P-MVSNet: Learning Patch-wise Matching Confidence Aggregation for Multi-View Stereo

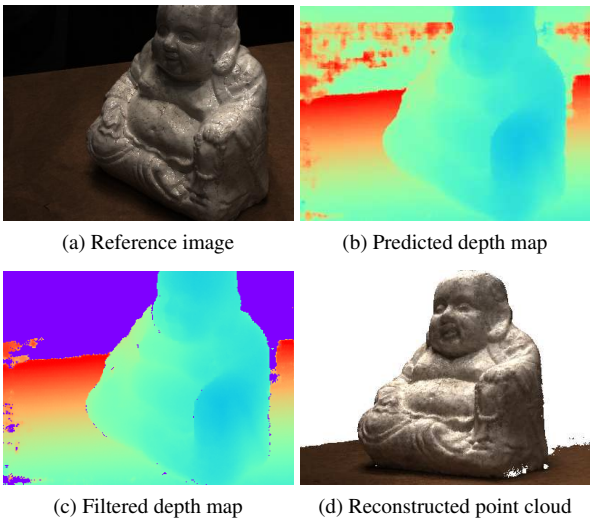
Keyang Luo[1], Tao Guan[1,3], Lili Ju[2,3], Haipeng Huang[3], Yawei Luo[1*]
[1]Huazhong University of Science and Technology, China
[2]University of South Carolina, USA [3]Farsee2 Technology Ltd, China
{kyluo, qd_gt, royalvane}@hust.edu.cn, ju@math.sc.edu, haipenghuang@farsee2.com

Abstract

Learning-based methods are demonstrating their strong competitiveness in estimating depth for multi-view stereo reconstruction in recent years. Among them the approaches that generate cost volumes based on the plane-sweeping algorithm and then use them for feature matching have shown to be very prominent recently. The plane-sweep volumes are essentially anisotropic in depth and spatial directions, but they are often approximated by isotropic cost volumes in those methods, which could be detrimental. In this paper, we propose a new end-to-end deep learning network of P-MVSNet for multi-view stereo based on isotropic and anisotropic 3D convolutions. Our P-MVSNet consists of two core modules: a patch-wise aggregation module learns to aggregate the pixel-wise correspondence information of extracted features to generate a matching confidence volume, from which a hybrid 3D U-Net then infers a depth probability distribution and predicts the depth maps. We perform extensive experiments on the DTU and Tanks & Temples benchmark datasets, and the results show that the proposed P-MVSNet achieves the state-of-the-art performance over many existing methods on multi-view stereo.

1. Introduction

Multi-view Stereo (MVS) aims to estimate a geometric representation of the underlying scene from a collection of images with known camera parameters, and is a fundamental computer vision problem which has been extensively studied for decades. Inspired by the great success of Convolutional Neural Networks (CNNs) in many computer vision fields like semantic segmentation [28, 26], scene understanding [27] and stereo matching [5], several learning-based MVS methods [43, 33] have been introduced recently and can be divided into two types: voxel based ones and depth-map based ones. The recent MVS benchmarks [1, 22] show that learning-based methods can produce high-quality 3D models comparable to the conventional state-of-the-arts although there are still has a large of rooms for improvement. Furthermore, it is also observed that the depth-map based algorithms outperform the voxel based ones.



(a) Reference image (b) Predicted depth map

(c) Filtered depth map (d) Reconstructed point cloud

Figure 1: Multi-view 3D reconstruction of *Scan114* of DTU dataset [1]. (a) The reference image; (b) the predicted depth map by the proposed P-MVSNet; (c) the filtered depth map; (d) the reconstructed 3D point cloud.

An essential step of the depth-map based learning methods is to construct a pixel-wise matching confidence/cost volume. The basic idea is to first build a plane-sweep volume based on the plane-sweep algorithm [6] at a reference image picked from the input images, then calculate the matching cost between each pixel in the reference images and its corresponding ones in other adjacent images on each sampled depth hypothesis. A popular matching metric used in most existing methods is the variance of features between the pair of pixels, in which the contributions of all involved pixel pairs to the matching cost are treated equally. Such metric is often not conducive to the pixel-wise dense matching actually. For instance, when the features of a pixel

*Corresponding author.

in adjacent non-reference images are very similar but do not match the corresponding feature in the reference image, a low matching cost will be generated for this pixel, which potentially tends to give it a wrong estimation in the depth map. Therefore, we argue that one should highlight the importance of pixels in the reference image during calculation of the matching confidence volume.

After accumulating the matching confidences from multiple images on each of the sampled planes and storing them in a cost volume, current methods usually regularize the pixel-wise cost volume or infer the depth-map directly, which is not very robust to noisy data. Moreover, the constructed plane-sweep volume contained in the corresponding frustum is essentially anisotropic – we can infer the corresponding depth map along the depth direction of the matching cost volume, but cannot get the same information along other directions. This fact can be used to guide the regularization of matching confidence volume.

Based on the above motivations, we propose a new end-to-end network of P-MVSNet for multi-view stereo. In the proposed P-MVSNet, we first construct a pixel-wise matching confidence volume based on the mean-square error (MSE) that gives preference to the reference image, then use a patch-wise confidence aggregation module to aggregate the pixel-wise matching confidence on all sampled planes, finally a hybrid 3D U-Net with isotropic and anisotropic 3D convolutions is employed to exploit the context information of the matching confidence volume and estimate the depth maps (with a refinement structure specially designed for the higher resolution level). The point-cloud reconstruction follows from the predicted depth maps with some filtering and fusion schemes.

The major contributions of this paper are summarized below:

- We propose a patch-wise matching confidence aggregation module to build the matching cost volume, which is robust and accurate for noisy data.
- We design a hybrid 3D U-Net to infer a latent probability volume from the matching confidence volume and estimate the depth maps.
- We develop depth-confidence and depth-consistency criteria for filtering and fusing depth maps in order to improve accuracy and completeness of the point-cloud reconstruction.
- Our method achieves the state-of-the-art performance over many existing methods for multi-view stereo on the DTU and *Tanks & Temples* benchmark datasets.

2. Related Work

Conventional MVS Based on the underlying object models, conventional MVS methods often can be categorized into four types: *Patch* based algorithms [10, 25] regard scene surfaces as collections of small spatial patches, which first reconstruct the patches in textured regions, and then propagate them to low-textured ones to densify the reconstructed patches; *Deformable polygonal meshes* based algorithms [46, 9, 24] require a good initial guess of the scene surface to initialize the surface evolution and then iteratively improve the multi-view photometric consistency; *Voxel* based algorithms [39, 32, 41] first compute a bounding box which contains the scene and divide it into voxel grids, and then pick out the voxels attached to the scene surface, thus the reconstruction accuracy is restricted by the voxel resolution in these algorithms; *Depth map* based algorithms [11, 37, 42] first estimate depth maps for individual images and then merge all depth maps into a consistent point cloud. Overall, *Depth map* based approaches outperform the other three, and a detailed review can be found in [8, 22].

Learning-based stereo Due to the power of deep learning techniques, stereo matching has made great progresses in recent years. Han *et al.* [14] and Zbontar *et al.* [47] introduced convolutional networks to compute the similarity of a pair of image patches at almost the same time. To refine the disparity maps, Guney *et al.* [13] proposed to use object knowledge to resolve matching ambiguities. Gidaris *et al.* [12] proposed to learn to detect incorrect labels and then replace the incorrect ones with new labels and optimize the renewed labels. Seki and Pollefeys [38] applied the predict SGM penalties to the cost regularization. GC-Net [19] and PSMNet [5] proposed to predict whole disparity maps without post-processing via end-to-end networks. Although learning-based stereo matching approaches significantly outperform the conventional methods, all of them require accurate rectified stereo image pairs. Unfortunately, acquiring the exact rectified image pairs is intractable, especially for images with more varying viewpoints. As a consequent, they may not be able to produce very accurate depth information and fail to reconstruct 3D models by fusing the depth maps.

Learning-based MVS To overcome the blemish of the stereo matching, several recent works focused on the learning-based MVS reconstruction. One route of these approaches is based on the volumetric representation of the scene surfaces. Ji *et al.* proposed the first learning-based MVS reconstruction system SurfaceNet [17], which first unprojects the images into a pre-computed 3D voxel space, then uses a generic 3D CNN to regularize and classify whether a voxel belongs to the scene surface. Both LSM [18] and RayNet [33] first encode the projection geometry into a cost volume, then LSM uses a 3D CNN to predict if each voxel is on the object surface while RayNet uses the unrolled Markov Random Field. All these *voxel* based approaches [17, 18, 33] suffer from the common de-

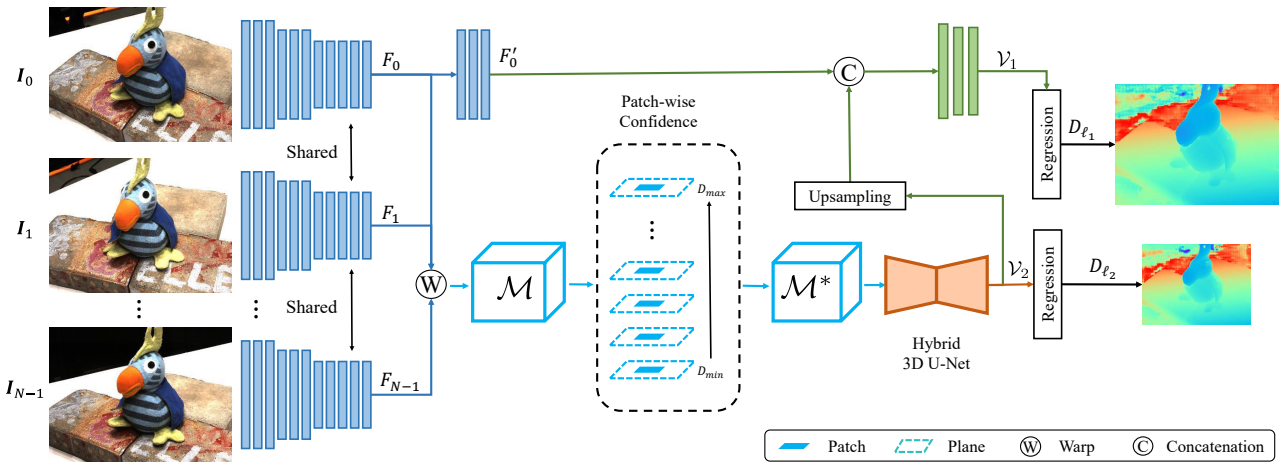


Figure 2: Architecture of the proposed P-MVSNet. It includes a weight-sharing image feature extractor (blue), a patch-wise matching confidence aggregation module (light blue), a hybrid 3D U-Net (orange) and a refinement structure (green).

ficiency of the voxel representation. Another route is based on the plane-sweep stereo to build the matching confidence volume and then the depth maps to represent the scene. Hartmann *et al.* proposed to directly estimate multi-patch similarity [15] by a multi-stream CNN architecture to displace the handcrafted metric function for MVS reconstruction and then reconstruct depth maps by a standard plane-sweep stereo. DeepMVS [16] transforms the depth estimation of each pixel in images into a multi-class classification problem. In this approach, the input image pairs are first matched via a shared patch matching network, then the matching result is aggregated into a intra-volume, and finally a max-pooling layer is used to aggregate the multi intra-volumes into an inter-volume to predict the depth map. In contrast, MVSNet [43] first extracts image features, and then generates the matching cost volume upon a pixel-wise variance-based metric, and finally a generic 3D U-Net is used to regularize the matching cost volume to estimate the depth maps.

3. Architecture of P-MVSNet

The proposed P-MVSNet is a deep learning neural network in an end-to-end manner, which includes a weight-sharing image feature extractor, a patch-wise matching confidence aggregation module, a hybrid 3D U-Net-based depth map inference network, and a refinement structure to improve spatial resolution of the estimated depth map. The overall architecture of P-MVSNet is illustrated in Figure 2.

3.1. Feature extraction

The weight-sharing feature extraction network follows the idea of the encoder-decoder architecture and its parameters are detailed in Table 1. For N input images of size $H \times W$, let $\boldsymbol{I}_0$ and $\{\boldsymbol{I}_j\}_{j=1}^{N-1}$ denote the input reference image and its adjacent images respectively. We first extract the feature map $F_i \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$ for each of the images $\boldsymbol{I}_i$ $(0 \leq i \leq N-1)$ using the encoder part, which consists of eleven 2D convolutional blocks. We define the F_i as the $level$-2 (ℓ_2 for short) *feature map* of the image $\boldsymbol{I}_i$. The decoder part consists of three 2D convolutional blocks and produces the $level$-1 (ℓ_1 for short) *feature map* $F_0' \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}$ only for the reference image. The number of feature channels C is set to be 16 for all output feature maps. The ℓ_2 feature maps $\{F_i\}_{i=0}^{N-1}$ will be used to construct the Matching Confidence Volume (MCV) at a relatively small spatial resolution, while the ℓ_1 feature map F_0' will be utilized to guide the estimation of higher resolution depth map.

Table 1: Summary of the feature extraction network. Each convolutional layer represents a block of convolution, batch normalization and ReLU non-linearization (unless otherwisely stated).

Input images size: H×W×3		
Name	Layer Description	Output Tensor
Encoder for all input images		
conv0_0	3×3 conv, stride 1	H×W×8
conv0_1	3×3 conv, stride 1	H×W×8
conv0_2	3×3 conv, stride 1	H×W×8
conv1_0	5×5 conv, stride 2	½H×½W×16
conv1_1	3×3 conv, stride 1	½H×½W×16
conv1_2	3×3 conv, stride 1 (no BN&ReLu)	½H×½W×16
conv2_0	5×5 conv, stride 2	¼H×¼W×32
conv2_1	3×3 conv, stride 1	¼H×¼W×32
conv2_2	3×3 conv, stride 1	¼H×¼W×32
	add conv2_0 & conv2_2 features	¼H×¼W×32
conv2_3	3×3 conv, stride 1	¼H×¼W×32
conv2_4	1×1 conv, stride 1 (no BN&ReLu)	¼H×¼W×16
Decoder for the reference image		
conv3_0	3×3 transposed conv, stride 2	½H×½W×16
	add conv3_0 & conv1_2 features	½H×½W×16
conv3_1	3×3 conv, stride 1	½H×½W×16
conv3_2	3×3 conv, stride 1 (no BN&ReLu)	½H×½W×16

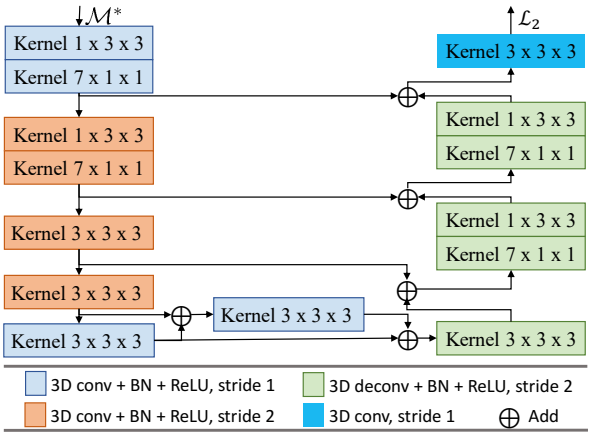


Figure 3: Architecture of the hybrid 3D U-Net network.

3.2. Learning patch-wise matching confidence

Based on the extracted ℓ_2 feature maps and their corresponding cameras parameters, we first construct a pixel-wise matching confidence volume (MCV) based on a plane-sweep volume generated by the standard plane-sweeping stereo, then learn to aggregate the pixel-wise MCV into a patch-wise MCV to increase the matching robustness and accuracy.

Denote the pixel-wise MCV as $\mathcal{M} = \mathcal{M}(d, \boldsymbol{p}, c)$, which represents the matching confidence of the c-th feature channel between the pixel $\boldsymbol{p}$ in F_0 and its corresponding pixels (in adjacent feature maps) induced by a plane hypothesis π_d (d is the depth value of π_d). Therefore, $\mathcal{M}$ is a $Z \times \frac{H}{4} \times \frac{W}{4} \times C$ shaped tensor where Z denotes the number of sampled hypothetical planes, and we define it by:

$$\mathcal{M}(d, \boldsymbol{p}, c) = \exp\left(-\frac{\sum_{j=1}^{N-1}(F_j(\boldsymbol{p}', c) - F_0(\boldsymbol{p}, c))^2}{N-1}\right) \quad (1)$$

where $\boldsymbol{p}'$ is the corresponding pixel of $\boldsymbol{p}$ in the adjacent feature map F_j and $F_j(\boldsymbol{p}', c)$ is computed using bilinear interpolation.

Next, we learn to aggregate $\mathcal{M}$ based on a patch around $\boldsymbol{p}$ on π_d to obtain a patch-wise matching confidence volume $\mathcal{M}^* = \mathcal{M}^\star(d, \boldsymbol{p}, c)$ defined as:

$$\begin{aligned} \mathcal{M}^\star(d, \boldsymbol{p}, c) &= tanh\left(\rho_3(\Omega_2(\mathcal{M}^a(d, \boldsymbol{p}, c)))\right), \\ \mathcal{M}^a(d, \boldsymbol{p}, c) &= \rho_1(\mathcal{M}(d, \boldsymbol{p}, c)) + \rho_2(\Omega_1(\mathcal{M}(d, \boldsymbol{p}, c))) \end{aligned} \quad (2)$$

where $\Omega_1(\cdot)$ defines a patch ω_1 of size 3×3 centered at $\boldsymbol{p}$ on the hypothesized plane π_d, $\Omega_2(\cdot)$ denotes the union of three adjacent patches along the depth direction centered at $\boldsymbol{p}$, and ρ_1, ρ_2 and ρ_3 are some learnable functions that take into account the multi-channel feature matching confidence. Here, we choose ρ_1 and ρ_2 to be $1 \times 1 \times 1$ and $1 \times 3 \times 3$ kernel-sized 3D convolutional blocks followed by BN and ReLU respectively, so that ρ_1 only focuses on the integration of multi-channel matching confidence at $\boldsymbol{p}$ while ρ_2 fuses the matching information of neighboring pixels in ω_1. ρ_3 is defined as a $3 \times 3 \times 3$ kernel-sized 3D convolutional layer followed by BN, which learns to aggregate matching confidence between multiple patches. Finally, a $tanh$ activation is utilized to regularize the confidence. Unlike the conventional MVS algorithms which aggregate the matching confidence/cost in a heuristic way, we use a learnable patch-wise aggregation function. The aggregated feature matching confidence at each pixel on each hypothesized plane is a vector rather than a scalar and the weight for each feature channel is adjusted automatically, which can improve the matching robustness and accuracy for noisy data.

3.3. Depth-map inference

As shown in Figure 3, $\mathcal{M}^*$ is fed into a hybrid 3D U-Net to infer a latent probability volume (LPV) denoted as $\mathcal{V}_2 = \mathcal{V}_2(d, \boldsymbol{p})$, which indicates the latent probability distribution of each pixel of F_0 along the depth direction and its size is $Z \times \frac{H}{4} \times \frac{W}{4}$. The hybrid 3D U-Net consists of several anisotropic and isotropic 3D convolutional blocks as well as a deep aggregation layer [45]. On shallow layers, we use two kinds of anisotropic convolution with kernel sizes of $1 \times 3 \times 3$ and $7 \times 1 \times 1$ respectively. The $1 \times 3 \times 3$ shaped computational blocks concentrate on fusing information on each sampled hypothetical planes, while the $7 \times 1 \times 1$ shaped 3D convolutional layers can enlarge the receptive field in the depth direction to exploit global information with a relatively low computational cost. On deep layers and the output layer, we use the isotropic $3 \times 3 \times 3$ shaped 3D convolutions to fuse more context information.

Next we use the depth regression as proposed in [43] to estimate the depth map D_{ℓ_2}. A probability volume (PV) $\mathcal{P}_2$ is first calculated from $\mathcal{V}_2$ via the softmax operation $\sigma(\cdot)$ along the depth direction, which is referred to as a soft attention mechanism and more robust than classification-based methods. The predicted depth at a labeled pixel $\boldsymbol{p}$ in D_{ℓ_2} is then calculated as the sum of each depth d weighted by its probability for $\boldsymbol{p}$ as

$$D_{\ell_2}(\boldsymbol{p}) = \sum_{d=D_{\min}}^{D_{\max}} d \cdot \mathcal{P}_2(d, \boldsymbol{p}) \quad (3)$$

where $D_{\min}$ and $D_{\max}$ denote the minimum and maximum sampled depth respectively.

In practice, the depth map D_{ℓ_2} is often relatively low-resolution, therefore we use the ℓ_1 feature map F_0' to guide the estimation of a depth map D_{ℓ_1} at higher resolution through the refinement structure. First, F_0' and the upsampled $\mathcal{V}_2$ are concatenated as a $(C+Z)$-channel input, which is forwarded to a $(C+Z)$-channel 2D convolutional layer and two Z-channel 2D convolutional layers to obtain the latent probability volume $\mathcal{V}_1$. BN and ReLU are included in

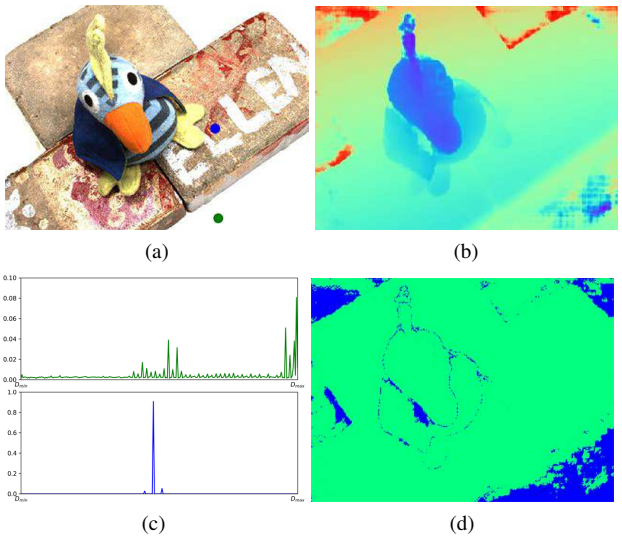


Figure 4: An illustration of the inferred depth map and its confidence map. (a) The reference image, in which we mark one pixel with an outlier depth in green and another pixel with an inlier depth in blue; (b) the inferred depth map from P-MVSNet; (c) two typical behaviors of the probability distribution in depth direction (the "multi-peak" green line for the outlier depth, and the "single-peak" blue line for the inlier depth); (d) the estimated confidence map.

the first two layers but excluded from the last layer. Then we obtain the probability volume $\mathcal{P}_1$ and the final estimated depth map D_{ℓ_1} at the higher resolution based on $\mathcal{V}_1$, just like the way of obtaining $\mathcal{P}_2$ and D_{ℓ_2}.

3.4. Loss function

For the depth regression, we utilize the differences between the ground truth depth maps and the estimated depth maps to train the proposed P-MVSNet. The loss function is formulated as

$$\begin{aligned} Loss = & \frac{\alpha}{|\Phi_2|} \sum_{\boldsymbol{p} \in \Phi_2} \left\| D_{\ell_2}(\boldsymbol{p}) - D^{\star}_{\ell_2}(\boldsymbol{p}) \right\|_1 \\ & + \frac{1-\alpha}{|\Phi_1|} \sum_{\boldsymbol{p} \in \Phi_1} \left\| D_{\ell_1}(\boldsymbol{p}) - D^{\star}_{\ell_1}(\boldsymbol{p}) \right\|_1 \end{aligned} \tag{4}$$

where Φ_2 and Φ_1 are the set of labeled pixels, $D^{\star}_{\ell_1}$ and $D^{\star}_{\ell_2}$ are the corresponding ground truth depth maps. The hyper-parameter α controls the relative importance of the two terms, which is set to be 0.5 in experiments.

4. Point-Cloud Reconstruction

After a set of N raw depth maps are inferred from the proposed P-MVSNet by taking each of the input images in turns as the reference image, a concern is that they may not agree well with each other on common regions due to errors in the estimated depths. We introduce two filtering criterions to discard the wrongly predicted depth values: 1) the *depth-confidence criterion* to remove the obviously untrustworthy prediction and 2) the *depth-consistency criterion* to abandon inconsistent depth values across adjacent images.

Depth-confidence It is clear that the estimated depth would hold a great confidence when the probability distributions along depth direction of the pixel $\boldsymbol{p}$ has a single peak. We first define a confidence map C_2 corresponding to the depth map D_{ℓ_2} at the coarse resolution level as:

$$C_2(\boldsymbol{p}) = \max \left\{ \mathcal{P}_2(d, \boldsymbol{p}) \mid d \in [D_{\min}, D_{\max}] \right\} \tag{5}$$

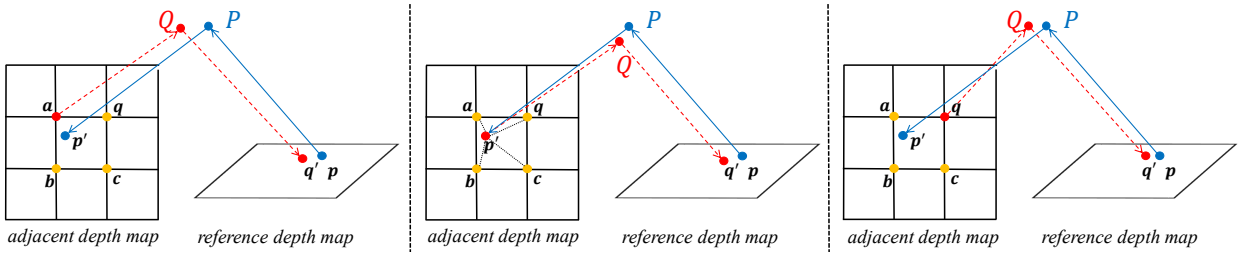
for each label pixel $\boldsymbol{p}$. The confidence map C_1 corresponding to D_{ℓ_1} at the fine level is calculated as follows: we first upsample C_2 to the same size as D_{ℓ_1}, denoted as U_1, then the confidence of D_{ℓ_1} at $\boldsymbol{p}$ is computed as

$$C_1(\boldsymbol{p}) = U_1(\boldsymbol{p}) + \max \left\{ \mathcal{P}_1(d, \boldsymbol{p}) \mid d \in [D_{\min}, D_{\max}] \right\}. \tag{6}$$

We refer to Figure 4 for an illustration of the inferred depth map and its confidence map. The depth confidence criterion aims to filter out the predicted depth with low confidence: for each pixel in a depth map, we regard it as a unreliable depth if its confidence is below ξ ($\xi = 0.5$ is set in experiments) and then abandon it.

Depth-consistency The *depth-consistency criterion* is used to enforce the consistency of the predicted depth among multiple adjacent depth maps. To achieve this goal, we first project a reference pixel $\boldsymbol{p}$ through its estimated depth $\hat{d}(\boldsymbol{p})$ (either D_{ℓ_1} or D_{ℓ_2} depth map as needed) to another depth map and determine its corresponding pixel $\boldsymbol{q}$ by the following way: if the ground truth camera parameters are available, the standard bilinear depth scheme is taken, otherwise a novel *"depth-consistency first"* strategy is used, as illustrated in Figure 5. Then we re-project $\boldsymbol{q}$ back to the reference depth map through its depth estimation $\hat{d}(\boldsymbol{q})$. If the reprojected point $\boldsymbol{q}'$ and its depth $\hat{d}(\boldsymbol{q}')$ satisfy $|\boldsymbol{q}' - \boldsymbol{p}| < \epsilon$ and $|\hat{d}(\boldsymbol{q}') - \hat{d}(\boldsymbol{p})|/\hat{d}(\boldsymbol{p}) < \eta$ ($\epsilon = 0.9$ and $\eta = 0.01$ are set in experiments), we deem the predicted depth $\hat{d}(\boldsymbol{p})$ at $\boldsymbol{p}$ is consistent between these two depth maps. If the predicted depth $\hat{d}$ can maintain consistency in at least μ ($\mu = 2$ is set in experiments) adjacent depth maps, we regard it as a reliable prediction, otherwise it is abandoned. Such strategy could improve the completeness of the fused point cloud.

After filtering all depth maps by the above two filtering strategies, most wrong predictions are expected to be removed and relatively clean depth maps are obtained. We then fuse all depth maps into a consistent point cloud to represent the 3D scene surface based on the method developed in [11]. In addition, we also remove some outlier points as usual using the point neighborhood statistics [35].



(a) nearest depth [29] (b) bilinear depth [11] (c) depth-consistent first depth (ours)

Figure 5: Illustrations of different depth picking schemes. The point $\boldsymbol{p}'$ in an adjacent depth map is the back-projected point of the pixel $\boldsymbol{p}$ in the reference depth map. Among the five candidate points $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{p}'$ and $\boldsymbol{q}$, $\boldsymbol{q}$ has the most similar depth to the 3D point P in the adjacent depth map while $\boldsymbol{p}$ is the nearest point of $\boldsymbol{p}'$. The nearest depth scheme picks point $\boldsymbol{a}$, the bilinear depth scheme uses $\boldsymbol{p}'$ directly, while our *depth-consistency first* strategy regards $\boldsymbol{q}$ as the true corresponding point of $\boldsymbol{p}$.

5. Experimental Results

5.1. Datasets

The following datasets are used for performance evaluation and comparison of the proposed P-MVSNet with many existing state-of-the-art methods for multi-view stereo.

DTU dataset [1]: The DTU robot image dataset is a large scale multi-view stereo benchmark. It composes of 124 different scenes and each scene captures 49 or 64 images of resolution 1600×1200 pixels under seven different lighting conditions. The difference of material, texture and geometric property of captured scenes varies greatly and the provided ground-truth point clouds are acquired by a structured light scanner. We generate the ground-truth depth maps using the same technical scheme as MVSNet [43]. We divide this whole dataset into the training, validation and evaluation sets[1] as done in SurfaceNet [17] and MVSNet [43]. There are a total of 27,097 images used for training of P-MVSNet. Notice that ground-truth models are not always complete and may contain holes in some areas.

Tanks & Temples dataset [22]: Unlike the DTU dataset acquired under well-controlled laboratory environment, the *Tanks & Temples* dataset benchmark sequences were acquired under realistic conditions. Its *intermediate set* consists of eight scenes: *Family*, *Francis*, *Horse*, *Lighthouse*, *M60*, *Panther*, *Playground*, and *Train*. These captured scenes have varying scales, surface reflection and exposure conditions, moreover, no camera parameter information is provided for them. We will use this dataset to validate the *generalization ability* of the tested methods.

5.2. Model specifications

We implemented P-MVSNet in TensorFlow [2]. Inspired by the recently proposed **SWATS** [20] which switches from Adam to SGD when certain conditions are satisfied in order to improve the generalization performance of the trained networks, we divided the training process of P-MVSNet into two phases. In the first stage, we used the Adam solver [21] ($\beta_1 = 0.9, \beta_2 = 0.999$) to train our network for 4 epochs, in which the learning rate is initialized to be 0.001 and then decayed every $10,000$ iterations with an exponential rate of 0.9. After reaping the benefits of the Adam solver's rapid convergence in the first stage, we next switched to SGD to fine-tune the pre-trained network for 4 epochs with a learning rate of 0.0005 which also decays similarly as the first stage. For both training stages, we used images of size $H = 512$ and $W = 640$ as inputs to P-MVSNet, and each of the training sample consists of 1 reference image and 2 adjacent images. The fronto-parallel hypothesized planes of each reference image were uniformly sampled from $D_{\min} = 425mm$ to $D_{\max} = 935mm$ with a resolution of $2mm$ (thus $Z = \frac{935-425}{2} + 1 = 256$). We trained P-MVSNet with one Nvidia Titan RTX GPU on the DTU dataset only, which took approximately three days.

[1]Th validation set: scans {3, 5, 17, 21, 28, 35, 37, 38, 40, 43, 56, 59, 66, 67, 82, 86, 106, 117}, the evaluation set: scans {1, 4, 9, 10, 11, 12, 13, 15, 23, 24, 29, 32, 33, 34, 48, 49, 62, 75, 77, 110, 114, 118} and the training set: the remaining 79 scans.

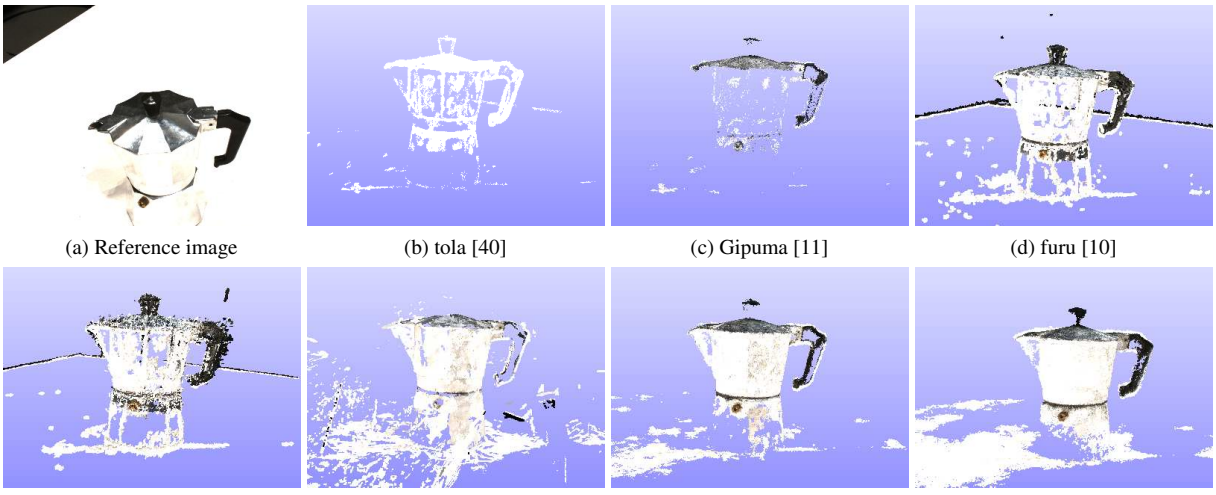
Table 2: Comparison of the depth maps produced by MVSNet and different model variants of the proposed P-MVSNet on the DTU dataset.

Method		Mean abs. depth err.	Prediction prec. (σ)	Prediction prec. (3σ)
MVSNet [43]		7.25	72.84%	87.96%
P-MVSNet (D_{ℓ_2})	w/o P	5.54	75.18%	89.25%
	H→ G	5.82	73.66%	88.71%
	Full ver.	**5.26**	**75.43%**	**90.88%**
P-MVSNet (D_{ℓ_1})	w/o P	5.74	73.06%	88.07%
	H→ G	6.13	72.76%	87.21%
	Full ver.	**5.43**	**73.97%**	**88.47%**

P: Patch-wise aggregation H: Hybrid U-Net G: Generic U-Net

5.3. Ablation study

We conduct an ablation study to compare some model variants of the proposed P-MVSNet on the performance of predicting depth maps with the DTU evaluation set, which consists of 7546 ground-truth depth maps (22 scans × 7



(a) Reference image (b) tola [40] (c) Gipuma [11] (d) furu [10]

(e) camp [4] (f) SurfaceNet [17] (g) MVSNet [43] (h) P-MVSNet

Figure 6: Visualization of the reconstructed point clouds for the model *scan* 77 from the DTU dataset by different methods.

Table 3: Performance results for the fused 3D point clouds of the DTU evaluation scenes.

Method	Mean accuracy	Mean completeness	Overall
P-MVSNet	0.406	**0.434**	**0.420**
camp [4]	0.836	0.555	0.696
furu [10]	0.612	0.939	0.776
tola [40]	0.343	1.190	0.767
Gipuma [11]	**0.274**	1.193	0.734
SurfaceNet [17]	0.450	1.043	0.746
MVSNet [43]	0.396	0.527	0.462

lighting patterns $\times$ 49 images/pattern). More specifically, one model variant is obtained by removing the patch-wise confidence aggregation module from the full version of P-MVSNet, and another by replacing the hybrid 3D U-Net with the generic 3D U-Net. To the best of our knowledge, MVSNet [43] is so far the top performer on the DTU dataset, so we also compare them with MVSNet. The quality of predicted depth maps is evaluated based on the commonly used *mean absolute depth error*, as well as the *prediction precision* defined by:

$$P(\tau) = \frac{100}{|\mathcal{R}|} \sum_{\boldsymbol{p} \in \mathcal{R}} \left[|\hat{d}(\boldsymbol{p}) - d^*(\boldsymbol{p})| < \tau \right], \qquad (7)$$

where $\mathcal{R}$ denotes the evaluated pixel set, $\hat{d}$ and d^* are the predicted depth and ground-truth depth respectively, τ is the distance threshold and $[\cdot]$ is the Iverson bracket. Here we set τ as σ and 3σ respectively where σ is the distance between two neighboring hypothesized planes. Table 2 reports the comparison results (both D_{ℓ_2} and D_{ℓ_1}), which show that the full version of P-MVSNet achieves significantly lower *mean absolute depth errors* and better *prediction precisions* than its two model variants and MVSNet. This study justifies the importance of the two core modules in P-MVSNet.

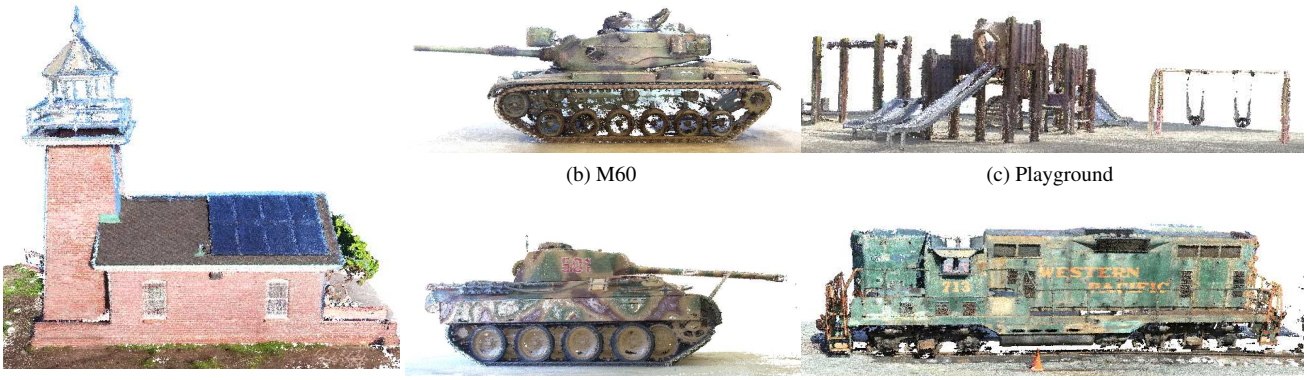
5.4. Comparisons with existing methods

First we evaluate and compare the quality of the fused 3D point clouds of the DTU evaluation scenes (22 models) produced by our P-MVSNet with some existing state-of-the-art methods including *camp* [4], *furu* [10], *tola* [40], *Gipuma* [11], *SurfaceNet* [17], and *MVSNet* [43]. The images are all cropped to the same size of $H = 1184$ and $W = 1600$, the number of adjacent images and hypothesized planes are set be 4 and 256 respectively, and for all images, the depth hypotheses are uniformly sampled from $D_{\min} = 425mm$ to $D_{\max} = 935mm$. The D_{ℓ_1} depth maps are used to reconstruct the point-cloud models. We use the evaluation protocol provided by the authors of the dataset, namely, we calculate the mean errors of the reconstruction *accuracy* and *completeness*, and the *overall* errors which is the average of the former two. The *accuracy* is measured as the distance from the reconstructed point cloud to the ground truth, while the *completeness* is defined as the distance from the ground truth to the reconstructed point cloud. Therefore, the lower the values of the three metrics, the better the reconstruction quality. Table 3 reports the evaluation results, and it is observed that while *Gipuma* achieves the highest *accuracy*, P-MVSNet performs the best in *completeness* and *overall*. Figure 6 shows a qualitative comparison of the reconstructed point clouds for the model *scan* 77 by different methods. In low-textured and reflected regions which are difficult to reconstruct, P-MVSNet produces the most complete point clouds.

Next we demonstrate the generalization ability of P-MVSNet (trained on DTU) by testing it on the *Tanks & Temples* dataset. The camera parameters of input images are estimated by the revised COLMAP [36]. We use the origin images to recover the camera models and sparse point

Table 4: Performance results on the *Tanks & Temples* dataset by different methods (as of February 19, 2019). Note: the top algorithms and some classic conventional methods are shown for comparison.

Method	Rank	Mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
P-MVSNet	**2.12**	55.62	70.04	44.64	40.22	**65.20**	**55.08**	**55.17**	**60.37**	**54.29**
Altizure-HKUST [3]	2.38	**56.22**	**74.60**	**61.30**	38.48	61.48	54.93	53.32	56.21	49.47
ACMH [42]	2.75	54.82	69.99	49.45	**45.12**	59.04	52.64	52.37	58.34	51.61
Dense R-MVSNet [44]	7.38	50.55	73.01	54.46	43.42	43.88	46.80	46.69	50.87	45.25
R-MVSNet [44]	7.75	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
MVSNet [43]	10.62	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
Pix4D [34]	11.12	43.24	64.45	31.91	26.43	54.41	50.58	35.37	47.78	34.96
COLMAP [36, 37]	12.25	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
OpenMVG [30] + OpenMVS [31]	13.38	41.71	58.86	32.59	26.25	43.12	44.73	46.85	45.97	35.27
OpenMVG [30] + MVE [7]	18.62	38.00	49.91	28.19	20.75	43.35	44.51	44.76	36.58	35.95
OpenMVG-G [30] + OpenMVS [31]	23.38	22.86	56.50	29.63	21.69	6.55	39.54	28.48	0.00	0.53
OpenMVG [30] + SMVS [23]	24.38	30.67	31.93	19.92	15.02	39.38	36.51	41.61	35.89	25.12
MVE [7]	25.00	25.37	48.59	23.84	12.70	5.07	39.62	38.16	5.81	29.19



(a) Lighthouse (b) M60 (c) Playground (d) Panther (e) Train

Figure 7: Some reconstructed point clouds from the *Tanks & Temples* dataset by P-MVSNet.

clouds, then obtain the undistorted images based on the estimated intrinsic parameters. In order to adapt to the input of the model, we crop all undistorted images to the size of $H = 1056$ and $W = 1920$, and the corresponding camera parameters are adjusted accordingly. The number of adjacent images and hypothesized planes of all scenes are set to be 4 and 256 respectively. The adjacent images and hypothesized planes of each reference image are determined according to the estimated camera poses and sparse point cloud. The D_{ℓ_1} depth maps are again used to reconstruct the 3D point-cloud models. The F-score is used as the evaluation metric, which can measure the accuracy and completeness of the reconstructed models simultaneously. The evaluation results are reported in Table 4. We can see that P-MVSNet achieves the state-of-the-art performance (5 best, 1 third, 1 fourth and 1 fifth scores out of 8 model scenes; the best *rank* and the second best *mean* measure) among all submissions (including many state-of-the-art learning-based or conventional MVS algorithms) according to the online leaderboard [22]. Some reconstructed 3D point clouds by P-MVSNet are shown in Figure 7 to demonstrate the quality of the reconstruction.

6. Conclusion

In this paper we have developed an effective end-to-end deep learning architecture of P-MVSNet for multi-view stereo. We have demonstrated that its outstanding reconstruction performance benefits from a series of novel modules proposed in P-MVSNet, e.g., the patch-wise confidence aggregation module to improve the matching accuracy and robustness, and the hybrid 3D U-Net to infer accurate depth maps. Extensive experimental results on the DTU sequences and *Tanks & Temples* benchmark datasets show that the proposed P-MVSNet clearly promotes the state-of-the-art performance for multi-view stereo over many existing learning-based or conventional methods. In future work, we will adapt our method to more complex scenes, e.g., the advanced *T2* sequences, the *ETH3D's high-res* dataset and aerial images, in which the memory load and computational cost for matching confidence aggregation and regularization are the main challenges to overcome. In addition, it would also be very interesting to combine the semantic label information with the proposed method to further improve the quality of multi-view reconstruction.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[3] Altizure. `https://www.altizure.com/`.

[4] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, pages 766–779. Springer, 2008.

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[6] R. T. Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, June 1996.

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

[8] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[9] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. *International Journal of Computer Vision*, 81(1):53–67, 2009.

[10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[11] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[12] Spyros Gidaris and Nikos Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5248–5257, 2017.

[13] Fatma Guney and Andreas Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015.

[14] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[15] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1595–1603. IEEE, 2017.

[16] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[17] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017.

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017.

[19] 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. *CoRR, vol. abs/1703.04309*, 2017.

[20] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[23] Fabian Langguth, Kalyan Sunkavalli, Sunil Hadap, and Michael Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016.

[24] Zhaoxin Li, Kuanquan Wang, Wangmeng Zuo, Deyu Meng, and Lei Zhang. Detail-preserving and content-aware variational multi-view stereo reconstruction. *IEEE Transactions on Image Processing*, 25(2):864–877, 2016.

[25] Alex Locher, Michal Perdoch, and Luc Van Gool. Progressive prioritized multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3244–3252, 2016.

[26] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[27] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[28] Yawei Luo, Zhedong Zheng, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Macro-micro adversarial network for human parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[29] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[30] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. Openmvg. an open multiple view geometry library. `https://github.com/openMVG/openMVG`.

[31] OpenMVS. open multi-view stereo reconstruction library. `https://github.com/cdcseacave/openMVS`.

[32] Ali Osman Ulusoy, Michael J Black, and Andreas Geiger. Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3280–3289, 2016.

[33] Despoina Paschalidou, Ali Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2018.

[34] Pix4D. `https://pix4d.com/`.

[35] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics & Autonomous Systems*, 56(11):927–941, 2008.

[36] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[37] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[38] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA*, pages 21–26, 2017.

[39] Sudipta N Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[40] 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.

[41] Ali Osman Ulusoy, Michael J Black, and Andreas Geiger. Semantic multi-view stereo: Jointly estimating objects and voxels. In *CVPR*, pages 4531–4540, 2017.

[42] Qingshan Xu and Wenbing Tao. Multi-view stereo with asymmetric checkerboard propagation and multi-hypothesis joint view selection. *CoRR*, abs/1805.07920, 2018.

[43] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.

[44] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *arXiv preprint arXiv:1902.10556*, 2019.

[45] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412. IEEE, 2018.

[46] Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Asian Conference on Computer Vision*, pages 166–175. Springer, 2007.

[47] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015.