In this paper, we propose an effective and efficient pyramid multi-view stereo (MVS) net for accurate and complete dense point cloud reconstruction. Different from existing deep-learning based MVS methods, our VA-MVSNet incorporates the cost variance between different views by introducing two novel self-adaptive view aggregation: pixel-wise view aggregation and voxel-wise view aggregation. Moreover, to enhance the point cloud reconstruction on the texture-less regions, we extend VA-MVSNet with pyramid multi-scale images input as PVA-MVSNet, where multi-metric constraints are leveraged to aggregate the reliable depth estimation at the coarser scale to fill-in the mismatched regions at the finer scale. Experimental results show that our approach establishes a new state-of-the-art on the DTU dataset with significant improvements in the completeness and overall quality of 3D reconstruction, and ranks 1st on the Tanks and Temples benchmark among all published deep-learning based methods. Our codebase is available at https://github.com/yhw-yhw/PVAMVSNet.

1. Introduction

Reconstructing 3D geometry from photographs is a classic Computer Vision problem for decades. One of the main approaches is Multi-view Stereo (MVS), aiming at recovering dense 3D representation of scenes using stereo correspondences as the main cue given more than two calibrated images [25, 21, 28, 31]. Traditional methods utilize hand-crafted similarity metrics (e.g. NCC) and engineered normalization (e.g. normalized cross correlation) to propagate points and optimize related geometry. Although they have achieved great success on MVS benchmarks [1, 20, 27], many of them still have limitations to handle matching ambiguity especially on the non-Lambertian and texture-less surfaces. Recently, the deep neural network has made tremendous progress in multi-view stereo [15, 40, 41, 16]. These methods learn the knowledge from training data which can guide the model to infer the information hard obtained by stereo correspondences in order to handle matching ambiguity. However, they do not learn and utilize the following important information.

Firstly, the one-stage end-to-end deep MVS architectures [40, 41, 16] that directly learn from images all follow the philosophy that all view images contribute equally to the matching cost volume [10]. For instances, MVS-
Net [40] and R-MVSNet [41] both apply the mean variance operation on multiple cost volume, and DPSNet [16] selects mean average operation. However, images from different views lead to heterogeneous image capture characteristics due to different illumination, camera geometric parameters, scene content variability etc. Based on this observation, we propose a self-adaptive view aggregation module to consider the different significance in multiple matching volumes among images from different views. Our module benefits from the aggregated features by a self-adaptive fusion, where better element-wise matched regions will be enhanced while the mismatched regions will be suppressed.

Secondly, multi-scale information is not leveraged well to measure similarity for texture-less surfaces or regions with repeated patterns. For example, Yao et al. [40, 41] extract the feature of the last layer to generate 3D cost volumes which lose different scale information for further process. In the case of multi-view stereo, a cost volume is a more discriminative representation of the depth than raw images or features [5, 32, 14]. Based on this intuition, we design two novel ways to incorporate multi-scale information. One is to estimate depth maps in a coarse-to-fine manner by utilizing multi-level pyramid feature matching volumes, where a downsampled feature volume has a larger receptive field and alleviates the local ambiguities in texture-less areas. The other is to aggregate multi-scale pyramid depth maps by multi-metric constraints. In particular, to correct the mismatched regions at the finer depth map, we propose to progressively aggregate the reliable depth at the coarser level which satisfies our multi-metric constraints to refine the finer depth map.

To this end, we propose a novel efficient and effective pyramid multi-view stereo network with self-adaptive view aggregation, denoted as PVA-MVSNet. Our method constructs multi-scale pyramid images and processes them parallelly in VA-MVSNet to produce pyramid depth maps. Contrast to previous methods only warping the last layer feature maps to generate 3D cost volumes, VA-MVSNet warps different scale feature maps with differentiable homography to generate 3D feature volumes with coarse-to-fine depth numbers. To regularize information from different views, we propose two self-adaptive element-wise view aggregation modules to regularize multi-view 3D feature volumes into one in an order-independent manner. By a coarse-to-fine depth regression, 3D cost volumes of different scales are utilized to estimate a depth map in a coarse-to-fine manner. After achieving pyramid depth maps through VA-MVSNet, our proposed multi-meteric pyramid depth aggregation corrects the mismatched regions at finer depth maps using the reliable depths at coarser depth maps where the reliabilities are measured by our multi-metric constraints, namely confidence filter, photometric and geometric consistency.

Our main contributions are listed below:

- We propose self-adaptive view aggregation to incorporate the element-wise variances among images from different views, guiding the multiple cost volumes to aggregate a normalized one.
- We design and investigate two ways of incorporating multi-scale information, namely coarse-to-fine depth regression in VA-MVSNet and multi-meteric pyramid depth maps aggregation in PVA-MVSNet.
- Our method establishes a new state-of-the-art on DTU [1] and ranks 1st in all published deep-learning based methods on Tanks and Temples [20].

2. Related Work

Traditional MVS Reconstruction: Traditional MVS reconstruction algorithms can be divided into four types: voxel-based [29, 35], surface-based [12, 4], patch-based [9, 7] and depth map-based methods [7, 2, 33, 8, 42, 26]. Among those methods, the depth map-based MVS reconstruction approaches are more concise and flexible. Recently, many advanced MVS algorithms estimate high-quality depth maps by view selection, local propagation and multi-scale aggregation strategies. These works effectively increase the accuracy of depth map estimation, and achieve competitive point clouds by depth fusions. Zheng et al. [42] propose a depth map estimation method by solving a probabilistic graphical model. Schnberger et al. [26] present a new MVS system named COLMAP which jointly estimates pixel-wise view selection, depth map and surface normal, so that photometric and geometric priors are used to better depict the probability of their graphical model. However, they both have difficulty handling the texture-less regions where photometric consistency is unreliable. Considering this, Xu et al. [37] propose a multi-scale MVS framework with adaptive checkerboard propagation and multi-hypothesis joint view selection to improve the performance. However, this method is highly resource-consuming and difficult to be fully paralleled.

Learning Based Stereo Matching: Recently, the convolutional neural network (CNN) has made tremendous progress in many vision tasks [18, 5, 39, 24, 30], including several attempts on multi-view stereo. Patch-Match learning-based methods [15, 6, 17] pre-warp the image to generate plane-sweep volumes as the input to their networks, which is memory-consuming and can not be trained end-to-end. Two promising approaches, MVSNet [40] and DPSNet [16] both propose the differential homography warping, which implicitly encodes multi-view camera geometries into the network to build the 3D cost volumes and enables an end-to-end training fashion. Furthermore, R-MVSNet [41] replaces 3D-CNN in MVSNet [41] by the recurrent gated recurrent unit (GRU) to reduce memory consumption with a comparable performance on 3D re-
construction during the inference phase, while it still requires the same memory in training phase. Conceptually, Yao et al. [40, 41] extract the feature of the last layer to generate 3D cost volumes, which loses different scale information for further process. In addition, those methods follow the philosophy that the feature volumes from different view images contribute equally, neglecting heterogeneous image capturing characteristics due to different illumination, camera geometric parameters and scene content variability. PointMVSNet [3] is a two-stage coarse-to-fine method, which first generates a coarse depth map by the lower-resolution version MVSNet [40] and iteratively refines the depth error in the point cloud format. Based on the analysis of aforementioned methods, we propose a self-adaptive view aggregation module to incorporate the different significance in multiple feature volumes among images from different views, where better element-wise matched features can be enhanced while the mismatched errors can be suppressed. Furthermore, it is challenging to measure similarity for texture-less and non-Lambertian surfaces. Based on the fact that downsampled features are more discriminative regardless the texture information in photometric and geometric information, we explore two approaches to integrating multi-scale information, namely a coarse-to-fine depth regression and a multi-metric pyramid depth aggregation cast in pyramid images.

3. Method

We first describe the overall architecture of PVA-MVSNet in Sec. 3.1. Then, we introduce the details of VA-MVSNet in Sec. 3.2 and Sec. 3.3. Finally, we present the multi-metric pyramid depth aggregation in Sec. 3.4.

3.1. Overall

Given a reference image $I_{i=0}$ and $I_{i=1,\ldots,N-1}$ neighboring images and corresponding calibrated camera parameters $Q_{i=0}$ and $Q_{i=1,\ldots,N-1}$, where $N$ represents the number of multi-view images, our goal is to estimate the depth map for each reference image. Afterwards, we filter and fuse all the estimated depth maps to reconstruct a 3D point cloud.

For depth estimation of a reference image, our main architecture is illustrated in Fig. 2. We construct an image pyramid with $K$ multiple scales for all images with a downsampling scale factor $\eta$. We denote $k$ pyramid images and corresponding camera parameters as $I_{k=0,\ldots,N-1}$ and $Q_{k=0,\ldots,N-1}$ respectively, where $k=0,\ldots,K-1$. The scale $k=0$ of the pyramid is the original image. We process each level images in the pyramid by V A-MVSNet to obtain depth maps of different scales in parallel. Then we progressively propagate the reliable depths from images with the lower resolution to correct the mismatched errors of images with the original resolution by replacements which satisfy multi-metric constraints, namely confidence filter, photometric and geometry consistency. Finally, we obtain the refined depth map of the raw image. We term our whole method PVA-MVSNet.

3.2. Self-adaptive View Aggregation

In VA-MVSNet (shown in Fig. 3), we first utilize a 2D convolutional network to extract multi-scale features $\{F^l\}^{N-1}_{l=0}$ at different layer $l$ from the $N$ input images. For efficient computation, the first level $l = l_0$ feature is downsampled by four to the original image size with 32 channels, following the downsampling scale factor $\eta = 1/2$ and $\{32, 64, 64\}$ channels of the corresponding layers $l = \{1, 2, 3\}$.

Then each level of different view features will be warped to the reference camera frustum by the differential homog-
raphy [40, 16] with sampling $D_i^l$ layers to build 3D plane-sweep feature volumes $V_i^l$. To handle arbitrary $N$-view images input and the variances among images from different sources, we propose self-adaptive view aggregation to merge $V_i^l_{=0, \ldots, N-1}$ 3D feature volumes into one cost volume $C_i^l$. Let $W_i^l, H_i^l, D_i^l, C_i^l$ denote the width, height, depth sample number and channel number of the l-level input 3D feature volume from image $i$ respectively, the feature volume size can be represented as $S_i^l = W_i^l \cdot H_i^l \cdot D_i^l \cdot C_i^l$, the cost volume $C_i^l$ aggregation can be defined as a function: $M : \mathbb{R}^{S^l} \times \cdots \times \mathbb{R}^{S^l} \rightarrow \mathbb{R}^{S^l}$. In previous work [40, 41, 16], this is a constant function where all views contribute equally, which is not robust for misleading image-registration from the Structure-from-Motion (SfM). Most traditional MVS methods [26, 38] define this function in a heuristic way considering different view variances to aggregate pixel-wise costs among all images. Thus, we propose to employ self-adaptive cost aggregation as this function to flexibly represent potentially different view variance, which can be learned from training data. To achieve this goal, we develop and investigate two self-adaptive view aggregation modules in Fig. 4, which shows how the self-adaptive view selection incorporates the variance between different views. We introduce the attention mechanism [34, 36] for guiding the network to select important matching information in different views. In the point-wise view selection, similar as ACMM [38], we consider that each pixel in the height and width dimension of 3D cost volume has different saliency but is consistent in the depth dimension. The voxel-wise view selection module is a 3D attention-guided mechanism to guide each voxel in 3D feature volumes to learn its own weight.

**Pixel-wise View Aggregation.** The pixel-wise view aggregation introduces a selective weighted attention map in the height and width dimension which considers the depth number hypothesis sharing the common focusing weight. Given multi-view feature volumes $V_i$ where $i = 0, \ldots, N-1$, our regularized cost volumes are aggregated as $c_{d,h,w}$:

$$v_{i,d,h,w} = v_{i,d,h,w} - v_{0,d,h,w},$$  

$$c_{d,h,w} = \frac{\sum_{i=1}^{N} (1 + w_{h,w}) \odot v'_{i,d,h,w}}{N-1}, (1)$$

where $w_{h,w}$ represents a 2D weighted attention map to encode the various pixel-wise saliency among images from different sources and the reference image, and $\odot$ represents element-wise multiply operation.

To generate a 2D weighted attention map, we design a weightnet which consists of several 2D convolutional filters and a ResNet block [11] with the squeezing 2D features as input to learn the $w_{h,w}$:

$$w_{h,w} = \text{weightnet}(f_{h,w}), (3)$$

$$f_{h,w} = \text{CONCAT}(\text{max}_{d,h,w}\text{pooling}(\|v'_{d,h,w}\|_1), \text{avg}_{d,h,w}\text{pooling}(\|v'_{d,h,w}\|_1)), (4)$$

where both max_{d,h,w}pooling and avg_{d,h,w}pooling are used to extract the highest and mean average cost matching information in the depth dimension, and CONCAT(\cdot) denotes the concatenation operation.

**Voxel-wise View Aggregation.** The voxel-wise view aggregation module considers that each pixel with different depth layer hypothesis $d$ is treated differently, where each voxel in 3D feature volume learns its own importance. Based on this, we design a weightnet-3d to directly learn...
the 3D weighted attention map with 3D convolutional filters for selecting useful cost information. The regularized 3D cost volumes \( c_{d,h,w} \) are aggregated by \( v'_{i,d,h,w} \):

\[
c_{d,h,w} = \frac{\sum_{i=1}^{N} (1 + w_{d,h,w}) \odot v'_{i,d,h,w}}{N-1}.
\]

(5)

### 3.3 Coarse-to-fine Depth Estimator

We design a depth estimator by leveraging normalized cost volumes on different levels from self-adaptive view aggregation to regress the depth map in a coarse-to-fine manner in Fig. 3. CNNs are good at extracting high level information by interpolating several convolutional layers with different strides and pooling layers, i.e. higher level features aggregate more semantic information but lack local-wise image details which are beneficial for pixel matching on the texture-less regions. While cost volumes are discriminative to measure the matching similarity between pixels in different images [14], constructing a full cost volume is computationally expensive. Therefore, our network constructs multiple “coarse-to-fine” cost volumes by using an increasing plane-sweep [15] depth number \( D^l = 192/(l+1) \) from level \( l = L - 1 \) to \( l_0 \), where \( L = 4 \). Inspired by recent learning-based stereo matching methods [32], we apply a encoder-decoder 3D CNN for refining the above cost volumes \( C^l \) to generate different scale probability volumes \( P^l \) with a softmax operation along the depth dimension on a 1 channel output volume through a \( 1 \times 1 \) convolutional filter.

To produce a continuous depth estimation, we use soft \( \text{argmin} \) operation [13] on the last output probability feature map to estimate the depth \( \mathbf{E}' \):

\[
\mathbf{E}' = \sum_{d = d_{\text{min}}}^{d_{\text{max}}} d^l \times \mathbf{P}(d^l),
\]

(6)

where \( \mathbf{P}(d^l) \) denotes the estimation probability of all pixels at level \( l \) for the depth hypothesis \( d \). The output depth map at level \( l \) is of the same size as the 2D image feature map at level \( l \). Following MVSNet [40], the probability map is calculated by the sum over the nearest four hypotheses in the 3D probability volume to measure the estimation quality.

#### Training Loss

We use the sum of multi-scale training losses where each scale loss is the same mean absolute error as MVSNet [40], defined as \( \mathcal{L} \):

\[
\mathcal{L} = \sum_{l=l_0}^{L} \lambda_l \sum_{x \in x_{\text{valid}}} \left\| \mathbf{d}^l(x) - \hat{\mathbf{d}}^l(x) \right\|_1,
\]

(7)

where \( x_{\text{valid}} \) denotes the set of valid pixels in the ground truth map, \( \mathbf{d}^l(x) \) and \( \hat{\mathbf{d}}^l(x) \) represent the estimated depth map and the ground truth at the level \( l \) respectively, while \( \lambda_l \) is used to balance the multi-scale depth estimation.

### 3.4 Multi-metric Pyramid Depth Map Aggregation

So far, our proposed network VA-MVSNet in the previous section generates good-enough depth maps for the point cloud reconstruction. We further explore to use reliable depth estimations in a lower-resolution depth map to replace mismatched errors in a higher-resolution depth map with the multi-metric pyramid depth map aggregation.

In a higher-resolution fine estimated depth map, there are still some inaccurate depths with low confidences especially on the texture-less regions, due to the matching ambiguity. Note that the same convolutional filter generally extracts less local-wise, but more global information due to a larger receptive field from a downsampled image in comparison to the original image. An intuitive idea comes out that, we input a image pyramid into VA-MVSNet to generate multi-scale depth maps in parallel, then progressively replace the ambiguous depth estimations at the higher scale by reliable depths at the lower scale. Unlike ACMM [38], we consider multi-metric constraints, specifically, depth confidence, geometric consistency and photometric consistency to guide
reliability and consistency checking. As a result we optimize both depth and probability maps as shown Fig. 5.

Considering a pyramid depth map $D_k = 0, \ldots, K-1$ and a corresponding probability map $P_k = 0, \ldots, K-1$ from VA-MVSNet, we expect to iteratively replace unreliable depth values with low confidence $P_k(p) < 0.5$ at the scale $k$ by reliable depths $P_{k+1}(p) > 0.9$ at the downsampling scale $k+1$, where $P_k(p)$ denotes the confidence of pixel $p$ in the probability map $P_k$. Furthermore, in the lower scale $k+1$ depth maps, we utilize both the photometric consistency and geometric consistency to select well-estimated depths, where the photometric consistency measures the matching quality and the geometric consistency measures the depth consistency among multiple views.

Given a pixel $p$ in $I_i$, we project $p$ to the corresponding pixel $p_{proj}$ in the neighbor image $I_j$ through $D_i(p)$ and camera parameters. In turn, we reproject $p_{proj}$ to the reference image as $p_{reproj}$ with $d_{reproj}$. The photometric and geometric consistency are followed respectively:

$$\|p - p_{reproj}\|_2 < 1,$$  \hspace{1cm} (8)

$$\|D_i(p) - d_{reproj}\|_1 < 0.01 \cdot D_i(p),$$  \hspace{1cm} (9)

where we consider the pixel which satisfies the multi-metric constraints in at least three neighbor views is reliable.

In this way, we progressively select the reliable depths at a lower scale $k+1$ to replace the mismatched depths at $k$ scale. It leads to a final refinement on the original depth map $D_{k=0}$.

### 4. Experiments

#### 4.1. Implementation Details

**Training** We train VA-MVSNet on the DTU dataset [1], which consists of 124 different indoor scenes scanned by fixed camera trajectories in 7 different lighting conditions.

**Evaluation** For testing, we use image view number $N = 5$ and $D = 192$ for depth plane hypothesis in an inverse depth setting. We evaluate VA-MVSNet and PVA-MVSNet with an original input image resolution: $1600 \times 1184$, the pyramid layer $K$ and the downsampling factor $\eta$ in PVA-MVSNet are set to 3 and 1/2 respectively. For Tanks and Temples dataset, the camera parameters are computed by OpenMVG [22] following MVSNet [40] and the input image resolution is set to $1920 \times 1056$.

**Filtering and Fusion** We fuse all the depth maps into a complete point cloud similar to other depth map based MVS methods [8, 40]. In our experiments, we only consider the reliable depth values with confidence larger than 0.9. We utilize the aforementioned photometric consistency and geometric consistency to select those pixels occurring in more than three neighbor views. Finally, the depths are projected to 3D space and fused to produce a 3D point cloud.

| Method            | Mean Distance (mm) |
|-------------------|---------------------|
|                   | Acc. | Comp. | overall |
| Camp [2]          | 0.835| 0.554 | 0.695   |
| Furu [7]          | 0.613| 0.941 | 0.777   |
| Tola [33]         | 0.342| 1.190 | 0.766   |
| Gipuma [8]        | 0.283| 0.873 | 0.578   |
| Colmap [26]       | 0.400| 0.664 | 0.532   |
| SurfaceNet [17]  | 0.450| 1.040 | 0.745   |
| MVSNet [40]       | 0.396| 0.527 | 0.462   |
| R-MVSNet [41]     | 0.385| 0.459 | 0.422   |
| PointMVSNet [3]   | 0.361| 0.421 | 0.391   |
| PointMVSNet-HiRes [3] | 0.342| 0.411 | 0.376   |

VA-MVSNet  
PVA-MVSNet  
0.381 0.361 0.371  
0.372 0.350 0.361  

Table 1. Quantitative results on the DTU evaluation dataset [1] (lower is better). Our methods VA-MVSNet and PVA-MVSNet outperforms all methods in terms of completeness and overall quality with a significant improvement.

Figure 6. Our qualitative results and the ground truth point clouds of Scan 10 and Scan 15 in the benchmark DTU [1].

Following the common practices [15, 17, 40, 41, 3], we split the dataset into training, validation and evaluation datasets which are non-overlapping. We train our network on the training dataset and evaluate on the evaluation dataset. For data preprocessing, we use the same ground truth rendered depth maps as in MVSNet [40]. During training, the input image size is set to $W \times H = 640 \times 512$ and the number of input images $N = 3$. The depth hypotheses are sampled from $425mm$ to $935mm$ with depth plane number $D = 192$ in an inverse manner. We implement our network on PyTorch [23] and the network is trained end-to-end for 16 epochs using Adam [19] with an initial learning rate 0.001 which is decayed by 0.9 every epoch. The $\lambda_{0,1,2,3}$ in the multi-scale training loss is $\{0.32, 0.16, 0.04, 0.01\}$. Batch size is set to 4 on 4 NVIDIA TITANX graphics cards.
4.2. Benchmarks Results

**DTU Dataset** We evaluate our proposed method on the DTU [1] evaluation set. Quantitative results are shown in Tab. 1. The accuracy and completeness are calculated using the official matlab script provided by the DTU [1] dataset. To evaluate the overall reconstruction quality, the overall score is mentioned in [40, 1] calculated by the average of the accuracy and completeness. While Gipuma [8] performs the best regarding to accuracy, our PVA-MVSNet and VA-MVSNet establish a new state-of-the-art both in completeness and overall quality with a significant margin compared with all previous methods [33, 40, 41, 3]. Qualitative results can be found in Fig. 6. In comparison to the ground truth point cloud obtained by scanners, our method generates denser and more complete point clouds especially for the texture-less surfaces. We also compare our depth maps with MVSNet [40] and R-MVSNet [41] in Fig. 7. VA-MVSNet predicts more accurate and complete depth map than other methods especially on the texture-less regions. Moreover, the depth estimation can be further improved by utilizing multi-metric pyramid depth aggregation in PVA-MVSNet.

**Tanks and Temples Benchmark** DTU dataset is taken indoor and under well-controlled environment with fixed camera trajectory. For further testing the generalization of PVA-MVSNet, we evaluate our proposed method **without any fine-tuning** on Tanks and Temples, which is a more complicated outdoor dataset. We utilize multi-view number $N = 5$, depth plane number $D = 192$ and image size $W \times H = 1920 \times 1056$ for all scenes. The depth range, camera parameters and matching source image pairs are the same as in MVSNet [40].

Our method ranks 1st among all published deep-learning based methods including Point-MVSNet [3], R-MVSNet [41] and MVSNet [40] on the intermediate set. The f-score increases from 43.48 to 49.08 (larger is better, date: Nov. 16, 2019) compared with MVSNet [40], which demonstrates the strong generalization of PVA-MVSNet. The reconstructed point clouds are shown in Fig. 8.

| Components          | Acc.  | Comp. | Overall |
|---------------------|-------|-------|---------|
| baseline            | 0.454 | 0.372 | 0.413   |
| + PixelVA           | 0.404 | 0.369 | 0.386   |
| + VoxelVA           | 0.390 | 0.368 | 0.379   |
| + CTF               | 0.437 | 0.366 | 0.402   |
| + VoxelVA + CTF     | 0.381 | 0.361 | 0.371   |
| + VoxelVA + CTF + MMP | **0.372** | **0.350** | **0.361** |

Table 2. Contributions of different components in our architecture on the DTU [1] evaluation dataset.

4.3. Ablation Studies

In this section, we provide ablation experiments to quantitatively and qualitatively analyze the strengths of the key components of our architecture. For following studies, all experiments use the same setting in Sec. 4.1 and are tested on the DTU [1] evaluation dataset in Tab. 2. Besides, we validate different components of our network during training on the validation dataset by the mean absolute error between the estimated depth map and the ground truth depth map as shown in Fig. 9, which is used for quantitative measurements on the depth map reconstruction.

**Self-adaptive View Aggregation** As shown in Tab. 2, compared with our baseline method, which is the implementation of MVSNet [40] in PyTorch [23], Both PixelVA and VoxelVA can improve the results of 3D reconstruction point cloud with a significant improvement, especially on the accuracy of reconstruction quality. Specifically, the VoxelVA provides a 0.064 increase on accuracy, which is better than the PixelVA 0.050 due to the learning variance of the depth wise hypothesis. During training, as shown in Fig. 9, the depth error on valuation dataset drops significantly, which also demonstrates the efficacy of self-adaptive view aggregation.
Figure 8. Point cloud results on the intermediate set of Tanks and Temples [20] benchmark, which show the generalization of our method on complex outdoor scenes.

Coarse-to-fine Depth Regression We replace the original depth estimator in the baseline method with our coarse-to-fine depth estimator, denoted as “CTF”. As shown in Tab. 2 and Fig. 9, the “CTF” enhances the performance on both depth map estimation and 3D point cloud reconstruction, especially on the completeness quality, which benefits from the aggregation of multi-scale features.

Multi-metric Pyramid Depth Aggregation We consider the influence of multi-metric pyramid depth aggregation quantitatively and qualitatively, which is denoted as “MMP” in Tab. 2 and PVA-MVSNet in Fig. 7. By utilizing “MMP”, both the accuracy and completeness of 3D point cloud reconstruction increase notably by 0.009 and 0.011 respectively. Cast in pyramid multi-scale images, by correcting the mismatched error depths in a higher scale depth map with reliable depth estimations in a lower scale depth map, PVA-MVSNet improves VA-MVSNet by generating more dedicated and complete depth maps, as shown in (e) and (f) in Fig. 7.

Methods | H,W,D | Mem. | Time. | Overall |
|--------|-------|------|-------|---------|
| MVSNet | 1600, 1184, 256 | 15.4GB | 1.18s | 0.462 |
| R-MVSNet | 1600, 1184, 512 | 6.7GB | 2.35s | 0.422 |
| PointMVSNet | 1260, 960, 96 | 7.2GB | 1.69s | 0.391 |
| PointMVSNet-HiRes | 1600, 1152, 96 | 8.7GB | 5.44s | 0.376 |
| VA-MVSNet | 1600, 1184, 192 | 17.3GB | 0.85s | 0.371 |
| PVA-MVSNet | 1600, 1184, 192 | 17.3GB | 0.95s | 0.361 |

Table 3. Comparisons on the time and memory consumption on the evaluation DTU [1] dataset. MVSNet and R-MVSNet are implemented in TensorFlow while other methods are in PyTorch.

To better prove the effectiveness of multi-metric depth aggregation with pyramid as input, we directly utilize multi-metric depth aggregation on the temporary intermediate multi-scale depth maps in the coarse-to-fine depth regression of VA-MVSNet with one original image as input, where a lower-resolution depth map is warped into 3D features to regress a higher-resolution depth map. The improvement of 3D point cloud by this is very slight and can be ignored, we find out that the intermediate multi-scale pyramid depth maps have similar distribution, because the priors contained in a coarser depth map have already been used for the regression of a finer depth map.

Considering this, our multi-metric depth aggregation with pyramid images as input can efficiently and effectively improve the quality of 3D point cloud.

4.4. Runtime and Memory Performance

Given time and memory performance in Tab. 3, all methods are tested on GeForce RTX 2080 Ti. Even if VA-MVSNet runs with biggest memory consumption, but also runs fast at a speed of 0.85s / view with a decent depth reconstruction. Furthermore, unlike PointMVSNet [3], multi-scale pyramid images can be processed independently in parallel. Therefore, with little extra time about 0.1s for multi-metric pyramid depth aggregation, the performance of 3D point cloud reconstruction increases significantly.
from 0.371 to 0.361 in PVA-MVSNet on DTU [1] dataset.

5. Conclusion

We have presented a novel pyramid multi-view stereo network with self-adaptive view aggregation. The proposed VA-MVSNet dynamically selects the element-wise feature importance while suppresses the mismatching cost, and regresses a depth map in a coarse-to-fine manner. Cast in multi-scale pyramid images, PVA-MVSNet estimates a refined depth map with multi-metric pyramid depth aggregation for further improving the depth estimation especially on texture-less regions. Experimental results demonstrate that our proposed method PVA-MVSNet not only establishes a new state-of-the-art on DTU dataset, but also ranks 1st on Tanks and Temples benchmark without any fine-tuning compared with other published deep-learning based methods.

Acknowledgements This project was supported by the National Key R&D Program of China (No.2017YFB1002700, No.2017YFB0203000) and NSFC of China (No.61632003, No.61661146002, No.61631001).

References

[1] Henrik Aanæs, Rasmus Ramsbol Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120(2):153–168, 2016. 1, 2, 6
[2] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008. 2, 6
[3] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. *arXiv preprint arXiv:1908.04422*, 2019. 3, 6, 7, 8
[4] Daniel Cremers and Kalin Kolev. Multiview stereo and silhouette consistency via convex functions over convex domains. *PAMI*, 33(6):1161–1174, 2010. 2
[5] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philipp Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2
[6] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *CVPR*, 2016. 2
[7] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2009. 2, 6
[8] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 2, 6
[9] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007. 2
[10] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *ICCV*, 2017. 1
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4
[12] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009. 2
[13] Sina Honari, Pavlo Molchanov, Stephen Tyree, Pascal Vincent, Christopher Pal, and Jan Kautz. Improving landmark localization with semi-supervised learning. In *CVPR*, 2018. 5
[14] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI*, 35(2):504–511, 2012. 2, 5
[15] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018. 1, 2, 5, 6
[16] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: end-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 1, 2, 3, 4
[17] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2017. 2, 6
[18] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 2
[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
[20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *TOG*, 36(4):78, 2017. 1, 2, 8
[21] Maxime Luillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *PAMI*, 27(3):418–433, 2005. 1
[22] Pierre Moulon, Pascal Monasse, Renaud Marlet, et al. Openmvg. an open multiple view geometry library, 2014. 6
[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 6
[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015. 2
[25] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1
[26] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2, 4, 6
[27] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 1
[28] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In CVPR, 2006. 1

[29] Sudipta N Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In ICCV, 2007. 2

[30] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In ACCV, 2018. 2

[31] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In CVPR, 2008. 1

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In CVPR, 2018. 2, 5

[33] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. Machine Vision and Applications, 23(5):903–920, 2012. 2, 6, 7

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, 2017. 4

[35] George Vogiatzis, Carlos Hernández Esteban, Philip HS Torr, and Roberto Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. PAMI, 29(12):2241–2246, 2007. 2

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, 2015. 4

[37] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In CVPR, 2019. 2

[38] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In CVPR, 2019. 2, 5

[39] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In ECCV, 2018. 2

[40] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In ECCV, 2018. 1, 2, 3, 4, 5, 6, 7

[41] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In CVPR, 2019. 1, 2, 3, 4, 6, 7

[42] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In CVPR, 2014. 2
Supplementary Material for Pyramid Multi-view Stereo Net with Self-adaptive View Aggregation

Number of views | Acc.(mm) | Comp.(mm) | Overall(mm)
---|---|---|---
$N = 2$ | 0.415 | 0.467 | 0.441
$N = 3$ | 0.380 | 0.379 | 0.380
$N = 5$ | 0.381 | 0.361 | 0.371
$N = 7$ | **0.378** | **0.359** | **0.369**

Table 1. Ablation study on different number of views $N$ on DTU [1] dataset. (The model is trained with the number of view $N = 3$)

1. Number of Views Ablation Study

In this section, we investigate the influence of number of views in VA-MVSNet. VA-MVSNet can process an arbitrary number of image views and well leverage the variant importance in different views due to our proposed self-adaptive view aggregation. Our model is trained with the number of views $N = 3$ and utilizes voxel-wise view aggregation referred to Sec. 3.2. We compare the reconstruction results of our model with the input number of views $N = 2, 3, 5, 7$ on DTU evaluation dataset. The results, as shown in Tab. 1, demonstrate the reconstruction ability is improved with an increasing number of views and also illustrate the efficacy of our proposed self-adaptive view aggregation module.

2. Network Architecture

This section describes the network architecture details of VA-MVSNet in Tab. 3. VA-MVSNet consists of feature extractor, differentiable homography warping, self-adaptive view aggregation and coarse-to-fine depth regression. The details of weightnet in pixel-wise view aggregation and weightnet-3d in voxel-wise view aggregation are presented in Tab. 2. VA-MVSNet takes multi-view images as input to estimate depth maps in an end-to-end manner during training and testing, and all layers only require the GPU memory with size linear to the input image resolution.

3. Reconstruction Results

This section show all the reconstructions results on DTU [1] evaluation dataset in Fig. 1. PVA-MVSNet is able to reconstruct dense and accurate point clouds for all scenes.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120(2):153–168, 2016. 1, 3
| Input          | Layer                  | Output                  | Output Size          |
|---------------|------------------------|-------------------------|----------------------|
| \( \{ I_i \}_{i=0}^{N-1} \) | \( N \times H \times W \times 3 \) |                         |                      |

**Image Features Extractor**

| Layer                  | Output                  | Output Size          |
|------------------------|-------------------------|----------------------|
| \( \text{conv2d}_0 \)  | \( \text{ConvBR}, K = 3, S = 1, F = 8 \) | \( H \times W \times 8 \) |
| \( \text{conv2d}_1 \)  | \( \text{ConvBR}, K = 5, S = 2, F = 16 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 16 \) |
| \( \text{conv2d}_2 \)  | \( \text{ConvBR}, K = 3, S = 1, F = 16 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 16 \) |
| \( \text{conv2d}_3 \)  | \( \text{ConvBR}, K = 5, S = 2, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_4 \)  | \( \text{ConvBR}, K = 3, S = 1, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_5 \)  | \( \text{ConvBR}, K = 5, S = 2, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_6 \)  | \( \text{ConvBR}, K = 3, S = 1, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_7 \)  | \( \text{ConvBR}, K = 5, S = 2, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_8 \)  | \( \text{ConvBR}, K = 3, S = 1, F = 32 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 32 \) |
| \( \text{conv2d}_9 \)  | \( \text{ConvBR}, K = 5, S = 2, F = 64 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 64 \) |
| \( \text{conv2d}_{10} \)| \( \text{ConvBR}, K = 3, S = 1, F = 64 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 64 \) |
| \( \text{conv2d}_{11} \)| \( \text{ConvBR}, K = 5, S = 2, F = 64 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 64 \) |
| \( \text{conv2d}_{12} \)| \( \text{ConvBR}, K = 3, S = 1, F = 64 \) | \( \frac{1}{2} H \times \frac{1}{2} W \times 64 \) |

**Differentiable Homography Warping**

| Layer                  | Output                  | Output Size          |
|------------------------|-------------------------|----------------------|
| \( \{ V^0_i \}_{i=0}^{N-1} \) | \( \text{Warp} \) | \( N \times D \times \frac{1}{4} H \times \frac{1}{4} W \times 32 \) |
| \( \{ V^0_i \}_{i=0}^{N-1} \) | \( \text{Warp} \) | \( N \times 4 D \times \frac{1}{4} H \times \frac{1}{4} W \times 32 \) |
| \( \{ V^0_i \}_{i=0}^{N-1} \) | \( \text{Warp} \) | \( N \times \frac{1}{4} D \times \frac{1}{16} H \times \frac{1}{16} W \times 64 \) |
| \( \{ V^0_i \}_{i=0}^{N-1} \) | \( \text{Warp} \) | \( N \times \frac{1}{4} D \times \frac{1}{32} H \times \frac{1}{32} W \times 64 \) |

**Self-adaptive View Aggregation**

| Layer                  | Output                  | Output Size          |
|------------------------|-------------------------|----------------------|
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Aggregate} \) | \( D \times \frac{1}{4} H \times \frac{1}{4} W \times 32 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Aggregate} \) | \( D \times \frac{1}{8} H \times \frac{1}{8} W \times 32 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Aggregate} \) | \( D \times \frac{1}{16} H \times \frac{1}{16} W \times 64 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Aggregate} \) | \( D \times \frac{1}{32} H \times \frac{1}{32} W \times 64 \) |

**Coarse-to-fine Depth Regression**

| Layer                  | Output                  | Output Size          |
|------------------------|-------------------------|----------------------|
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3D}, K = 3, S = 1, F = 8 \) | \( D \times \frac{1}{4} H \times \frac{1}{4} W \times 8 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_1 \) | \( D \times \frac{1}{8} H \times \frac{1}{8} W \times 16 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_2 \) | \( D \times \frac{1}{16} H \times \frac{1}{16} W \times 32 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_3 \) | \( D \times \frac{1}{32} H \times \frac{1}{32} W \times 64 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_4 \) | \( D \times \frac{1}{64} H \times \frac{1}{64} W \times 128 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_5 \) | \( D \times \frac{1}{128} H \times \frac{1}{128} W \times 256 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_6 \) | \( D \times \frac{1}{256} H \times \frac{1}{256} W \times 512 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_7 \) | \( D \times \frac{1}{512} H \times \frac{1}{512} W \times 1024 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_8 \) | \( D \times \frac{1}{1024} H \times \frac{1}{1024} W \times 2048 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_9 \) | \( D \times \frac{1}{2048} H \times \frac{1}{2048} W \times 4096 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_{10} \) | \( D \times \frac{1}{4096} H \times \frac{1}{4096} W \times 8192 \) |
| \( \{ C^0_i \}_{i=0}^{N-1} \) | \( \text{Conv3d}_{11} \) | \( D \times \frac{1}{8192} H \times \frac{1}{8192} W \times 16384 \) |

Table 3. The details of VA-MVSNNet. We denote \( \text{Conv}, \text{DeConv} \) as 2D convolutional filter, 3D convolutional filter, 3D deconvolutional filter and use BR to represent the abbreviation of batch normalization and the ReLU. \( + \) and \( & \) represent the element-wise addition operation and concatenation. \( K, S, F \) are the kernel size, stride, output channel number respectively. \( N, H, W \) denote input view number, image height, image height and depth hypothesis number.
Figure 1. Reconstruction results on DTU [1] evaluation set.