International Journal of Computer Vision (2023) 131:199–214
https://doi.org/10.1007/s11263-022-01697-3

Vis-MVSNet: Visibility-Aware Multi-view Stereo Network

Jingyang Zhang[1] · Shiwei Li[2] · Zixin Luo[1] · Tian Fang[2] · Yao Yao[1]

Received: 20 August 2021 / Accepted: 26 September 2022 / Published online: 14 October 2022

Abstract
Learning-based multi-view stereo (MVS) methods have demonstrated promising results. However, very few existing networks explicitly take the pixel-wise visibility into consideration, resulting in erroneous cost aggregation from occluded pixels. In this paper, we explicitly infer and integrate the pixel-wise occlusion information in the MVS network via the matching uncertainty estimation. The pair-wise uncertainty map is jointly inferred with the pair-wise depth map, which is further used as weighting guidance during the multi-view cost volume fusion. As such, the adverse influence of occluded pixels is suppressed in the cost fusion. The proposed framework *Vis-MVSNet* significantly improves depth accuracy in reconstruction scenes with severe occlusion. Extensive experiments are performed on *DTU*, *BlendedMVS*, *Tanks and Temples* and *ETH3D* datasets to justify the effectiveness of the proposed framework.

Keywords Multi-view stereo · Visibility · MVSNet

1 Introduction

Multi-view stereo (MVS) is one of the core problems in computer vision, which is essential to a variety of applications including image-based 3D modeling, city-scale survey and autonomous driving. While the problem is mainly solved by classical methods (Campbell et al., 2008; Furukawa & Ponce, 2009; Tola et al., 2012; Galliani et al., 2015; Schönberger et al., 2016), recent learning-based methods (Yao et al., 2018, 2019; Gu et al., 2020) have also shown competitive results compared with previous state-of-the-arts. Learning-based methods usually extract deep image features from input images, which implicitly introduces global semantic such as specularity and reflection priors during the reconstruction process. Moreover, MVS networks usually apply 3D convolution neural networks (CNNs) for the cost volume regularization, which is more powerful than engineered cost regularization in classical methods.

One critical factor in MVS is the pixel-wise visibility: whether a 3D point is visible in given images. However, such visibility information is unknown before the 3D model is densely recovered, which implies a chicken-and-egg problem. In traditional MVS algorithms, the visibility issue is well understood: some approaches simply reject patch pairs according to pre-determined criteria, and then update the cost aggregation with only the inlier patch pairs (Furukawa & Ponce, 2009; Tola et al., 2012; Xu & Tao, 2019). More advanced approaches, such as COLMAP (Zheng et al., 2014; Schönberger et al., 2016), compute the visibility information and aggregate the pair-wise matching cost based on a probabilistic framework, where visibility and depth are alternatively updated in E-step and M-step.

However, very few of the current learning-based MVS methods have acknowledged this problem and have explicitly handled the visibility issue. For example, MVSNet and its following works (Yao et al., 2018, 2019; Chen et al., 2019; Gu et al., 2020; Cheng et al., 2020; Yang et al., 2020) feed multi-view features from all views into a variance-based cost metric regardless of the visibility of the pixel. Other methods

Communicated by William Smith.

✉ Yao Yao
yyaoag@cse.ust.hk

Jingyang Zhang
jzhangbs@cse.ust.hk

Shiwei Li
sli@altizure.com

Zixin Luo
zluoag@cse.ust.hk

Tian Fang
fangtian@altizure.com

[1] The Hong Kong University of Science and Technology, Sai Kung District, Hong Kong

[2] Everest Innovation Technology, Kowloon, Hong Kong

apply either averaging (Hartmann et al., 2017) or max pooling (Huang et al., 2018) to aggregate the matching cost. While it is possible that the network could implicitly learn how to discard the invisible views for each pixel, the unsolved visibility problem may inevitably deteriorate the final reconstruction.

In this work, we present an end-to-end network architecture that takes pixel-wise visibility information into account. The depth map is estimated from multi-view images in a two-step manner. First, matching is performed for each reference-source image pair and a latent volume representing the pair-wise matching quality is obtained. This volume further regresses to an intermediate estimation of a depth map and an uncertainty map, where the uncertainty is transformed from the depth-wise entropy of the probability volume. Second, to attenuate unmatchable pixels, we fuse all pair-wise latent volumes to one multi-view cost volume by using pair-wise matching uncertainties as weighting guidance. The fused volume is regularized and regresses to the final depth estimation. We also integrate several practical components from recent MVS networks, including group-wise correlation and Guo et al. (2019) coarse-to-fine strategy (Gu et al., 2020) to further boost the overall reconstruction quality. Our network is end-to-end trainable and the uncertainty part is trained in an unsupervised manner. In this case, we can directly utilize existing MVS datasets with only ground truth depth maps to train the visibility-aware MVS network.

The proposed Vis-MVSNet is evaluated on *DTU* (Jensen et al., 2014), *BlendedMVS* (Yao et al., 2020) datasets, and is benchmarked on *Tanks and Temples* (Knapitsch et al., 2017) and ETH3D (Schops et al., 2017) datasets. Our method ranks 1^{st} among all submissions in the *Tanks and Temples* online benchmark (until May 1, 2020) and is the top-tier learning-based method on both *Tanks and Temples* advanced set and *ETH3D* high-res set. Comparisons with previous methods and ablation studies in the experiment section demonstrate the significant improvement bought by our approach, especially when the occlusion problem is severe in input images. An example can be found in Fig. 1.

This paper extends Zhang et al. (2020a) with the following contents:

- Detailed explanation of depth map filtering and fusion (Sect. 3.8).
- Benchmarking on the large-scale *Tanks and Temples* advanced set (Sect. 4.2).
- Benchmarking on the *ETH3D* high-res test dataset (Sect. 4.3).

2 Related Work

Multi-view Stereo Multi-view stereo reconstructs surfaces from multiple images mainly by checking the consistency of the image projections. A detailed review can be found in Seitz et al. (2006). For example, Lhuillier and Quan (2005) and Furukawa and Ponce (2009) directly produce point clouds by iteratively propagating and densifying the points. If the scene is represented by voxel grid, space carving (Kutulakos & Seitz, 2000; Slabaugh et al., 2004; Furukawa & Ponce, 2006) can be applied to gradually discard non-photo consistent voxels. Alternatively, a given surface can be further refined by optimizing towards a more photo consistent solution (Grum & Bors, 2014). However, when processing large scale scenes, whole scene reconstruction often suffers from huge memory consumption and long processing time. In contrast, other methods (Tola et al., 2012; Campbell et al., 2008; Galliani et al., 2015; Schönberger et al., 2016; Yao et al., 2017) simplify the problem by only considering the surface inside the camera frustum of a reference view and checking photo consistency with only neighboring views. And the reconstruction of the whole scene is obtained by fusing the resulting depth maps of each view. In this paper, we follow the latter strategy.

Learning-based MVS Learning-based methods have shown great potentials to replace each step in traditional MVS reconstructions. The learnable multi-view cost metric (Hartmann et al., 2017) is first proposed to measure the multi-view photo-consistency between image patches. Later, SurfaceNet (Ji et al., 2017) is proposed to learn the cost volume regularization from geometry ground truth. The authors of LSM (Kar et al., 2017) apply the differentiable projection in the network and propose the first end-to-end learnable network for low-resolution MVS reconstruction. DeepMVS (Huang et al., 2018) reprojects images to 3D plane-sweeping volumes, performs intra-volume aggregation, and applies inter-volume aggregation to fuse the volumes and generate the depth map output. RayNet (Paschalidou et al., 2018) encodes the camera projection to the network, and utilizes the Markov Random Field to predict the surface label.

Another recent popular network for MVS reconstruction is MVSNet (Yao et al., 2018). MVSNet first extracts deep image features and then warps these features into the reference camera frustum to build a cost volume via differentiable homographies. To reduce the memory consumption during the network inference, the follow-up R-MVSNet (Yao et al., 2019) replaces the 3D CNNs regularization module with a 2D GRU recurrent network. Point-MVSNet (Chen et al., 2019) proposes a point-based depth map refinement network to improve the output accuracy and MVS-CRF (Xue et al., 2019) introduces the conditional random field optimization during the depth map estimation. More recently, CasMVSNet (Gu et al., 2020), CVP-MVSNet (Yang et al., 2020) and UCSNet (Cheng et al., 2020) integrate the coarse-to-fine strategy to the learning-based MVS reconstruction. These works preserve an image feature pyramid and generate an initial depth estimation with large depth interval at a low reso-

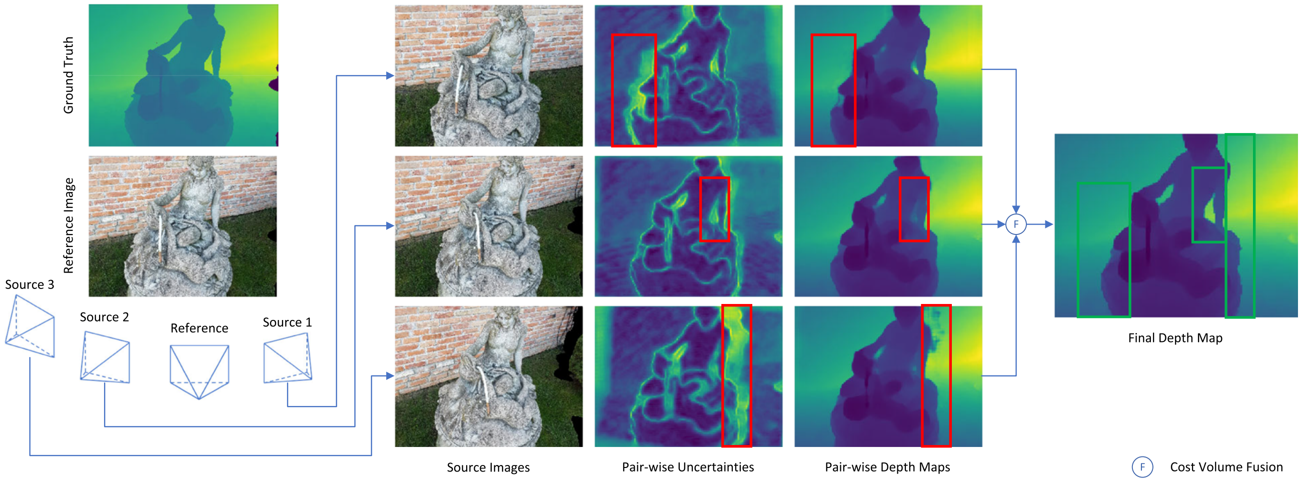


Fig. 1 Illustration of the visibility-aware fusion. For each reference-source pair, the uncertainty map successfully estimates per pixel visibility. During the fusion, occluded pixels are attenuated, resulting in a well reconstructed final depth map

lution. In following stages, cost volumes are constructed with a narrow depth range centering at the depth estimation from previous stages. The coarse-to-fine architecture successfully reduces memory consumption so that they support deeper backbone networks and higher resolution outputs. However, these methods all apply a variance-based cost metric, which is under the assumption that a given pixel is visible in all input images. As a result, an increasing number of input images would lead to even a worse depth map estimation quality.

Visibility Estimation Visibility estimation is a well-acknowledged problem in classic MVS reconstructions. Previous works include heuristic cost thresholding methods (Furukawa & Ponce, 2009; Tola et al., 2012; Xu & Tao, 2019) and more complicated joint depth-visibility estimation methods (Zheng et al., 2014; Schönberger et al., 2016). For latter approaches, the per-pixel visibility is usually jointly recovered during the depth map estimation process through an EM-based method. However, these methods apply a probabilistic framework which is hard to be directly integrated with deep neural networks. To handle the visibility issue in the learning-based frameworks, we should consider other alternatives for joint depth map and visibility estimation.

Current deep learning methods take visibility into account in an implicit manner. MVSNet (Yao et al., 2018) reduces the feature volumes from different source views by variance metric which considers each view equally and claims that information from invisible pixels can be filtered out in the regularization. Such implicit method heavily relies on the regularization of the neural network. Besides, DeepMVS (Huang et al., 2018) applies max pooling of multiple feature volumes to select the best latent representation, which is expected to be generated from a matchable pair. However, the fused volume is only related to the information from the best view, which loses the advantage of MVS that a more robust prediction can be produced by multiple observation. Instead, we start from pair-wise cost volumes to identify the pair-wise matching quality, and fuse the pair-wise volumes by weighted sum where weights of unmatchable pairs are reduced.

Uncertainty Estimation In our approach, visibility is indicated by the matching uncertainty of the pair-wise depth map. Uncertainty (or confidence) estimation for two-view depth or disparity estimation has been widely studied for classic methods by Hu and Mordohai (2012). The majority of such methods examine the properties of the probability distribution over all the depth or disparity hypotheses. End-to-end deep neural networks (Poggi and Mattoccia, 2016; Kim et al., 2018, 2019; Tosietal., 2018) are also applied to estimate the uncertainty map for two-view stereo. Recently, Kendall and Gal (2017) propose to jointly estimate the network output and its uncertainty based on the Bayesian neural network. However, this method cannot be directly adopted in our framework because they operate on 2D outputs, while we believe that it is more reasonable to estimate uncertainty from the 3D probability volume. Therefore we follow (Zhang et al., 2020b) to use the depth-wise entropy of the probability volume to explicitly measure the pair-wise matching uncertainty.

3 Method

3.1 Overview

Our baseline architecture is similar to CasMVSNet (Gu et al., 2020), where we apply a coarse-to-fine strategy for multi-view depth map estimation. The outline of the visibility-aware MVS network is illustrated in Fig. 2. Given

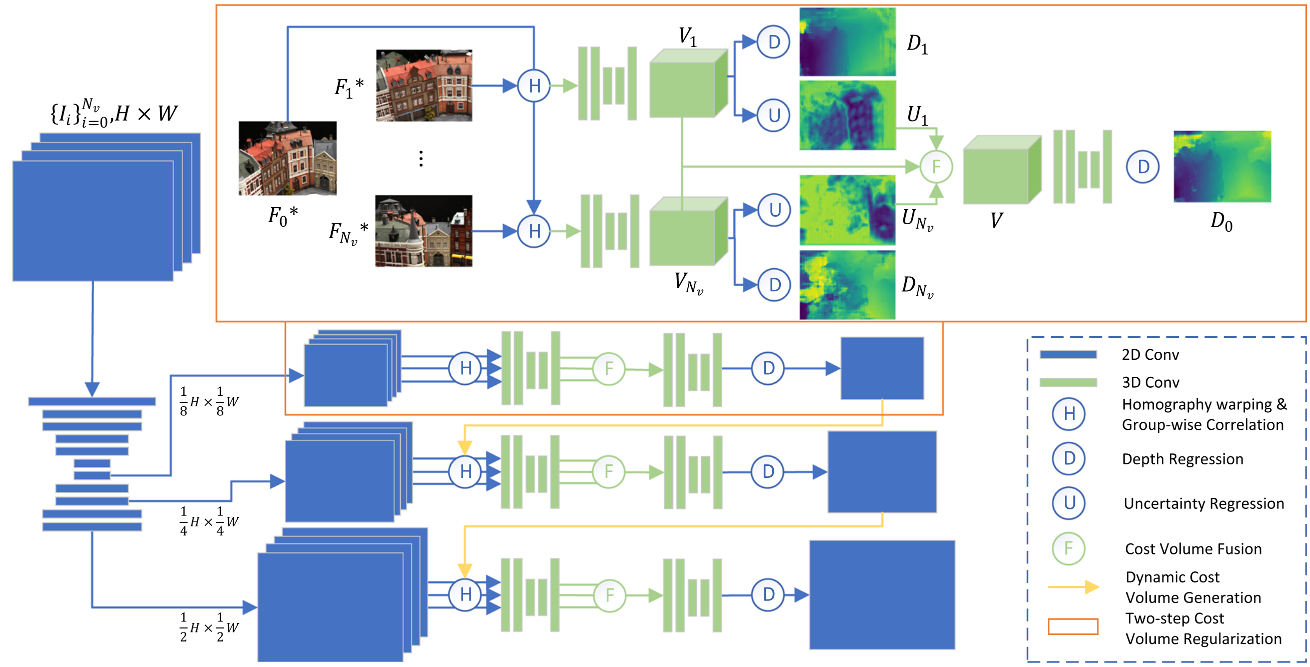

$\{I_i\}_{i=0}^{N_v}, H \times W$
F_1*
F_0*
$F_{N_v}*$
H
H
V_1
D
U
D_1
U_1
F
V
D
D_0
U
D
U_{N_v}
V_{N_v}
D_{N_v}
$\frac{1}{8}H \times \frac{1}{8}W$
$\frac{1}{4}H \times \frac{1}{4}W$
$\frac{1}{2}H \times \frac{1}{2}W$
H
F
D
2D Conv
3D Conv
H Homography warping & Group-wise Correlation
D Depth Regression
U Uncertainty Regression
F Cost Volume Fusion
Dynamic Cost Volume Generation
Two-step Cost Volume Regularization

Fig. 2 The proposed framework. For every reference-source pair, we jointly infer the depth map and the uncertainty map. The latent volumes are fused according to the uncertainty. And the fused volume is further regularized for the final depth map regression. $*F_0$, F_1 and F_{N_v} are he feature maps. The images here only show the original image of the feature maps

a reference image $\mathbf{I}_0$ and a set of neighboring source images $\{\mathbf{I}_i\}_{i=1}^{N_v}$, the framework predicts a reference depth map $\mathbf{D}_0$ aligned with $\mathbf{I}_0$. First, all images are fed into a 2D UNet (Ronneberger et al., 2015) for the extraction of multi-scale image features, which are used for depth estimation in three stages from low to high resolutions. For the reconstruction at the k-th stage, a cost volume is constructed, regularized and used to estimate a depth map $\mathbf{D}_{k,0}$ with the same resolution to the input feature map. Intermediate depth maps from previous stages will be used for the cost volume construction at next stages. Finally, $\mathbf{D}_{3,0}$ will be served as the final output $\mathbf{D}_0$ of the system. Detailed network architectures of each sub-networks are listed in Table 1.

3.2 Feature Extraction

In our network, deep image features are extracted by an hourglass-shaped encoder-decoder UNet (Ronneberger et al., 2015) architecture. The encoder generates a feature pyramid of input images, and the number of scales of the pyramid is equal to the length of the array F_{enc}. In each level of scale, the feature map from previous layer is fed into a downsizing residual block and $N_{enc} - 1$ ordinary residual blocks (He et al., 2016). The numbers of channels in each level are listed in the array F_{enc} in canonical order.

The decode upsamples feature maps back to the original size as an inverse pyramid. In each level of scale, the feature map is fed into a transposed convolutional layer with stride 2. The result is then concatenated with the feature map that has the same size in the encoder pyramid along the channel dimension. The concatenated feature map is further processed by N_{dec} residual blocks. The numbers of channels in each level are listed in the array F_{dec} in canonical order.

The extracted features at the last three scales in the decoder part are further converted to 32 channels by additional convolutions. These 32-channel feature maps are used to construct cost volumes at different resolutions.

3.3 Cost Volume and Regularization

We construct cost volumes at different scale stages. In the k-th scale stage, instead of directly constructing a unified cost volume from all views, we first construct pair-wise cost volumes for each reference-source pairs. For the i-th pair, by assuming that the reference image has depth d, we can obtain a warped feature map $\mathbf{F}_{k,i\rightarrow 0}(d)$ from the source view. Inspired by Guo et al. (2019), we apply the group-wise correlation to calculate a cost map between the reference and the warped source feature map. Specifically, given two 32-channel feature maps, we divide all the channels into 8 groups each with 4 channels. Then correlations are computed between each corresponding pair of group, resulting in 8 values for each pixel. Then the cost maps for all the depth hypothesis are stacked together as the cost volume. The resulting cost volume $\mathbf{C}_{k,i}$ of the i-th image pair in the k-th stage is of size $N_{d,k} \times H \times W \times N_c$, where $N_{d,k}$ is the depth hypothesis number in the k-th stage and $N_c = 8$ is the group number of the group-wise correlation operation. The set of the hypotheses is predetermined for the first stage, and is dynamically determined for the second and third stages according to the depth map output of the previous stage. The calculation of the dynamic depth range will be explained in Sect. 3.6.

As mentioned in Sect. 1, our cost regularization is performed in a two-step manner. First, every pair-wise cost volume is regularized to a latent volume $\mathbf{V}_{k,i}$ separately. Then, all latent volumes are fused to $\mathbf{V}_k$ which is further regularized to probability volume $\mathbf{P}_k$ and regresses to the final depth map of the current stage $\mathbf{D}_{k,0}$ via *soft-argmax* (Kendall et al., 2017) operation. The fusion of the latent volumes is visibility-aware. Concretely, we first measure visibility by jointly inferring pair-wise depth and uncertainty. Each latent volume is transformed to a probability volume $\mathbf{P}_{k,i}$ through additional 3D CNNs and the *softmax* operation. Then, the depth map $\mathbf{D}_{k,i}$ and the corresponding uncertainty map $\mathbf{U}_{k,i}$ are jointly inferred via *soft-argmax* and *entropy* operation, which will be explained in Sect. 3.4. The uncertainty map will be used as a weighting guidance during latent volume fusion (Sect. 3.5).

3.4 Pair-wise Joint Depth and Uncertainty Estimation

As stated in the previous section, pair-wise probability volumes are obtained for joint depth and uncertainty estimation. Similar to other current learning-based MVS methods, the depth map is regressed from the probability volume via the *soft-argmax* operation. For simplicity, the stage number k is omitted below. We denote the probability distribution over all the depth hypotheses as $\{\mathbf{P}_{i,j}\}_{j=1}^{N_d}$. The *soft-argmax* operation is equivalent to computing the expectation of this distribution and $\mathbf{D}_i$ is computed as:

$$\mathbf{D}_i = \sum_{j=1}^{N_d} d_j \mathbf{P}_{i,j} \tag{1}$$

To jointly regress the depth estimation and its uncertainty, we assume that the depth estimation follows the Laplacian distribution (Kendall & Gal, 2017). In this case, the estimated depth and the uncertainty maximize the likelihood of the observed ground truth:

$$p(\mathbf{D}_{gt,i}|\mathbf{D}_i, \mathbf{U}_i) = \frac{1}{2\mathbf{U}_i} \cdot \exp\left(\frac{|\mathbf{D}_i - \mathbf{D}_{gt,i}|}{\mathbf{U}_i}\right) \tag{2}$$

where U_i is pixel-wise uncertainty of the depth estimation. Notice that the probability distribution $\{\mathbf{P}_{i,j}\}_{j=1}^{N_d}$ also reflects the matching quality. We thus apply the entropy map $\mathbf{H}_i$ of $\{\mathbf{P}_{i,j}\}_{j=1}^{N_d}$ to measure the depth estimation quality. And the uncertainty map $\mathbf{U}_i$ is transformed from $\mathbf{H}_i$ by a function f_u, which is presented as a shallow 2D CNN in the network:

$$\mathbf{U}_i = f_u\left(\mathbf{H}_i) = f_u(\sum_{j=1}^{N_d} -\mathbf{P}_{i,j} \log \mathbf{P}_{i,j}\right) \tag{3}$$

The reason of adopting the entropy is that the randomness of the distribution is negatively related to the uni-modal distribution. And the uni-modality is an indicator of high confidence of the depth estimation.

To jointly learn the depth map estimation $\mathbf{D}_i$ and its uncertainty $\mathbf{U}_i$, we minimize the negative log likelihood described above:

$$\begin{aligned} L_i^{joint} &= \frac{1}{|I_0^{valid}|} \sum_{x \in I_0^{valid}} -\log\left(\frac{1}{2\mathbf{U}_i} \exp \frac{|\mathbf{D}_i - \mathbf{D}_{gt,i}|}{\mathbf{U}_i}\right) \\ &= \frac{1}{|I_0^{valid}|} \sum_{x \in I_0^{valid}} \frac{1}{\mathbf{U}_i} |\mathbf{D}_i - \mathbf{D}_{gt,i}| + \log \mathbf{U}_i \end{aligned} \tag{4}$$

Constants are omitted in the formula. For numerical stability, in practice we infer $\mathbf{S}_i = \log \mathbf{U}_i$ instead of $\mathbf{U}_i$ directly. The log uncertainty map $\mathbf{S}_i$ is also transformed from the entropy map $\mathbf{H}_i$ by a shallow 2D CNN.

The loss (Eq. 4) can also be interpreted as attenuation to the L_1 loss between the estimation and the ground truth with a regularization term. The intuition is that the interference from the erroneous samples should be reduced during training.

3.5 Volume Fusion

In this section we introduce the visibility-aware volume fusion. For simplicity, the stage number k is omitted. Given the pair-wise latent cost volumes $\{\mathbf{V}_i\}_{i=1}^{N_v}$, a single volume $\mathbf{V}$ is fused from the volumes by weighted sum, where the weight is negatively related to the estimated pair-wise uncertainty.

$$\mathbf{V} = \left(\sum_{i=1}^{N_v} \frac{1}{\exp \mathbf{S}_i}\right)^{-1} \sum_{i=1}^{N_v} \left(\frac{1}{\exp \mathbf{S}_i} \mathbf{V}_i\right) \tag{5}$$

From our observations, pixels with large uncertainty are more likely to be located in occluded regions. Thus, these values in the latent volume could be attenuated.

An alternative to the weighted sum is applying threshold for $\mathbf{S}_i$ and perform a hard visibility selection for each pixel. However, without an interpretation of the value $\mathbf{S}_i$, we can only do empirical thresholding that may not be universal. Instead, our weighted sum formulation naturally fuses all views and considers the log uncertainty $\mathbf{S}_i$ in a relative manner.

Table 1 Detailed network architecture

Name	Layer	Output
input		$H \times W \times 3$
Feature Extraction		
feat-conv0	5x5 conv, stride=2	$1/2H \times 1/2W \times 16$
feat-UNet	$N_{enc} = 2$, $N_{dec} = 1$ $F_{enc} = [32, 64, 128]$, $F_{dec} = [64, 32]$	$1/2H \times 1/2W \times 32$
feat-out1	conv on $1/8$ scale, w/o BN, ReLU	$1/8H \times 1/8W \times 32$
feat-out2	conv on $1/4$ scale, w/o BN, ReLU	$1/4H \times 1/4W \times 32$
feat-out3	conv on $1/2$ scale, w/o BN, ReLU	$1/2H \times 1/2W \times 32$
Pair-wise Cost Volume		
cost-volume	Groupwise Correlation	$N_d \times H_k \times W_k \times 8$
Pair-wise Regularization		
reg0-UNet	$N_{enc} = 1$, $N_{dec} = 0$ $F_{enc} = [8, 16]$, $F_{dec} = [8]$	$N_d \times H_k \times W_k \times 8$
Pair-wise Depth and Uncertainty Estimation		
reg0-conv	3D conv w/o BN, ReLU	$N_d \times H_k \times W_k \times 1$
prob-volume	*softmax* along N_d	$N_d \times H_k \times W_k \times 1$
pair-depth	*soft argmax* along N_d on prob-volume	$(1\times)H_k \times W_k \times 1$
pair-entropy	*entropy* along N_d on prob-volume	$(1\times)H_k \times W_k \times 1$
uncert-res	residual block on pair-entropy	$H_k \times W_k \times 8$
uncertainty	conv w/o BN, ReLU on uncert-res	$H_k \times W_k \times 1$
Volume Fusion		
fused	weighted average on all reg0-UNet	$N_d \times H_k \times W_k \times 8$
Post-fusion Regularization		
reg1-UNet	$N_{enc} = 1$, $N_{dec} = 0$ $F_{enc} = [8, 16]$, $F_{dec} = [8]$	$N_d \times H_k \times W_k \times 8$
reg1-out	3D conv w/o BN, ReLU	$N_d \times H_k \times W_k \times 1$
Final Depth Estimation		
final-prob-vol	*softmax* along N_d	$N_d \times H_k \times W_k \times 1$
final-depth	*soft argmax* along N_d	$(1\times)H_k \times W_k \times 1$

All convolutions are without bias, with a kernel size of 3 and a stride of 1, and are followed by Batch Normalization and ReLU unless otherwise specified. H_k, W_k denote heights and weights at different scale stages

3.6 Coarse-to-Fine Architecture

Our coarse-to-fine architecture mainly follows the recent Cas-MVSNet (Gu et al., 2020). In all stages, depth hypothesis are uniformly sampled from a depth range. The first stage takes image features at low resolution and constructs cost volume with the predetermined depth range but with a larger depth interval, while the following stages use higher spatial resolutions, narrower depth ranges and smaller depth intervals.

For the first stage, the depth range is $[d_{min}, d_{min} + 2\Delta d)$ and the depth number is $N_{d,1}$, where d_{min}, Δd and $N_{d,1}$ is predetermined. For the k-th stage ($k \in \{2, 3\}$), the depth range, sample number and interval are reduced. And the ranges are centered at the depth estimation from the previous stage, which are different for each pixel. The depth range for pixel x is $[\mathbf{D}_{k-1,0} - w_k\Delta d, \mathbf{D}_{k-1,0} + w_k\Delta d)$ and the depth number is $p_k N_{d,k}$, where $w_k < 1$ and $p_k < 1$ are the predefined scaling factors, and $\mathbf{D}_{k-1,0}$ is the final depth estimation of pixel x from the last stage $k - 1$.

3.7 Training Loss

For each stage, we compute the pair-wise L_1 loss, the pair-wise joint loss and the L_1 loss of the final depth map. The total loss is the weighted sum of the losses from all three stages. To normalize the scale in different training scenes, all depth differences are divided by the pre-defined depth interval of the final stage.

$$L = \sum_{k=1}^{3} \lambda_k \left[L_{1,k}^{final} + \frac{1}{N_v} \sum_{i=1}^{N_v} (L_{1,k,i}^{pair} + L_{k,i}^{joint}) \right] \tag{6}$$

The pair-wise L_1 losses are included because the uncertainty loss tends to over-relax the pair-wise depth and uncertainty estimation. The pair-wise L_1 losses here could guarantee a qualified pair-wise depth map estimation.

3.8 Point Cloud Generation

After generating depth maps of all views, the final step of the proposed method is to fuse all depth maps into a unified point cloud. Following Yao et al. (2018), depth maps are filtered and fused to ensure both photometric and geometric consistency from different views.

Probability Maps Because *soft-argmax* operation can always generate a final estimation despite the quality of the probability distribution, we additionally generate probability maps to filter out unreliable pixels. The total probabilities of depth hypothesis within range $[\mathbf{D}-2, \mathbf{D}+2]$, *i.e*let@tokeneonedothypothesis around the final estimation, are calculated as the probability map of a given depth map output. Moreover, in our coarse-to-fine architecture, we consider all probability maps at different stages, and the filtering criterion is that a pixel in a reference view will be preserved if and only if all probability maps from all three stages are higher than the corresponding thresholds $p_{t,1}, p_{t,2}, p_{t,3}$.

Geometric Consistency For a pixel $\mathbf{p}_r$ in a reference depth map $\mathbf{D}_r$, we can obtain its reprojected pixel $\mathbf{p}_{reproj}$ and depth d_{reproj} from each source view by following steps: 1) back-project the pixel with depth $\mathbf{D}_r(\mathbf{p}_r)$ to space. 2) project the space point to the source depth map $\mathbf{D}_s$ as $\mathbf{p}_{r\to s}$. 3) back-project $\mathbf{D}_s(\mathbf{p}_{r\to s})$ to space. 4) project the second space point to reference view as $\mathbf{p}_{reproj}$ with depth d_{reproj}. We consider the estimated $\mathbf{D}_r(\mathbf{p}_r)$ is geometrically consistent for this source view if and only if $\|\mathbf{p}_r - \mathbf{p}_{reproj}\| < 1$ and $\frac{|\mathbf{D}_r(\mathbf{p}_r)-d_{reproj}|}{\max(\mathbf{D}_r(\mathbf{p}_r), d_{reproj})} < 1\%$. During the filtering, pixels with less than N_f consistent views are discarded.

Geometric Visibility Fusion We follow the visibility-based depth map fusion in Merrell et al. (2007). All the source depth maps are projected to the reference view, where each pixel in the reference depth map may receive different number of depth values. For each pixel, we calculate the following metrics for each depth: (1) occlusion, which is the number of depths occluding this one (depth value is smaller than this one); (2) violation, which is the number of views in which this depth will be in the free-space after projection (projected depth is smaller than the value at the corresponding location in source views); (3) stability, which is occlusion minus violation. Finally the smallest depth value with non-negative stability is selected as the new depth value of this pixel. More details of occlusion, violation and stability can be found in the original paper (Merrell et al., 2007). Compared with simply taking the median of depth candidates, we observe that the visibility-based fusion slightly improves the point cloud quality.

Geometric Average Fusion The noise of the estimated depth values can be reduced by averaging the reprojected depths from source views. For a pixel $\mathbf{p}_0$ in a reference depth map with depth d_0, we gather reprojected depths $\{d_i\}_{i\in\mathbf{I}_c}$ from all the consistent source views $\mathbf{I}_c$. The depth from average fusion is $(d_0 + \sum_{i\in\mathbf{I}_c} d_i)/(|\mathbf{I}_c|+1)$.

Table 2 Fusion parameters used for the experiments on each datasets

	N_v	N_f	Prob. Threshold		
			$p_{t,1}$	$p_{t,2}$	$p_{t,3}$
TnT Intermediate	7	4	0.8	0.7	0.8
TnT Advanced	20	3	0.3	0.4	0
ETH3D	20	2	0.1	0.1	0
DTU	5	2	0.6	0.6	0.6

Small Segment Filter Finally, we introduce the small segment filter in our pipeline. We observe that small clusters of flying points are usually noises in space. We can easily remove them according to their cluster size, which can be done in depth map level. Given a depth map, a graph can be built where there is an edge between two adjacent pixels if both pixels are valid and the depth difference is not large. Then we remove the connected components with small number of pixels. In practice, we use the threshold of depth difference percentage as 0.05% and cluster size as 10.

Fusion Pipeline The whole filtering and fusion pipeline is listed as follows: (1) Probability map filtering; (2) Geometric consistency filtering; (3) Geometric visibility fusion; (4) Geometric consistency filtering; (5) Geometric average fusion; (6) Geometric consistency filtering; (7) Small segment filtering. If a pixel is filtered out, it will be excluded in all the following steps. An illustration of the filtering and fusion pipeline can be found in Fig. 3. The fusion parameters used for each dataset is listed in Table 3.

4 Experiment

4.1 Implementation

Pre-selection of Source Images The considered source views should be well-conditioned in the sense that the union of the overlapped areas of the reference-source pairs can cover most of the reference image. Therefore, all the source views are sorted according to the score considering all the baseline angle of the common tracks in the sparse reconstruction. Generally, the source views that are close to the reference view are preferred.

The selection criteria follows Yao et al. (2018). We use sparse tracks from the structure from motion (SfM) step to rate the closeness of each image pair. For a view pair i, j, we calculate a score $s(i, j) = \sum_{\mathbf{p}} \mathcal{G}(\theta_{i,j}(\mathbf{p}))$ that considers the baseline angle $\theta_{i,j}(\mathbf{p})$ of all the common tracks $\mathbf{p}$ in this view pair. The baseline angle can be derived as $\theta_{i,j}(\mathbf{p}) = (180/\pi)\arccos[(\mathbf{c}_i - \mathbf{p})\cdot(\mathbf{c}_j - \mathbf{p})]$, where $\mathbf{c}$ is the camera

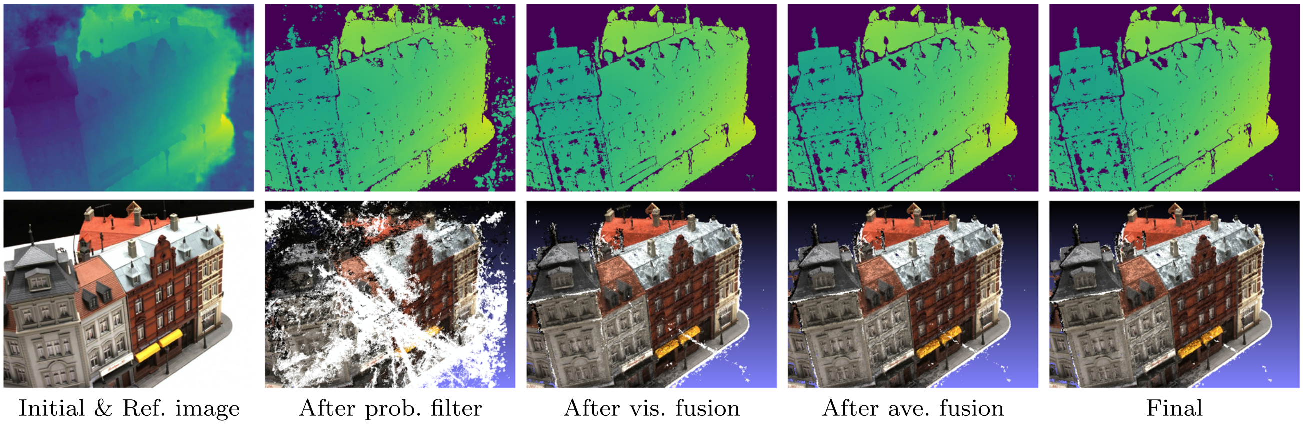


Fig. 3 Illustration of intermediate results during depth map filter and fusion (Sect. 3.8) on *scan9* of *DTU* dataset

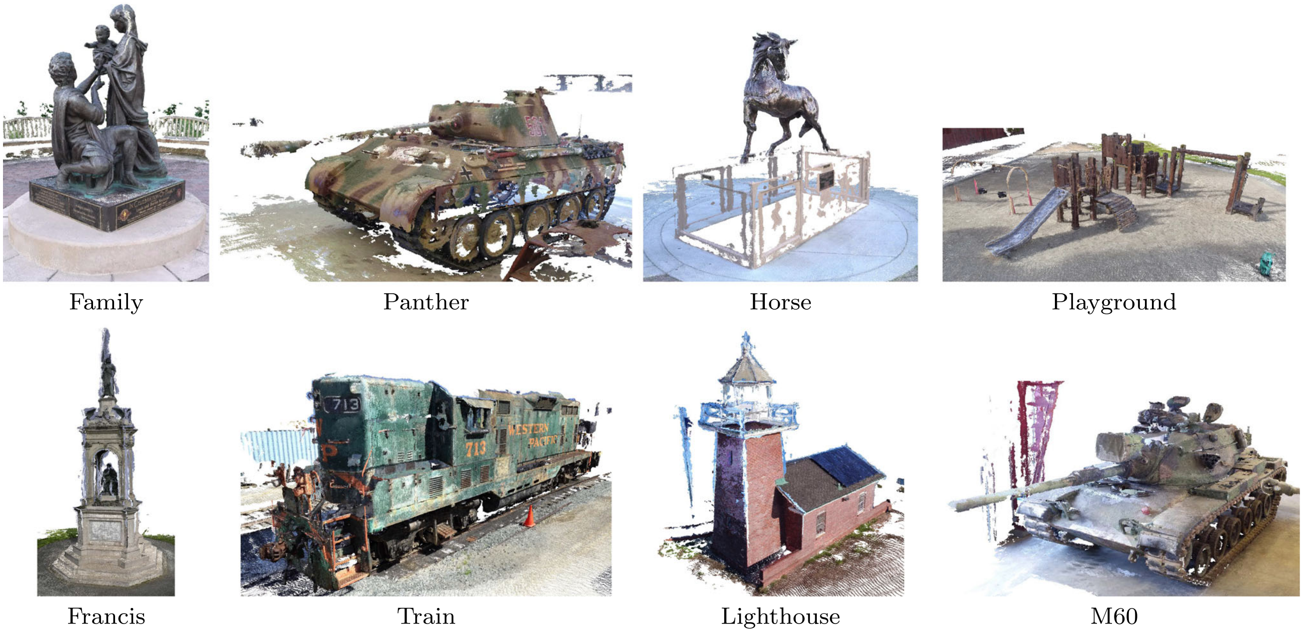


Fig. 4 Qualitative results of point clouds on the *intermediate set* of *Tanks and Temples*

center. $\mathcal{G}(\theta)$ is a piecewise Gaussian function (Zhang et al., 2015):

$$\mathcal{G}(\theta) = \begin{cases} \exp(-\frac{(\theta-\theta_0)^2}{2\sigma_1^2}), & \theta \leq \theta_0 \\ \exp(-\frac{(\theta-\theta_0)^2}{2\sigma_2^2}), & \theta > \theta_0 \end{cases} \tag{7}$$

In all the experiments, $\theta_0 = 5$, $\sigma_1 = 1$ and $\sigma_2 = 10$.

Training Our network is trained on *BlendedMVS* (Yao et al., 2020) training set for most experiments (Sects. 4.2 and 4.5) and is trained on DTU training set (Jensen et al., 2014) for DTU benchmarking (Sect. 4.4). For both training sets, we use the input image size of 640 × 512 and output depth map size of 320 × 256. We set the number of source views to $N_v = 3$ during training. For depth samples at different stages, we set the depth hypothesis numbers to $N_{d,1}, N_{d,2}, N_{d,3} = 32, 16, 8$, and depth range scaling factors to $w_2, w_3 = \frac{1}{4}, \frac{1}{16}$ respectively. The loss weights for each stage $\lambda_1, \lambda_2, \lambda_3 = 0.5, 1, 2$. The network is trained for 160k iterations with a batch size of 2 by an Adam (Kingma & Ba, 2014) optimizer. The initial learning rate is 0.001 and is halved at the 100, 120 and 140k steps. All experiments are performed using one NVidia V100 GPU.

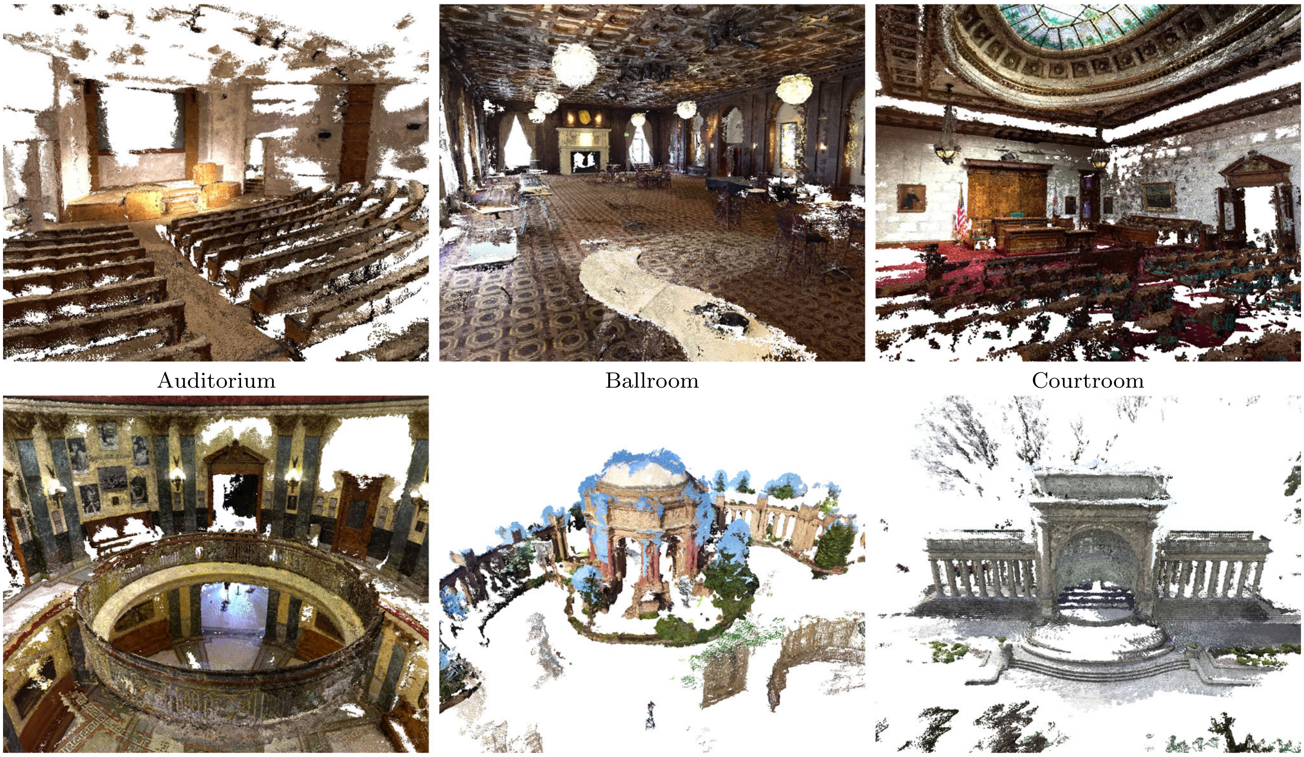
4.2 Benchmarking on Tanks and Temples Dataset

We first evaluate our method on the *Tanks and Temples* dataset (Knapitsch et al., 2017). As mentioned in Sect. 4.1, we use the *BlendedMVS* training set (Yao et al., 2020) to train the network. *BlendedMVS* is a recent MVS dataset contain-

Table 3 Quantitative results of point clouds on the *intermediate set* of *Tanks and Temples*

		Mean		Family	Francis	Horse	Light.	M60	Panther	Play.	Train
	Precision	Recall	F-score ↑								
COLMAP Schönberger et al. (2016)	43.16	44.48	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
MVSNet Yao et al. (2018)	40.23	49.70	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
Point-MVSNet Chen et al. (2019)	41.27	60.13	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
SurfaceNetPlus Ji et al. (2020)	51.86	50.30	49.38	62.38	32.35	29.35	62.86	54.77	54.14	56.13	43.10
R-MVSNet Yao et al. (2019)	39.80	71.96	50.55	73.01	54.46	43.42	43.88	46.80	46.69	50.87	45.25
PatchmatchNet Wang et al. (2021)	43.64	69.37	53.15	66.99	52.64	43.24	54.87	52.87	49.54	54.21	50.81
CVP-MVSNet Yang et al. (2020)	51.41	60.19	54.03	76.50	47.74	36.34	55.12	57.28	54.28	57.43	47.54
UCSNet Cheng et al. (2020)	46.66	70.34	54.83	76.09	53.16	43.03	54.00	55.60	51.49	57.38	47.89
PCF-MVS Kuhn et al. (2019)	49.82	65.68	55.88	70.99	49.60	40.34	63.44	57.79	58.91	56.59	49.40
CasMVSNet Gu et al. (2020)	47.62	74.01	56.84	76.37	58.45	46.26	55.81	56.11	54.06	58.18	49.51
ACMM Xu and Tao (2019)	49.19	70.85	57.27	69.24	51.45	46.97	63.20	55.07	57.64	60.08	54.48
BP-MVSNet Sormann et al. (2020)	51.26	68.77	57.60	77.31	**60.90**	47.89	58.26	56.00	51.54	58.47	50.41
ACMP Xu and Tao (2020)	49.06	73.58	58.41	70.30	54.06	**54.11**	61.65	54.16	57.60	58.12	57.25
D2HC-RMVSNet Yan et al. (2020)	49.88	**74.08**	59.20	74.69	56.04	49.42	60.08	59.81	59.61	60.04	53.92
DeepC-MVS Kuhn et al. (2020)	**59.11**	61.21	59.79	71.91	54.08	42.29	**66.54**	55.77	**67.47**	60.47	**59.83**
Vis-MVSNet	54.44	70.48	**60.03**	**77.40**	60.23	47.07	63.44	**62.21**	57.28	**60.54**	52.07

Bold entries are the best results among the compared methods F-scores are shown for each scene. The proposed method achieves the best mean F-score among all published works



Auditorium

Ballroom

Courtroom

Museum

Palace

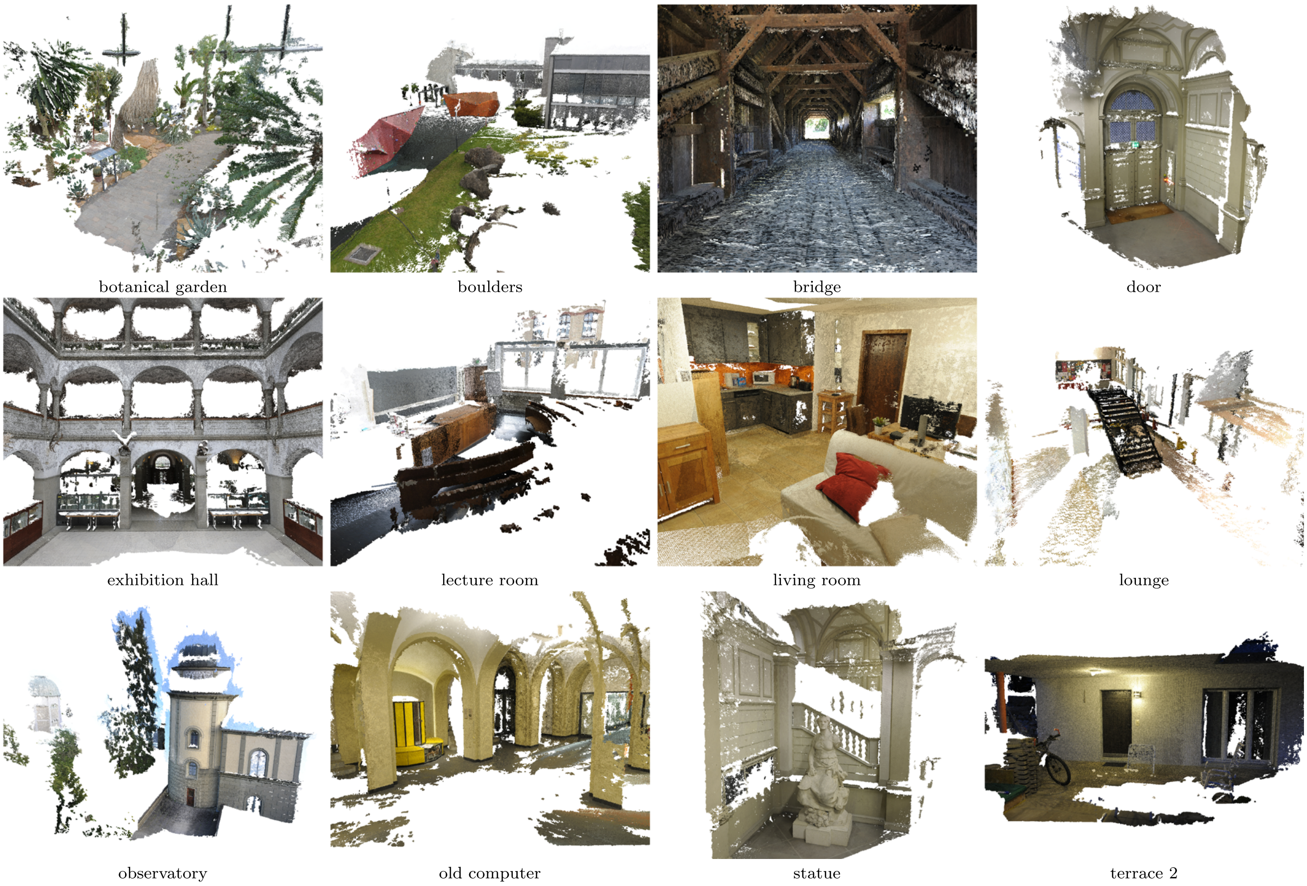
Temple

Fig. 5 Qualitative results of point clouds on the *advanced set* of *Tanks and Temples*

Table 4 Quantitative results of point clouds on the *advanced set* of *Tanks and Temples*

		Mean		Auditorium	Ballroom	Courtroom	Museum	Palace	Temple
	Precision	Recall	F-score ↑						
Colmap Schönberger et al. (2016)	33.65	23.96	27.24	16.02	25.23	34.70	41.51	18.05	27.94
R-MVSNet Yao et al. (2019)	28.03	33.63	29.55	19.49	31.45	29.99	42.31	22.94	31.10
CasMVSNet Gu et al. (2020)	29.68	35.24	31.12	19.81	38.46	29.10	43.87	27.36	28.11
BP-MVSNet Sormann et al. (2020)	29.62	35.61	31.35	20.44	35.87	29.63	43.33	27.93	30.91
PatchmatchNet Wang et al. (2021)	27.27	41.66	32.31	23.69	37.73	30.04	41.80	28.31	32.29
Vis-MVSNet	30.16	41.42	33.78	20.79	**38.77**	32.45	44.20	**28.73**	**37.70**
ACMM Xu and Tao (2019)	35.63	34.90	34.02	23.41	32.91	41.17	48.13	23.87	34.60
DeepC-MVS Kuhn et al. (2020)	**40.68**	31.30	34.54	26.30	34.66	43.50	45.66	23.09	34.00
PCF-MVS Kuhn et al. (2019)	37.52	35.36	35.69	28.33	38.64	35.95	48.36	26.17	36.69
ACMP Xu and Tao (2020)	34.57	**42.48**	**37.44**	**30.12**	34.68	**44.58**	**50.64**	27.20	37.43

Bold entries are the best results among the compared methods F-scores are shown for each scene. The proposed method outperforms other end-to-end learning-based methods


botanical garden

boulders

bridge

door

exhibition hall

lecture room

living room

lounge

observatory

old computer

statue

terrace 2

Fig. 6 Qualitative results of point clouds on the test set of *ETH3D*

Table 5 Quantitative result (F-score) of point clouds on the test set of *ETH3D*

	all ↑	Botani.	Boulde.	Bridge	Door	Exhibi.	Lectur.	Living.	Lounge	Observ.	Old co.	Statue	Terrac.
Colmap Schönberger et al. (2016)	73.01	87.13	65.63	88.3	84.19	62.96	63.80	87.69	38.04	92.56	46.66	74.91	84.24
PatchmatchNet Wang et al. (2021)	73.12	83.18	60.85	79.63	78.57	64.13	71.73	79.81	51.20	85.97	57.40	76.36	88.66
LTVRE_ROB Kuhn et al. (2017)	76.25	88.60	64.38	79.24	89.12	70.76	69.79	87.86	49.09	93.20	56.21	80.16	86.65
PLC Liao et al. (2019)	78.05	**91.88**	67.80	88.19	89.46	64.33	69.87	88.38	53.66	93.58	58.86	82.73	87.86
PCF-MVS Kuhn et al. (2019)	80.38	87.71	68.99	83.65	91.46	63.00	77.77	90.28	66.10	95.09	61.40	88.22	90.94
ACMM Xu and Tao (2019)	80.78	89.31	68.37	89.99	91.60	70.28	77.25	89.66	53.37	93.53	74.24	82.85	88.85
ACMP Xu and Tao (2020)	81.51	89.42	68.69	89.97	91.50	74.46	75.04	90.41	53.82	94.54	77.56	82.91	89.86
MAR-MVS Xu et al. (2020)	81.84	91.32	71.53	89.82	90.92	69.77	78.50	86.21	62.36	94.52	65.36	92.05	89.77
CLD-MVS Li et al. (2020)	82.31	88.47	67.73	85.81	92.35	75.78	78.33	86.58	64.01	94.35	76.05	87.48	90.79
Vis-MVSNet	83.46	90.25	68.74	90.30	90.13	76.33	80.87	92.89	61.00	94.20	72.45	90.97	93.35
DeepC-MVS Kuhn et al. (2020)	**87.08**	91.16	**72.32**	**90.31**	**93.94**	**77.12**	**82.82**	**94.23**	**72.13**	**97.09**	**84.83**	**95.37**	**93.65**

Bold entries are the best results among the compared methods The proposed method is the second best among all the learning-based methods

ing 113 indoor and outdoor scenes with 16904 MVS training samples in total. The dataset is split into 106 training scenes and 7 validation scenes. The trained model is directly applied to the *Tanks and Temples* benchmarking without fine-tuning.

Intermediate Set The *Tanks and Temples* intermediate set contains 8 scenes captured with outside-look-in camera trajectories. We use the original input image size of 1920 × 1080 for our evaluation. The source image number is set to $N_v = 7$ for network inference and we choose $N_f = 4$, $p_{t,1}, p_{t,2}, p_{t,3} = 0.8, 0.7, 0.8$ for depth map filter and fusion. Quantitative results are shown in Table 3 and corresponding point cloud reconstructions are illustrated in Fig. 4. Our Vis-MVSNet achieves a mean F-score of 60.03 and ranks 1^{st} among all methods in the benchmark (until May 1, 2020), which outperforms all classical MVS methods (Schönberger et al., 2016; Xu & Tao, 2019) and recent learning-based approaches (Yao et al., 2018; Chen et al., 2019; Yang et al., 2020; Cheng et al., 2020; Gu et al., 2020). Qualitatively, the points are dense and accurate in well-textured regions. The visibility handling mechanism improves the depth accuracy so that more points survive in the point cloud fusion, which improves both accuracy and recall. However, incompleteness and noise are observed in non-Lambertian and textureless regions such as the foundation of the statue and the edge between the objects and the sky.

Advanced Set The advanced set contains 2 outdoor scenes, as well as 4 indoor scenes captured with inside-look-out camera trajectories. We still use the original image resolution. Because the images are captured densely, we use $N_v = 20$ in order to enlarge the total overlap between reference image and source images. Also because the depth range is larger than the intermediate set, we double the depth sample numbers in all three stages. For depth map filter and fusion, we use $N_f = 3$, $p_{t,1}, p_{t,2}, p_{t,3} = 0.3, 0.4, 0$. Quantitative results are shown in Table 4 and corresponding point cloud reconstructions are illustrated in Fig. 5. Our Vis-MVSNet achieves a mean F-score of 33.78, which outperforms other end-to-end learning-based methods. Similar to the intermediate set, well-textured regions are well reconstructed. But there is still large incompleteness in the indoor scenes. One possible reason is that the depth range of the indoor scenes is much wider than the outdoor ones, which reduce the depth accuracy given fixed number of depth hypothesis.

4.3 Benchmarking on ETH3D Dataset

We further evaluate our method on the *ETH3D* dataset (Schops et al., 2017). *ETH3D* test set contains 12 scenes captured from both indoor and outdoor scenarios. Number of views varies from 10 to 100. Because of memory constraint, we downsize the input images to 2400 × 1600. The number of source views is still $N_v = 20$ but we additionally prune the views with selection scores smaller than 10% of the best score, which achieves a good balance between total overlap coverage and quality of the source views. For depth map filter and fusion, we use $N_f = 2$, $p_{t,1}, p_{t,2}, p_{t,3} = 0.1, 0.1, 0$. Quantitative results are shown in Table 5 and corresponding point cloud reconstructions are illustrated in Fig. 6. Qualitatively, the well and poorly reconstructed regions are similar to the Tanks and Temples scenes. Quantitatively, our Vis-MVSNet achieves a mean F-score of 83.46, which is the second best among all the learning-based methods. Also, it is noteworthy that the best performing method DeepC-MVS (Kuhn et al., 2020) is additionally trained on *ETH3D* training set, while our method directly uses the model pretrained on *BlendedMVS*.

4.4 Benchmarking on DTU Dataset

The proposed method is also benchmarked on the DTU evaluation set (Jensen et al., 2014). *DTU* dataset contains 128 scans under fixed camera trajectories and 7 sets of lighting

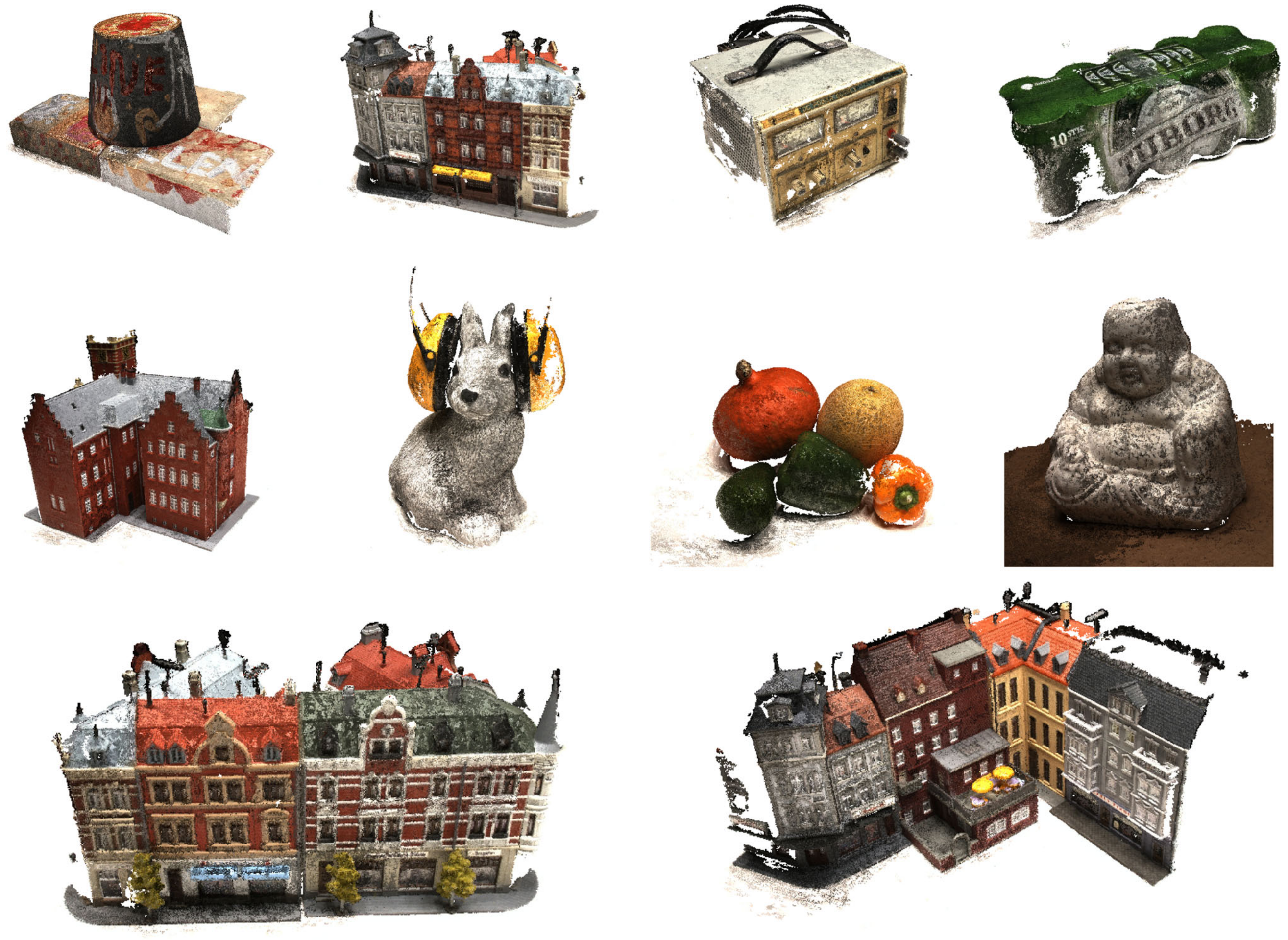


Fig. 7 Qualitative results of the point clouds on the *DTU* dataset

Table 6 Quantitative result of the point cloud on the test set of *DTU*

	Acc.	Comp.	Overall
COLMAP Schönberger et al. (2016)	0.400	0.664	0.532
MVSNet Yao et al. (2018)	0.396	0.527	0.462
Point-MVSNet Chen et al. (2019)	0.342	0.411	0.376
CVP-MVSNet Yang et al. (2020)	**0.296**	0.406	0.351
UCSNet Cheng et al. (2020)	0.338	**0.349**	**0.344**
CasMVSNet Gu et al. (2020)	0.325	0.385	0.355
Vis-MVSNet	0.369	0.361	0.365

Bold entries are the best results among the compared methods The proposed method achieves comparable overall distance (*mm*) on *DTU*

configuration. Every scan has 49 views with given camera parameters. As suggested by previous methods (Ji et al., 2017; Yao et al., 2018), DTU dataset is split into training set, validation set and evaluation set. Our model is trained on the DTU training set, which is mentioned in Sect. 4.1.

For depth map estimation, we use an input image size of 1600 × 1200 and a fixed depth range of $[d_{min}, d_{max}] = [425mm, 905mm]$ for all input images. The source image number is set to $N_v = 5$. We choose $N_f = 2$ and $p_{t,1}, p_{t,2}, p_{t,3} = 0.6, 0.6, 0.6$ for the depth map filter and fusion step. Quantitative results are shown in Table 6 and corresponding point cloud reconstructions are illustrated in Fig. 7. Our method achieves a overall score of 0.365, which is comparable with other state-of-the-art methods. Qualitatively, the objects are mostly well-reconstructed because the image capturing process is carefully controlled. But there is still incompleteness in non-Lambertian and texture-less regions.

4.5 Ablation Study

In this section, we discuss other alternative volume fusion methods with implicit or explicit visibility awareness. To keep the simplicity of the network and clear demonstrate the effectiveness of the proposed component, we remove the

Table 7 Quantitative result of the depth map on the validation set of *BlendedMVS* with $N_v = 7$

	Fusion Method	Loss	<1 (%)	<3 (%)
base-var	Variance	1.50	79.31	92.25
base-ave	Average	0.999	83.03	94.95
base-max	Max Pooling	0.956	84.71	95.19
base-vis	Proposed	**0.908**	**85.35**	**95.48**
proposed	+ Coarse-to-fine	0.759	90.86	96.05

Bold entries are the best results among the compared methods Among the configurations without the coarse-to-fine strategy, the setting with the proposed fusion method achieves better result than others

coarse-to-fine architecture and directly use a MVSNet-like network as our baseline. The ablation study is performed on the BlendedMVS validation set and three types of evaluation metrics are considered: (1) the average L1 loss between the inferred depth map and the ground truth depth map; (2) the percentage of pixels with L1 error smaller than 1 depth-wise pixel ($<$ 1 percentage); and (3) the $<$ 3 percentage. Quantitative results are shown in Table 7 and Fig. 8

Baseline In this setting (*base-var*), we directly use the variance metric to fuse the feature volumes into one cost volume. The *base-var* setting is widely adopted by MVSNet and its following works (Yao et al., 2018; Chen et al., 2019; Yang et al., 2020; Cheng et al., 2020; Gu et al., 2020). However, the variance operation is under the assumption that all pixels in the reference should be visible from all views. As a result, the increasing input image number would lead to even worse evaluation metrics (see Fig. 8)

Averaging In this setting (*base-ave*), pair-wise cost volumes are fused to one multi-view volume by direct element-wise averaging. To fairly compare this setting with the proposed setting, we also apply the two step regularization as in the proposed framework. As is shown in Fig. 8, the <1 percentage accuracy of the *base-ave* is consistently increasing with the input image number. We believe the visibility information is implicitly encoded in the latent space and is dealt with by the two-step regularization. However, such implicit visibility awareness is apparently inferior to the proposed visibility fusion approach (see *base-vis* in Table 7 and Fig. 8).

Max Pooling In this setting (*base-max*), the fused volume is obtained by finding the element-wise maximum of all pair-wise volumes. This setting follows the fusion strategy of only considering the best matching pair among all reference-source image pairs. Similarly, all pair-wise losses are not counted toward the final loss. As is shown in Table 7 and Fig. 8, *base-max* outperforms *base-ave* but is still inferior to the proposed *base-vis*.

Weighted Averaging This setting (*base-vis*) is the proposed Vis-MVSNet without the coarse-to-fine architecture. Compared with *base-ave* and *base-max*, this setting utilizes the intermediate uncertainty as the weighting guidance for the pair-wise volume fusion. As the result, the significance of invisible pixels will be explicitly reduced in the volume fusion step.

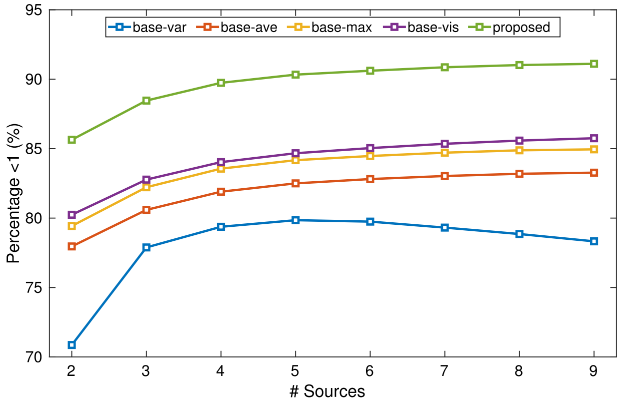


Fig. 8 Percentage of <1 of the depth maps on *BlendedMVS* w.r.t. N_v. The visibility-aware systems perform better than others and do not suffer from increasing N_v

The quantitative comparison is shown in Table 7 and Fig. 8. A significant improvement can be observed after introducing the two step regularization to the baseline (*base-ave* and *base-max* v.s. *base-var*). In addition, the proposed fusion further improves the result (*base-vis* v.s. *base-ave* and *base-max*). Finally, the full model with coarse-to-fine architecture outperforms others by a significant margin (*proposed* v.s. others).

If we apply the coarse-to-fine strategy to the backbone network, however, the improvement brought by the proposed fusion method will be shadowed on *BlendedMVS* whose images are relatively well-captured. Instead, we do the evaluations on the training set of *Tanks and Temples* whose images are captured in the wild. The evaluation metrics follow the benchmark method of *Tanks and Temples* described in Sect. 4.2. As shown by Table 8, the proposed fusion method outperforms the averaging and the max pooling method (*proposed* v.s. *cas-ave* and *cas-max*). And the proposed system achieves the best F-score in the majority of the scenes.

4.6 Memory and Time Consumption

In this section we discuss the memory and the time consumption of the inference. Because the volume fusion can be calculated online, we do not need to preserve previous feature maps and pair-wise cost volumes. Therefore, the memory consumption does not linearly increase with respect to the source view number. However, the time consumption of the pair-wise regularization increases linearly. Figure 9 shows the results of inference on the intermediate set of the *Tanks and Temples* (Knapitsch et al., 2017) dataset w.r.t. number of sources. The size of the inputs is $H \times W = 1056 \times 1920$.

Table 8 Quantitative result of the point cloud (F-score) on *Tanks and Temples* with $N_v = 20$

	Barn	Caterpillar	Church	Courthouse
cas-ave	66.20	**68.92**	57.48	18.74
cas-max	63.84	66.75	31.26	**20.80**
proposed	**68.00**	67.90	**58.60**	18.95
	Ignatius	Meetingroom	Truck	Mean
cas-ave	**91.72**	46.79	63.00	58.98
cas-max	88.85	38.89	57.11	52.50
proposed	89.29	**48.43**	**64.59**	**59.39**

Bold entries are the best results among the compared methods Among the configurations with the coarse-to-fine strategy , the setting with the proposed fusion method achieves better result than others

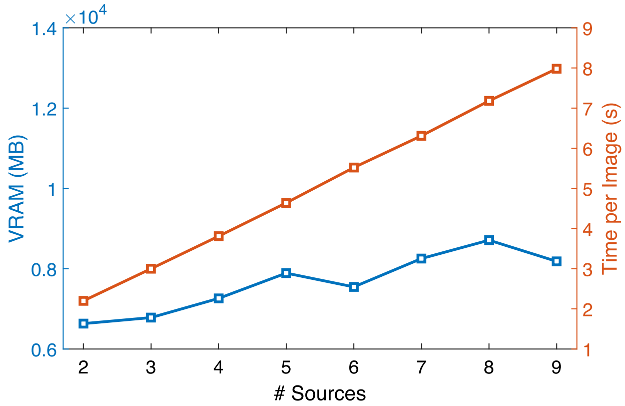


Fig. 9 VRAM and time consumption of the inference on *Tanks and Temples* w.r.t. N_v. Time consumption grows linearly, while the memory consumption does not grow significantly

Note that the memory consumption is not monotonic because of some engineering issues of PyTorch.

5 Conclusion

We have presented a visibility-aware depth inference framework for multi-view stereo reconstruction. We have proposed the two-step cost volume regularization, the joint inference of the pair-wise depth and the uncertainty, and the weighted average fusion of pair-wise volumes according to the uncertainty maps. The proposed method has been extensively evaluated on several datasets. Qualitatively, the system can produce more accurate and dense point clouds, which demonstrates the effectiveness of the proposed visibility-aware depth inference framework.

Acknowledgements This work is supported by Hong Kong RGC GRF 16206819, 16203518 and T22-603/15N.

References

Campbell, N. D., Vogiatzis, G., Hernández, C., & Cipolla, R. (2008). Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pp. 766–779.

Chen, R., Han, S., Xu, J., & Su, H. (2019). Point-based multi-view stereo network. In *International Conference on Computer Vision (ICCV)*, pp. 1538–1547.

Cheng, S., Xu, Z., Zhu, S., Li, Z., Li, L. E., Ramamoorthi, R., & Su, H. (2020). Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 2524–2534.

Furukawa, Y. & Ponce, J. (2006). Carved visual hulls for image-based modeling. In *European Conference on Computer Vision (ECCV)*, Springer, pp. 564–577.

Furukawa, Y., & Ponce, J. (2009). Accurate, dense, and Robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(8), 1362–1376.

Galliani, S., Lasinger, K., & Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision (ICCV)*, pp 873–881.

Grum, M., & Bors, A. G. (2014). 3d modeling of multiple-object scenes from sets of images. *Pattern Recognition, 47*(1), 326–343.

Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F. & Tan, P. (2020). Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 2495–2504.

Guo, X., Yang, K., Yang, W., Wang, X. & Li, H. (2019). Group-wise correlation stereo network. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3273–3282.

Hartmann, W., Galliani, S., Havlena, M., Van Gool, L. & Schindler, K. (2017). Learned multi-patch similarity. In *International Conference on Computer Vision (ICCV)*, pp 1586–1594.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Hu, X., & Mordohai, P. (2012). A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(11), 2121–2133.

Huang, P. H., Matzen, K., Kopf, J., Ahuja, N.,& Huang, J. B. (2018). Deepmvs: Learning multi-view stereopsis. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 2821–2830.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., & Aanæs, H. (2014). Large scale multi-view stereopsis evaluation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 406–413.

Ji, M., Gall, J., Zheng, H., Liu, Y., & Fang, L. (2017). Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *International Conference on Computer Vision (ICCV)*, pp. 2307–2315.

Ji, M., Zhang, J., Dai, Q., & Fang, L. (2020). Surfacenet+: An end-to-end 3d neural network for very sparse multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(11), 4078–4093.

Kar, A., Häne, C., & Malik, J. (2017). Learning a multi-view stereo machine. In *Neural Information Processing Systems (NeurIPS)*, vol 30.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Neural Information Processing Systems (NeurIPS)*, vol 30.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., & Bry, A. (2017). End-to-end learning of geometry and context for deep stereo regression. In *International Conference on Computer Vision (ICCV)*, pp. 66–75.

Kim, S., Min, D., Kim, S., & Sohn, K. (2018). Unified confidence estimation networks for robust stereo matching. *IEEE Transactions on Image Processing, 28*(3), 1299–1313.

Kim, S., Kim, S., Min, D., & Sohn, K. (2019). Laf-net: Locally adaptive fusion networks for stereo confidence estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 205–214.

Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Knapitsch, A., Park, J., Zhou, Q. Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG), 36*(4), 78.

Kuhn, A., Hirschmüller, H., Scharstein, D., & Mayer, H. (2017). A tv prior for high-quality scalable multi-view stereo reconstruction. *International Journal of Computer Vision, 124*(1), 2–17.

Kuhn, A., Lin, S., & Erdler, O. (2019). Plane completion and filtering for multi-view stereo reconstruction. In *German Conference on Pattern Recognition (GCPR)*, pp. 18–32.

Kuhn, A., Sormann, C., Rossi, M., Erdler, O., & Fraundorfer, F. (2020). Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In *International Conference on 3D Vision (3DV)*, pp. 404–413.

Kutulakos, K. N., & Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision, 38*(3), 199–218.

Lhuillier, M., & Quan, L. (2005). A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(3), 418–433.

Li, Z., Zuo, W., Wang, Z., & Zhang, L. (2020). Confidence-based large-scale dense multi-view stereo. *IEEE Transactions on Image Processing, 29*, 7176–7191.

Liao, J., Fu, Y., Yan, Q., & Xiao, C. (2019). Pyramid multi-view stereo with local consistency. *Computer Graphics Forum, 38*(7), 335–346.

Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J. M., Yang, R., Nistér, D., & Pollefeys, M. (2007). Real-time visibility-based fusion of depth maps. In *International Conference on Computer Vision (ICCV)*, pp. 1–8.

Paschalidou, D., Ulusoy, O., Schmitt, C., Van Gool, L., & Geiger, A. (2018). Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3897–3906.

Poggi, M., & Mattoccia, S. (2016). Learning from scratch a confidence measure. In *British Machine Vision Conference (BMVC)*, vol 2, pp. 4.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention*, pp. 234–241.

Schönberger, J. L., Zheng, E., Frahm, J. M., & Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pp. 501–518.

Schops, T., Schonberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., & Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3260–3269.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, vol 1, pp. 519–528.

Slabaugh, G. G., Culbertson, W. B., Malzbender, T., Stevens, M. R., & Schafer, R. W. (2004). Methods for volumetric reconstruction of visual scenes. *International Journal of Computer Vision, 57*(3), 179–199.

Sormann, C., Knöbelreiter, P., Kuhn, A., Rossi, M., Pock, T., & Fraundorfer, F. (2020). Bp-mvsnet: Belief-propagation-layers for multi-view-stereo. In *International Conference on 3D Vision (3DV)*, pp. 394–403.

Tola, E., Strecha, C., & Fua, P. (2012). Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications, 23*(5), 903–920.

Tosi, F., Poggi, M., Benincasa, A., & Mattoccia, S. (2018). Beyond local reasoning for stereo confidence estimation with deep learning. In *European Conference on Computer Vision (ECCV)*, pp. 319–334.

Wang, F., Galliani, S., Vogel, C., Speciale, P., & Pollefeys, M. (2021). Patchmatchnet: Learned multi-view patchmatch stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14194–14203.

Xu, Q., & Tao, W. (2019). Multi-scale geometric consistency guided multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 5483–5492.

Xu, Q., & Tao, W. (2020). Planar prior assisted patchmatch multi-view stereo. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(07), 12516–12523.

Xu, Z., Liu, Y., Shi, X., Wang, Y., & Zheng, Y. (2020). Marmvs: Matching ambiguity reduced multiple view stereo for efficient large scale scene reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5981–5990.

Xue, Y., Chen, J., Wan, W., Huang, Y., Yu, C., Li, T., & Bao, J. (2019). Mvscrf: Learning multi-view stereo with conditional random fields. In *International Conference on Computer Vision (ICCV)*, pp. 4312–4321.

Yan, J., Wei, Z., Yi, H., Ding, M., Zhang, R., Chen, Y., Wang, G., & Tai, Y.W. (2020). Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision (ECCV)*, pp. 674–689.

Yang, J., Mao, W., Alvarez, J. M., & Liu, M. (2020). Cost volume pyramid based depth inference for multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 4877–4886.

Yao, Y., Li, S., Zhu, S., Deng, H., Fang, T., & Quan, L. (2017). Relative camera refinement for accurate dense reconstruction. In *2017 International Conference on 3D Vision (3DV)*, IEEE, pp. 185–194.

Yao, Y., Luo, Z., Li, S., Fang, T., & Quan, L. (2018). Mvsnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pp. 767–783.

Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., & Quan, L. (2019). Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5534.

Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., & Quan, L. (2020). Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1790–1799.

Zhang, J., Yao, Y., Li, S., Luo, Z., & Fang, T. (2020). Visibility-aware multi-view stereo network. In *British Machine Vision Conference (BMVC)*.

Zhang, J., Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., & Quan, L. (2020). Learning stereo matchability in disparity regression networks. In *International Conference on Pattern Recognition (ICPR)*, pp. 1611–1618.

Zhang, R., Li, S., Fang, T., Zhu, S., & Quan, L. (2015). Joint camera clustering and surface segmentation for large-scale multi-view stereo. In *International Conference on Computer Vision (ICCV)*, pp. 2084–2092.

Zheng, E., Dunn, E., Jojic, V., & Frahm, J. M, (2014). Patchmatch based joint view selection and depthmap estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1510–1517.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.