test the robustness of our method. We resize the images of intermediate scenes to 1728×960 and the images of advanced scenes to 1536 × 864, then process the camera intrinsic matrices accordingly. Other parameters are the same as the model evaluated on the DTU dataset.

C. ACCURACY METRICS

To evaluate the estimated reconstruction performance on the DTU dataset, we follow the MATLAB script provided by [21] to compute the accuracy error (Acc.) and completeness error (Comp.) of each reconstruction result, and combine them to calculate the overall error (OA.), which is defined as:

$$\text{OA.} = \frac{\text{Acc.} + \text{Comp.}}{2} \tag{10}$$

Low OA. indicates that the 3D point cloud has a good reconstruction effect.

We calculate the F-score $F(d)$ to evaluate the results generated on the Tanks & Temples dataset. The F-score is defined as:

$$F(d) = \frac{2R(d)P(d)}{R(d) + P(d)} \tag{11}$$

where the $R(d)$ is the recall of the reconstruction and $P(d)$ is the precision of the reconstruction with the distance threshold d. The calculation of $R(d)$ and $P(d)$ can be found in [22]. High $F(d)$ means the 3D cloud point has a good reconstruction effect. We upload the reconstructed point cloud to the official website for evaluation.

D. RESULTS

In this section, we show our reconstruction results on different datasets.

1) RESULTS ON DTU DATASET

The results of our method and other SOTA networks are shown in Table 3. Among all methods, our method ranks 7*th* in Acc., 3*rd* in Comp. and 2*nd* in OA.. A significant advantage of our method is that we can calculate a high-quality photometric consistency to filter the miscellaneous points. We compare our photometric consistency with baseline and patchmatchnet (with the lowest computing memory), as shown in Figure 13. The two networks simply take the probabilities sum over four depth hypotheses nearest the estimated depth to measure the estimation quality. This is an experimental conclusion that if the depth estimation is accurate, the sum of adjacent four-layer depth hypotheses' probabilities is relatively high. Due to the lack of restrictions on the shape of probability, some pixels may experience situations where the depth estimation is accurate but the sum is small, or the depth estimation is inaccurate but the sum is high, resulting in unreasonable filtering. Our filtering method has theoretical support, which can obtain more accurate depth maps and achieve better reconstruction results. The black areas are the filtered pixels and the white areas are the reserved pixels. The photometric consistency generated by CVP-MVSNet only filters several pixels on the object's boundary. The PatchmatchNet's photometric consistency filters many pixels on the object. Our photometric

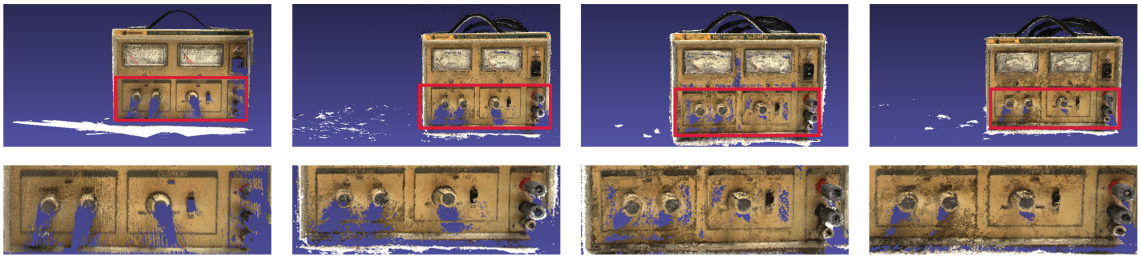


FIGURE 15. The comparison of different methods' reconstruction results on the 11*th* scene of DTU dataset. Our result is more accurate and complete.

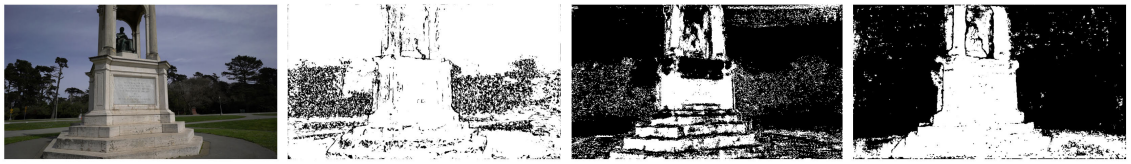


FIGURE 16. The comparison of different methods' photometric consistencies on the Francis scene of Tanks & Temples dataset. Our photometric consistency contains relatively complete object information.

TABLE 3. Quantitative results of reconstruction quality on the DTU evaluation dataset (lower is better). Bold numbers perform best in each column.

Methods	Acc.(mm)	Comp.(mm)	OA.(mm)
Camp [42]	0.835	0.554	0.695
Furu [17]	0.613	0.941	0.777
Tola [13]	0.342	1.190	0.766
Gipuma [19]	**0.283**	0.873	0.578
MVSNet [16]	0.396	0.527	0.462
R-MVSNet [48]	0.383	0.452	0.417
Point-MVSNet [30]	0.342	0.411	0.376
FastMVSNet [35]	0.336	0.403	0.370
CasMVSNet [33]	0.325	0.385	0.355
UCS-Net [36]	0.338	0.349	0.344
PatchmatchNet [34]	0.427	**0.277**	0.351
LANet [38]	0.320	0.349	0.335
TransMVSNet [54]	0.321	0.289	**0.305**
PA-MVSNet [55]	0.313	0.437	0.375
HighRes-MVSNet [37]	0.346	0.345	0.346
Baseline [32]	0.296	0.406	0.351
Ours	0.335	0.312	0.323

consistency preserves relatively complete object information, thus improving the reconstruction effect.

We compare our filtered depth map with these two methods, as shown in Figure 14. The two methods' results are incomplete in some areas, while ours is more explicit in details. The comparison illustrates that the Gaussian distribution iteration method can enhance reconstruction quality.

We also make a visual comparison on other scene as shown in Figure 15. Our result is more accurate and complete at the bottom of the object.

2) MEMORY COMPARISON ON DTU DATASET

The comparison of our method with baseline and TransMVSNet which achieves the highest reconstruction effect is shown in Table 4. The memory comparison of all methods is shown on the left of Figure 2. Our model uses two

TABLE 4. Relating GPU memory to the input resolution of 1152 × 864 on DTU's evaluation set (lower is better). Bold numbers perform best in each column.

Methods	OA.(mm)	Memory(GB)
TransMVSNet	**0.305**	10
Baseline	0.351	5.9
Ours	0.323	**5.1**

strategies to reduce memory and computing time: (1) The group-wise correlation module can significantly reduce the number of feature channels. The number of extracted feature maps's channels are 64, 32, 16 on stage 2, 1, 0 and the module convents those to 8, 8, 4, which saves many resources. (2) We perform adaptive sampling on the Gaussian distribution, which only needs a few depth hypotheses.

3) RESULTS ON TANKS & TEMPLES DATASET

We test the robustness of our method on the Tanks & Temples dataset and compare it with other networks. This dataset is divided into intermediate scenes and advanced scenes. The reconstruction results are shown in Table 5. We use the F-score (higher is better) to evaluate the final results.

Compared with other methods, our result ranks 2*nd* in the intermediate scenes, 3*rd* in the advanced scenes, and 2*nd* in the overall scenes. TransMVSNet has the best reconstruction effect in all scenes, but it takes almost twice as much memory as our method, as shown in Table 4. We compare our photometric consistency with CVP-MVSNet and PatchmatchNet, as shown in Figure 16. CVP-MVSNet's result retains more miscellaneous points and PatchmatchNet's result filters out the pixels on the object, while ours result contains relatively complete object information. The comparison of filtered depth maps is shown in Figure 17, which shows the robustness of our network.

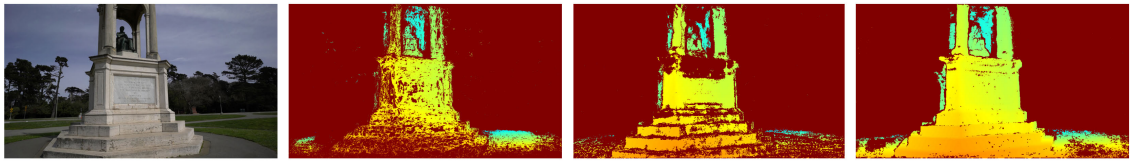


FIGURE 17. The comparison of different methods' depth maps on the Francis scene of Tanks & Temples dataset. Ours is more robust when changing datasets.

FIGURE 18. 3D reconstruction results of some intermediate scenes. Our method has a good reconstruction effect on texture details and texture-less areas.

FIGURE 19. 3D reconstruction results of some advanced scenes. Our method can reconstruct the whole area almost perfectly.

TABLE 5. Different methods perform on the Tanks & Temples dataset evaluating by F-score (higher is better). Bold numbers perform best in each column. Some methods refrain to evaluate on more challenging advanced scenes.

Methods	Intermediate	Advanced	Overall
COLMAP [2]	42.14	27.24	34.69
MVSNet [16]	43.48	-	-
R-MVSNet [48]	48.40	24.91	36.65
CIDER [50]	46.76	23.12	34.94
Point-MVSNet [30]	48.27	-	-
FastMVSNet [35]	47.39	-	-
CasMVSNet [33]	56.42	31.12	43.77
UCS-Net [36]	54.83	-	-
PatchmatchNet [34]	53.15	32.31	42.73
LANet [38]	55.70	-	-
TransMVSNet [54]	**63.52**	**37**	**50.26**
HighRes-MVSNet [37]	49.81	-	-
Baseline [32]	54.03	-	-
Ours	56.42	31.31	43.86

Figure 18. shows our results on some intermediate scenes. The scenes of Family and Panther show our ability to reconstruct texture details, and the Playground scene shows the reconstruction effect for the texture-less areas. Compared with the intermediate scenes, the advanced scenes have more texture-less regions and more demanding lighting transformations, increasing the reconstruction challenge. Figure 19. shows the results on some advanced scenes. Scenes of the Courtroom and Ballroom are indoor conditions, and we almost completely reconstruct them. The scene of Temple is a challenging outdoor scene, and we show its integrity and texture details well. Our method has a significant robustness to environmental impact.

4) A VISUAL COMPARISON ON THE BLENDEDMVS DATASET

We verify the generalization ability of our method on the BlendedMVS dataset. This dataset contains many large-scale scenes increasing the difficulty of reconstruction. The results are shown in Figure 20. Compared with the other two methods, our result is more explicit in detail and complete on the whole, indicating our method has an excellent generalization ability.

E. ABLATION STUDIES

In this section, we do some ablation experiments to prove the rationality and superiority of our network structure. All experiments are trained and evaluated on the DTU dataset.

1) CORE MODULES

Our core modules are the Gaussian distribution iteration (GDI), initialization module (IM) and view weights module (VW). Table 6 shows the effect of these modules.

Overall, the baseline's Acc. is the lowest, while our final method No.4 has the best Comp. and OA.. The reason is

FIGURE 20. Comparison of 3D reconstruction results on the BlendedMVS dataset. Our result is more explicit in detail and more complete on the whole.

TABLE 6. Different core modules perform on the DTU evaluation dataset (lower is better). Bold numbers perform best in each column.

No.	Baseline	GDI	IM	VW	Acc.	Comp.	OA.
1	✓				**0.296**	0.406	0.351
2		✓			0.337	0.433	0.385
3		✓	✓		0.350	0.324	0.337
4		✓	✓	✓	0.335	**0.312**	**0.323**

TABLE 7. GDINet using different loss functions performs on the DTU evaluation dataset (lower is better). Bold numbers perform best in each column.

Loss Function	Acc.(mm)	Comp.(mm)	OA.(mm)
NLL [39]	**0.306**	0.831	0.568
GL	0.335	**0.312**	**0.323**

that CVP-MVSNet has fewer parameters and is easier to converge but ours has a more vital generalization ability. No.2 initializes each image's pixel with the same Gaussian distribution, making the iteration slow and difficult. Comparing No.3 with No.2, the reconstruction effect can be significantly improved when the initialization module gives each pixel a differemt Gaussian distribution. Comparing No.4 with No.3, view weights can promote the accuracy and completeness of reconstruction.

2) LOSS FUNCTION

MaGNet [39] proposes an NLL loss function to train the Gaussian distribution. However, this function is too flat when exceeding the extreme point of standard deviation. We propose a novel loss function, GL (Gaussian loss), with excellent convergence. We use two loss function to train GDINet, and the results are shown in Table 7.

The results show that the Acc. of NLL is lower than GL, but the Comp. is much higher than GL. The reason is NLL uses the quadratic term L_2 while GL uses the single term L_s to optimize each pixel's mean value. So NLL is easier to make the mean convergence. However, L_2 could cause a considerable variance, which leads to an unreasonable sampling of depth hypotheses. Moreover, unreasonable variance leads to unreasonable filtering, which badly impacts the completeness of reconstruction.

3) THE NUMBER OF STAGES IN PYRAMID FRAMEWORK

In our network, we use 3 stages to refine the Gaussian map. We compare it with GDINet using 2 stages. The results are shown in Table 8.

TABLE 8. GDINet with different number of stages performs on the DTU evaluation dataset (lower is better). Bold numbers perform best in each column.

Methods	Acc.	Comp.	OA.	Memory
Baseline	**0.296**	0.406	0.351	5.9
Ours (3 stages)	0.335	**0.312**	**0.323**	5.1
Ours (2 stages)	0.344	0.323	0.333	**4.8**

TABLE 9. Details of similarity network in the initialization module. Input is the feature map (FeaMap) generated by FPN. Conv3D means the 3D convolution operation. BnRelu denotes the batch normalization and rectified linear unit. The output channels, kernal size, stride, padding of convolutional layers are denoted as (output channels, kernal size, stride, padding). D denotes the number of depth hypotheses, H × W denotes the resolution of input images.

Input	Layer	Output	OutDimension
FeaMap	Conv3D,BnRelu,(16,1,1,0)	F_1	D×H×W×16
F_1	Conv3D,BnRelu,(8,1,1,0)	F_2	D×H×W×8
F_2	Conv3D,BnRelu,(1,1,1,0)	**Out**	D×H×W×1

TABLE 10. Details of the GNet. Conv2D means the 2D convolution operation. Input is the probability volume (PV). Relu denotes the rectified linear unit. The output channels, kernal size, stride, padding of convolutional layers are denoted as (output channels, kernal size, stride, padding). H × W denotes the resolution of current stage feature maps.

Input	Layer	Output	OutDimension
PV	Conv2D,Relu,(128,3,1,1)	F_1	H×W×128
F_1	Conv2D,Relu,(128,1,1,0)	F_2	H×W×128
F_2	Conv2D,Relu,(128,1,1,0)	F_3	H×W×128
F_3	Conv2D,Relu,(2,1,1,0)	**Out**	H×W×2

We can conclude that more stages can improve the accuracy and completeness of reconstruction results. But much GPU memory can be saved by using fewer stages while the OA. only increases by 0.01.

V. CONCLUSION

In this paper, we propose a novel probabilistic method significantly improving the reconstruction effect. First, we give each pixel a Gaussian distribution to construct a Gaussian map. The mean value is the estimated depth, and the variance is used for filtering. In addition, we use the pyramid framework to gradually refine the Gaussian map and design a loss function with excellent convergence to train the Gaussian distribution. Finally, we design an initialization module to limit the coarse parameters of distribution within a reasonable range. Our results rank *2nd* on both DTU and Tanks & Temples datasets. The memory consumption is also less than our baseline.

TABLE 11. Details of the regularization network. Input is the cost volume (CV) generated by weighted aggregating similarity volumes. Conv3D means the 3D convolution operation. BnRelu denotes the batch normalization and rectified linear unit. The output channels, kernal size, stride, padding of convolutional layers are denoted as (output channels, kernal size, stride, padding). It should be noted that some convolutions have different stride on the (D, H, W) dimension. D denotes the number of depth hypotheses, H × W denotes the resolution of current stage feature maps. DeConv3D means upsampling convolution, and it has an extra parameter termed output padding (OutPadding).

Input	Layer	Output	OutDimension
CV	Conv3D,BnRelu,(8,3,1,1)	F_1	D×H×W×8
F_1	Conv3D,BnRelu (16,3,(1,2,2),1)	F_2	D×H/2×W/2×16
F_2	Conv3D,BnRelu,(16,3,1,1)	F_3	D×H/2×W/2×16
F_3	Conv3D,BnRelu (32,3,(1,2,2),1)	F_4	D×H/4×W/4×32
F_4	Conv3D,BnRelu,(32,3,1,1)	F_5	D×H/4×W/4×32
F_5	Conv3D,BnRelu (64,3,(1,2,2),1)	F_6	D×H/8×W/8×64
F_6	Conv3D,BnRelu,(64,3,1,1)	F_7	D×H/8×W/8×64
F_7	DeConv3D,BnRelu (32,3,(1,2,2),1) OutPadding=(0,1,1)	F_8	D×H/4×W/4×32
F_8,F_5	Addition	F_9	D×H/4×W/4×32
F_9	DeConv3D,BnRelu (16,3,(1,2,2),1) OutPadding=(0,1,1)	F_{10}	D×H/2×W/2×16
F_{10},F_3	Addition	F_{11}	D×H/2×W/2×16
F_{11}	DeConv3D,BnRelu (8,3,(1,2,2),1) OutPadding=(0,1,1)	F_{12}	D×H×W×8
F_{12},F_1	Addition	F_{13}	D×H×W×8
F_{13}	Conv3D,(1,3,1,1)	**Out**	D×H×W×1

TABLE 12. Details of feature extraction pyramid (FPN) network. Conv2D means the 2D convolution operation. BnRelu denotes the batch normalization and rectified linear unit. The output channels, kernal size, stride, padding of convolutional layers are denoted as (output channels, kernal size, stride, padding). We use the bilinear interpolation for up sampling. H×W denotes the resolution of input images.

Input	Layer	Output	OutDimension
Image	Conv2D,BnRelu,(16,3,1,1) Conv2D,BnRelu,(16,3,1,1) Conv2D,BnRelu,(16,3,1,1)	F_1	H×W×16
F_1	Conv2D,BnRelu,(32,5,2,2)	F_2	H/2×W/2×32
F_2	Conv2D,BnRelu,(32,3,1,1) Conv2D,BnRelu,(32,3,1,1)	F_3	H/2×W/2×32
F_3	Conv2D,BnRelu,(64,5,2,2)	F_4	H/4×W/4×64
F_4	Conv2D,BnRelu,(64,3,1,1) Conv2D,BnRelu,(64,3,1,1)	$\mathbf{Out}_1$	H/4×W/4×64
F_3	Conv2D,(64,1,1,1)	F_5	H/2×W/2×64
F_1	Conv2D,(64,1,1,1)	F_6	H×W×64
Out_1	Interpolate	F_7	H/2×W/2×64
F_5, F_7	Addition	F_8	H/2×W/2×64
F_8	Conv2D,(32,1,1,1)	$\mathbf{Out}_2$	H/2×W/2×32
F_8	Interpolate	F_9	H×W×64
F_6, F_9	Addition	F_{10}	H×W×64
F_{10}	Conv2D,(16,1,1,1)	$\mathbf{Out}_3$	H×W×16

TransMVSNet [54] is the most effective reconstruction network. This method uses an attention mechanism to enhance feature description. In the future, we will add an attention layer after the multi-scale feature extraction structure to extract more accurate object contour information, which may significantly improve the accuracy and completeness of the reconstruction.

APPENDIX
DETAILS OF SOME NETWORK STRUCTURES

Table 9 shows the details of the the similarity network in the initialization module. Table 10, Table 11 and Table 12 describes the GNet, regularization network, and FPN network in the pyramid framework.

REFERENCES

[1] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, Mar. 2005.

[2] J. L. Schönberger and J. Frahm, "Structure-from-Motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.

[3] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 519–528.

[4] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[5] Q. Xu and W. Tao, "Multi-scale geometric consistency guided multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5478–5487.

[6] C. Häne, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, "Stereo depth map fusion for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 1618–1625.

[7] J. Valentin, "Depth from motion for smartphone AR," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–19, Dec. 2018.

[8] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–12, Jun. 2013.

[9] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf, "Casual 3D photography," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–15, Dec. 2017.

[10] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-DOF VR videos with a single 360-camera," in *Proc. IEEE Virtual Reality (VR)*, Mar. 2017, pp. 37–44.

[11] P. Hedman and J. Kopf, "Instant 3D photography," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Aug. 2018.

[12] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[13] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Mach. Vis. Appl.*, vol. 23, no. 5, pp. 903–920, Sep. 2012.

[14] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[15] J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1592–1599.

[16] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 767–783.

[17] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[18] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven, "High accuracy and visibility-consistent dense multiview stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 889–901, May 2012.

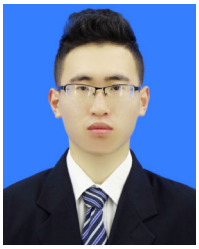
[19] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 873–881.

[20] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 501–518.

[21] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 153–168, Nov. 2016.

[22] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Aug. 2017.
[23] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2538–2547.
[24] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, and K. Schindler, "Learned multi-patch similarity," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1595–1603.
[25] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "SurfaceNet: An end-to-end 3D neural network for multiview stereopsis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2326–2334.
[26] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 365–376.
[27] X. Xiang, Z. Wang, S. Lao, and B. Zhang, "Pruning multi-view stereo net for efficient 3D reconstruction," *ISPRS J. Photogramm. Remote Sens.*, vol. 168, pp. 17–27, Oct. 2020.
[28] E. Akleman, "Deep learning," *Computer*, vol. 53, no. 9, p. 17, Sep. 2020.
[29] K. Luo, T. Guan, L. Ju, H. Huang, and Y. Luo, "P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10451–10460.
[30] R. Chen, S. Han, J. Xu, and H. Su, "Point-based multi-view stereo network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1538–1547.
[31] P. Chen, H. Yang, K. Chen, and Y. Chen, "MVSNet++: Learning depth-based attention pyramid features for multi-view stereo," *IEEE Trans. Image Process.*, vol. 29, pp. 7261–7273, 2020.
[32] J. Yang, W. Mao, J. M. Alvarez, and M. Liu, "Cost volume pyramid based depth inference for multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4876–4885.
[33] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2492–2501.
[34] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "PatchmatchNet: Learned multi-view patchmatch stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14189–14198.
[35] Z. Yu and S. Gao, "Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and Gauss–Newton refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1946–1955.
[36] S. Cheng, Z. Xu, S. Zhu, Z. Li, L. E. Li, R. Ramamoorthi, and H. Su, "Deep stereo using adaptive thin volume representation with uncertainty awareness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2521–2531.
[37] R. Weilharter and F. Fraundorfer, "HighRes-MVSNet: A fast multi-view stereo network for dense 3D reconstruction from high-resolution images," *IEEE Access*, vol. 9, pp. 11306–11315, 2021.
[38] X. Zhang, Y. Hu, H. Wang, X. Cao, and B. Zhang, "Long-range attention network for multi-view stereo," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3781–3790.
[39] G. Bae, I. Budvytis, and R. Cipolla, "Multi-view depth estimation by fusing single-view depth probability with multi-view geometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2832–2841.
[40] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 199–218, Jul. 2000.
[41] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 151–173, 1999.
[42] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2008, pp. 766–779.
[43] Y. Yao, S. Li, S. Zhu, H. Deng, T. Fang, and L. Quan, "Relative camera refinement for accurate dense reconstruction," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 185–194.
[44] S. Shen, "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1901–1914, May 2013.
[45] M. Bleyer, C. Rhemann, and C. Rother, "PatchMatch stereo–stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.
[46] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.
[47] M. Rothermel, K. Wenzel, D. Fritsch, and N. Haala, "SURE: Photogrammetric surface reconstruction from imagery," in *Proc. LC3D Workshop*, vol. 8, no. 2, Berlin, Germany, 2012, pp. 1–9.
[48] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent MVSNet for high-resolution multi-view stereo depth inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5520–5529.
[49] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[50] Q. Xu and W. Tao, "Learning inverse depth regression for multi-view stereo with correlation cost volume," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12508–12515.
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[52] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan, "BlendedMVS: A large-scale dataset for generalized multi-view stereo networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1787–1796.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[54] Y. Ding, W. Yuan, Q. Zhu, H. Zhang, X. Liu, Y. Wang, and X. Liu, "TransMVSNet: Global context-aware multi-view stereo network with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8575–8584.
[55] K. Zhang, M. Liu, J. Zhang, and Z. Dong, "PA-MVSNet: Sparse-to-dense multi-view stereo with pyramid attention," *IEEE Access*, vol. 9, pp. 27908–27915, 2021.

XIAOHAN ZHANG received the B.S. degree from Xinjiang University, in 2020. He is currently pursuing the M.Sc. degree with the School of Mathematics and Statistics, Shandong University, Weihai, China. His major is statistics. His research interests include 3D vision, including multi-view stereo and the application of 3D reconstruction technology in large-scale scenes.



SHIKUN LI received the B.S. degree from Shandong University, Weihai, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Mathematics and Statistics. His major is statistics. His current research interests include the application of statistics in point cloud processing and the efficient solution of 3D vision task by deep learning.