Mesh-Guided Multi-View Stereo with Pyramid Architecture

Yuesong Wang[1], Tao Guan[1,3]*, Zhuo Chen[1], Yawei Luo[1], Keyang Luo[1], Lili Ju[2]

[1]School of Computer Science & Technology, Huazhong University of Science & Technology, China
[2]University of South Carolina, USA [3]Farsee2 Technology Ltd, China

{yuesongw, qd_gt, cz_007, royalvane, kyluo}@hust.edu.cn, ju@math.sc.edu

Abstract

Multi-view stereo (MVS) aims to reconstruct 3D geometry of the target scene by using only information from 2D images. Although much progress has been made, it still suffers from textureless regions. To overcome this difficulty, we propose a mesh-guided MVS method with pyramid architecture, which makes use of the surface mesh obtained from coarse-scale images to guide the reconstruction process. Specifically, a PatchMatch-based MVS algorithm is first used to generate depth maps for coarse-scale images and the corresponding surface mesh is obtained by a surface reconstruction algorithm. Next we project the mesh onto each of depth maps to replace unreliable depth values and the corrected depth maps are fed to fine-scale reconstruction for initialization. To alleviate the influence of possible erroneous faces on the mesh, we further design and train a convolutional neural network to remove incorrect depths. In addition, it is often hard for the correct depth values for low-textured regions to survive at the fine-scale, thus we also develop an efficient method to seek out these regions and further enforce the geometric consistency in these regions. Experimental results on the ETH3D high-resolution dataset demonstrate that our method achieves state-of-the-art performance, especially in completeness.

1. Introduction

Obtaining geometric information of a target scene is a very important task in many applications and multi-view stereo (MVS) is probably the most convenient approach for 3D geometry reconstruction in terms of efficiency and cost since MVS only requires a set of calibrated images as the input and extracts 3D geometric information based on their photo-consistency. Consequently, MVS has been a hot research topic in computer vision for decades.

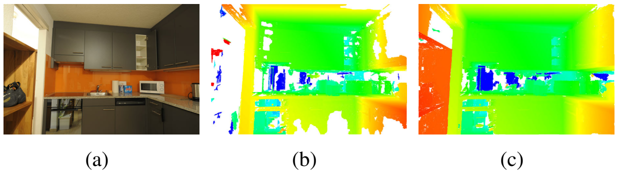


Figure 1. (a) An input image from the ETH3D test dataset; (b) the depth map produced by MVS without guidance; (c) the depth map by our method.

One of the classical strategies for MVS is plane-sweep [5, 2, 8] which sweeps a plane through the target scene to obtain its 3D geometry. However, only scenes consisting of planar surfaces can be correctly estimated using such a strategy. Many PatchMatch-based algorithms [9, 6, 32, 42, 7, 3, 29] have successfully overcome this limitation and achieved impressive results. These methods assume that the scene consists of a vast number of small planar patches; by projecting a 3D patch into images without occlusion, image patches can be correspondingly obtained and the photometric consistency between them must be high. However, it is often hard to predefine the size of the patches. Small patch size produces accurate points in areas with rich texture but perform badly in low-textured regions, and enlarging the patch size could make the algorithm more robust but may decrease the accuracy. Image pyramid [1] is then implanted into MVS in [37, 22] to deal with this issue. Although MVS with the pyramid architecture solves the problem brought by the patch size, it still suffers a lot from textureless areas of the scene since the photometric consistency performs poorly in textureless regions even at the coarsest scale. This drawback has been partially addressed in [19, 28] based on the assumption that textureless regions are often piecewise flat. In this spirit, images are segmented into superpixels which are then treated as planes. Most textureless regions can be reconstructed correctly while errors may also be consequently induced since the depths in superpixels are unreliable sometimes, which results in faulty estimation of corresponding plane parame-

*Corresponding author.

ters.

With the development of convolutional neural networks (CNN), many researches show that CNN-based patch descriptors [33, 34, 41, 26, 10, 27, 21] can outperform the handcrafted ones in low-textured regions. Based on these works, some CNN-based MVS methods [38, 13, 11, 40, 23] further improve the robustness against untextured regions as the receptive field of each feature is much larger than the patch size. On the other hand, these MVS algorithms are usually limited by the memory sizes of devices, for instance, they usually can't handle high-resolution images very well (see the leaderboard page of the ETH3D high-resolution multi-view benchmark [30]). Downsampling images might not be a good choice for CNN-based methods since the accuracy will decrease at the same time. Thus, there still exist difficulties when dealing with low-textured regions of the high-resolution scenes.

In this paper, we present a novel mesh-guided MVS method (MG-MVS) with a pyramid structure that utilizes the surface mesh as the guidance to achieve high completeness for high-resolution scenes with low-textured regions. Specifically, we first adopt a pyramid architecture and fuse the depth maps of the coarsest scale to construct a surface mesh which is then used to enhance the completeness of these depth maps. We then feed them to the finer scales and retain correct estimates at untextured regions by enforcing geometric consistency. Figure 1 shows a depth map estimated by our method compared with the result without the guidance of the surface mesh.

The main contributions of this paper are as follows: 1) we propose to leverage the surface mesh produced at the coarse-scale to guide the MVS process at the fine-scale to improve the completeness of the depth map estimation; 2) to avoid the influence brought by the erroneous faces in the surface mesh, we design a deep neural network to generate confidence maps of the depth prediction, which are then used for removal of erroneous depth values; 3) we design a textureless region detector to enforce that the correct values in textureless regions can be retained at the fine-scale; 4) our method achieves the "state-of-the-art" performance on the ETH3D high-resolution multi-view dataset.

2. Related Work

PatchMatch-based MVS methods. PatchMatch-based MVS has been the research hotspot of MVS for more than ten years. PMVS [6] divides an image into cells and estimates the depth and normal for each cell. Shen [32] proposes pixel-wise depth estimation and propagation of the reliable depths to neighbor pixels. Gipuma [7] designs a checkerboard propagation that can be implemented on GPU to speed up the computation. Zheng *et al*. [42] select proper source views for each pixel to improve the performance of MVS against occlusions and illumination aberration. COLMAP [29] further embeds normal estimation into the framework of [42] and uses geometric priors to improve the robustness of view selection. Although PatchMatch-based MVS has been quite successful, it still has an obvious weakness that textureless regions are seldom managed correctly, as demonstrated by low completeness of the results produced by these mentioned methods. To overcome this difficulty, Xu *et al*. [37] first downsample input images and execute ACMH (Adaptive Checkerboard sampling and Multi-Hypotheses joint view selection) on the coarse-scale to enlarge the receptive field of patches. Then the depth maps from the coarser scale are upsampled to the finer scale using the joint bilateral upsampler [17] and play the role of guider for the finer-scale processing. The geometry in textureless regions is preserved using geometric consistency. Liao *et al*. [22] propose to build a similar pyramid architecture in MVS. They assume that neighbor pixels with similar colors may come from the same surface and enforce local consistency to deal with textureless regions. Due to the unreliability of photometric consistency in low-textured regions, both of [37, 22] still do not perform very well in completeness. The idea of [22] is quite similar to TAPA-MVS [28] which segments images into superpixels since the pixels in a superpixel are alike in color. The superpixels are then treated as planes during the MVS procedure. Kuhn *et al*. [19] further improve TAPA-MVS by merging similar superpixels so that there will be enough valid points in single superpixel to estimate the corresponding plane. However, inaccurate points in superpixels will cause faulty estimations of planes. Moreover, planes sometimes are not capable of expressing the geometry of superpixels.

CNN-based MVS methods. Deep learning networks have achieved great success in recent years and a vast number of CNN-based methods for tasks in computer vision have shown incredible performance. There also exist some remarkable networks for MVS. Yao *et al*. [39] design MVSNet for depth map inference and demonstrate its effectiveness on the DTU [12] and Tanks and Temples [16] datasets. They further combine MVSNet with the recurrent neural network to reduce the memory consumption [40]. P-MVSNet [23] makes use of a confidence metric based on the mean-square error and a hybrid 3D U-Net to aggregate the photometric consistency into a patch-wise matching confidence volume. However, due to the limitation of GPU memory, CNN-based methods usually could not perform well on high-resolution datasets such as the ETH3D dataset. Huang *et al*. [11] try to solve this problem by breaking images into patches, but the result on the ETH3D dataset is still unsatisfactory. Besides these depth-estimating networks, some researchers also use CNNs to improve the outputs from other ways. Wu *et al*. [36] add a semantic inference network into their baseline model and the semantic segmentation results are further improved by [24, 25]. Kuhn

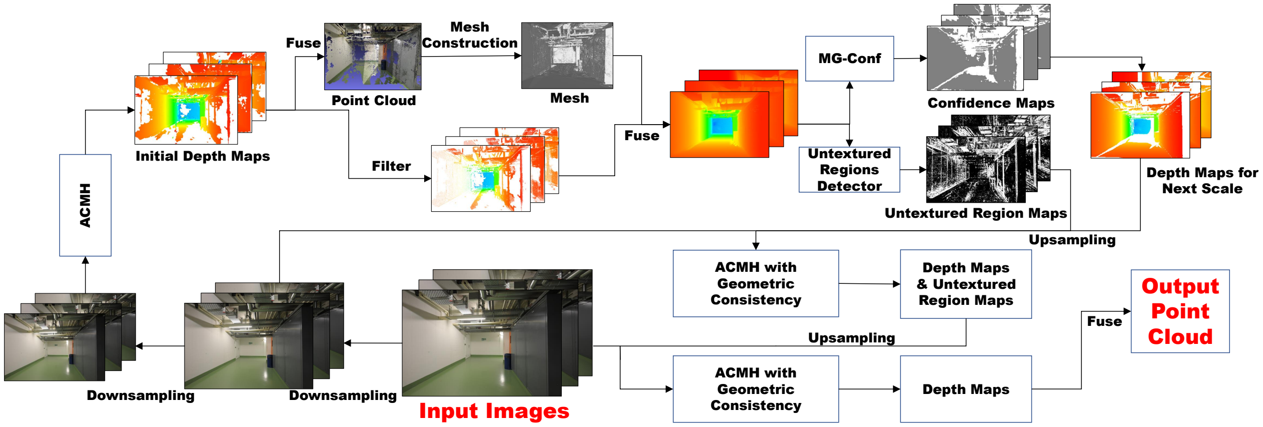


Figure 2. An overview of the proposed method. Starting with the input images, we build a pyramid of image sets with three scales by downsampling, and then use ACMH to obtain the initial depth maps of the coarsest scale and fuse them into a coarse point cloud, which is reconstructed to a surface mesh (Section 3.2). Next, we leverage this mesh to complete the filtered depth maps (Section 3.3). For further removal of erroneous depths, we feed the depth maps to a specially designed neural network "MG-Conf" to predict the corresponding confidence maps and subsequently remove those depths with low confidence (Section 3.4). Meanwhile, we also use an untextured region detector (Section 3.5) to identify untextured regions. Both untextured region maps and depth maps are upsampled to guide the MVS process at the next finer scale, and we enforce geometric consistency in untextured regions to retain reliable estimates. Finally, the depth maps of the finest scale are fused to produce the final point cloud.

et al. [19] filter out outliers in segmented sky areas via a deep learning network. Fabio *et al.* [35] build a network to predict the confidence of the disparity maps. Sunok *et al.* [15] combines information from the disparity map and the cost volume to estimate the confidence. They further propose a scale inference network [14] to improve the accuracy of confidence prediction. In this paper we also train a network to process intermediate results in the pipeline.

3. Proposed Method

3.1. Motivation

We could regard the MVS reconstruction using pyramid architecture as a process of sculpturing. Making a sculpture usually starts with carving a rough model, then the details are carved based on the rough model. Similarly, MVS reconstruction first uses the coarse-scale images to obtain rough depths, around which the local optimal depth values are then found at the fine-scale. Carving at the finer scales relies heavily on the rough depths, which makes the completeness of the coarsest-scale depth maps an important factor. It is well-known that some points can be correctly estimated at certain particular views but wrong at other views due to illumination and other external conditions. Propagating these correct depth values to neighbor views can improve the completeness of the corresponding depth maps but needs considering occlusions. Since the surface mesh contains all occlusion relations, it is a perfect medium for depth propagation between the views. Then the completed depth maps are fed to the next finer scale for further refinement. Increasing the completeness of depth maps is a key issue since the fusion process of depth maps only keeps the estimates supported by enough views. Propagating the correct estimates to neighbor views also means gaining more support for these estimates to ensure that they can be kept in the final point cloud.

To realize the above idea, we first construct a pyramid of image sets with three scales by downsampling from the original input images, and use the basic MVS method, ACMH [37], to generate the initial depth maps at the coarsest scale. Then, we fuse these depth maps into a point cloud using a relatively loose constraint to retain correct estimates as many as possible and build the corresponding surface mesh using [20]. By mesh projection, we propagate the correct depth values to views that are incomplete. To further refine the depth maps, we filter the depth maps using the geometric consistency and replace those values of low geometric consistency with the values from mesh projection. Influenced by outliers, the mesh may still contain some wrong faces and these faces could cause errors in depth maps, thus we also design a neural network to filter out these wrong depth values. We next upsample the depth maps to the finer scale and execute ACMH with geometric consistency. The same upsampling and ACMH with geometric consistency are repeated twice. To maintain accuracy, we only enforce the geometric consistency in untextured regions (identified by the untextured region detector). The whole pipeline of our method is illustrated in Figure 2.

3.2. Mesh Construction

We first downsample the input images to the coarsest scale and execute ACMH at this scale to get the initial depth maps. For efficiency of ACMH, we only use perturbing hypotheses in the refinement step and the range of perturbation changes along with the iteration: $r_{now} = r_{init} \cdot (0.5)^m$ for the m-th iteration. The initial depth maps usually contain lots of discontinuous segments, especially in textureless regions. We apply the segments removing strategy used in OpenMVS [3] to clean them.

We then fuse the depth maps to obtain a coarse point cloud by using the relative depth differences [29]. Given a depth from a reference view V_r, we convert it into a 3D point in world coordinate, which we then project to a neighbor view V_{sn} to get the depth d_r of this point in the coordinate of V_{sn} and the corresponding depth d_{sn} from the depth map of V_{sn}. We consider the two depths to be a consistent match if they fulfill:

$$\frac{|d_r - d_{sn}|}{d_r} < \epsilon_3, \tag{1}$$

where ϵ_3 is the similarity threshold at the coarsest scale. We set $\epsilon_3 = 4\epsilon_1$ since the images of the coarsest scale are four times smaller than those of the finest scale. The 3D point from V_r is then projected to each neighbor view and if the number of matches fulfils $N_{match} >= 1$, we add this 3D point to the point cloud. We cast each of views as the reference view in turn to get the complete point cloud. The loose threshold of N_{match} ensures correct estimates at untextured regions retained, but it makes the point cloud noisy. To extract the surface from points with a vast number of outliers, we adopt the surface reconstruction algorithm based on visibility information [20] for its strong robustness. The point cloud is used to build Delaunay tetrahedra, which are then marked as inside or outside the object, and the target surface lies between the tetrahedra with different labels.

3.3. Mesh Guidance

After obtaining the surface mesh at the coarsest scale, we use it to guide the MVS processing. We first project the mesh to each of the views to fill up the segments removed by the segment removing strategy. Moreover, we also wish to refine the depth maps using this mesh, thus we filter the depth maps using geometric consistency which is similar to the above fusion step. We here filter out the depths with $N_{match} < 2$ and label them invalid, and replace those invalid values with depths from mesh projection. The reason why the threshold of N_{match} in the fusion step is looser than that in this filter step is that the surface reconstruction method after the fusion step has the ability to further filter out more outliers (some valid depths may be treated as outliers) while there will be a lot of incorrect depths marked as valid if we keep depths with $N_{match} = 1$ in the filter step.

By mesh projection, we can propagate reliable estimates in untextured regions from a reference view to its neighbor views with occlusion considered, which can increase the completeness of the neighbor depth maps dramatically (Figure 3). Conversely, the complete neighbor depth maps contribute to the estimation of the reference view at the finer scale due to geometric consistency. Besides, using the surface mesh to guide MVS also can achieve plane completion as shown in Figure 4. Different from TAPA-MVS and PCF which use depths from only one view to estimate the planes, our surfaces are reconstructed using information from all different views.

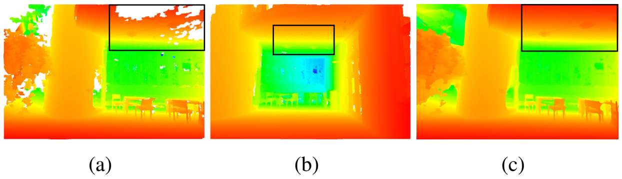


Figure 3. Depth propagation between views. (a) The depth map of view A; (b) the depth map of its neighbor view B (the black boxes pointing to the basically same region in both images); (c) the depth map of view A after mesh projection.

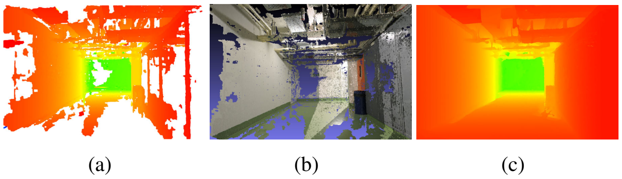


Figure 4. Plane completion. (a) The depth map at the coarse scale; (b) the point cloud at this scale; (c) the depth map after mesh projection.

The resulting depth maps are then fed to the next finer scale to further guide the MVS processing. With good initial values, better estimates can be found at the finer scale and the searching range is also constrained by geometric consistency, thus it helps avoid them being trapped into other local optimal values.

3.4. Confidence Prediction Networks

Despite the robustness of the surface reconstruction algorithm, it still could produce some wrong faces which would induce errors in depth maps and then misguide the MVS processing at the finer scale. Traditional methods for removing wrong depths are mostly based on photometric and geometric consistency, which are of no use in our situation since the errors brought by mesh projection have no geometric difference and usually appear in untextured regions. It has been shown in [14] and [35] that convolutional neural networks can achieve outstanding performance in estimating the confidence map for a given initial disparity map. Thus based on the work of LAF-Net [14], we design a deep neural network "MG-Conf", to solve this problem, whose architecture is presented in Figure 5.

Our MG-Conf considers both matching costs, *depths* and *colors*, and predicts confidence of the depth at each pixel.

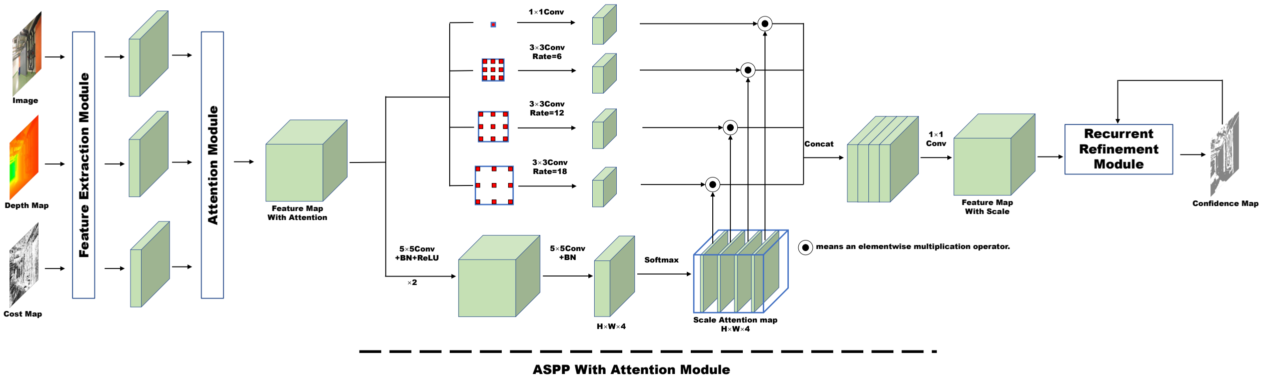
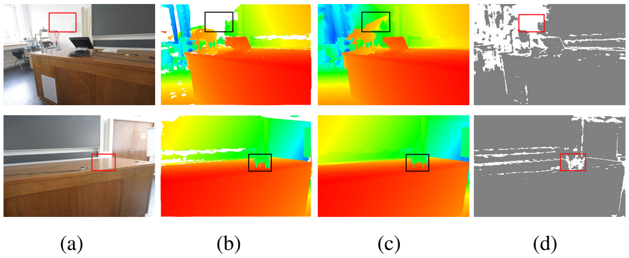


Figure 5. The architecture of the neural network "MG-Conf". Feature extraction module takes colors, depths and costs as input to generate feature maps which are then fused by the following attention module. ASPP with attention module convolutes the fused features using different receptive field sizes to get the confidence map and the recurrent refinement module further improves its accuracy.



(a) (b) (c) (d)

Figure 6. (a) The color images; (b) the depth maps before mesh projection; (c) the depth maps after mesh projection; (d) the confidence maps predicted by MG-Conf (pixels with white color correspond to wrong depths). The top row shows that MG-Conf can find out errors brought by mesh projection, and the bottom row demonstrates that errors from the the basic MVS method can also be located.

Rather than directly feeding the current matching costs to MG-Conf, we would like the network to adaptively determine whether a matching cost is distinctive enough to be considered as a globally optimal solution. To obtain the distinctiveness of the current costs, the current depths and normals are perturbed for three times, we then compute the costs of these perturbations using the view weights from ACMH. The current costs together with costs from perturbations are then fed to the network together. We set the perturbing range as $r_{init} \cdot (0.5)^{\frac{\tau}{2}}$ where τ is the total iteration number of ACMH.

Similarly to LAF-Net, we extract features from costs, depths and colors and fuse them using the attention module. Our MG-Conf then uses the fused feature maps to predict the confidence map. Large receptive fields yield robust results while causing loss of details, thus LAF-Net proposes a scale inference module to infer the optimal size of the receptive field for each pixel. However, its consumption of memory is unbearable for our device. Noticing that the convolution in the scale inference module is quite similar to dilated convolution, we combine Atrous Spatial Pyramid Pooling (ASPP) [4] with attention layers to achieve the adjustment of the receptive field for each pixel. The receptive field size of one feature can be calculated using:

$$R = n(k-1) + r_{ori}, \tag{2}$$

where n denotes the number of convolution layers, k is the size of the kernel and r_{ori} is the receptive field size of the input. The feature extraction module contains three convolution layers and the kernel size is three, thus the receptive field size of one feature equals seven. After three layers of convolution with the kernel size of five, the receptive field size of the scale attention map enlarges to nineteen, which corresponds to the receptive field of the second layer in ASPP since we are concerned with two situations: 1) the features in the receptive field of the scale attention map are capable of predicting the confidence; 2) Information from features outside this field is also demanded. We will use the output of ASPP with attention to predict a confidence map which will be fed to the recurrent refinement module for further improvement. We binarize the final confidence map and directly remove those depths which are labeled as unreliable by MG-Conf.

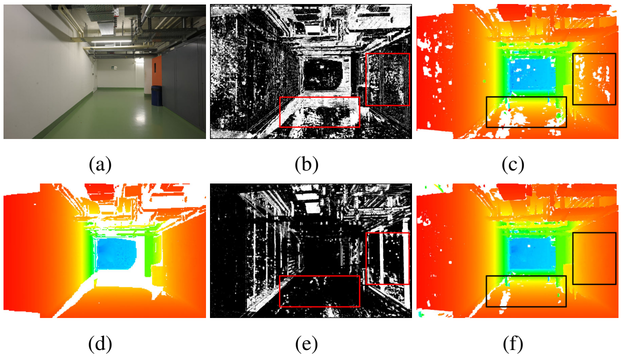
Besides eliminating the negative effects of mesh projection, MG-Conf also can help remove the errors from the basic MVS method (Figure 6). Limited by the device memory, we need rescale the input data into smaller sizes using the nearest neighbor interpolation, which results in mistaken deletion such as boundaries. However, the influence on completeness is slight since the propagation of ACMH can fill up those mistakenly deleted depths as long as there still exist correct depths nearby.

3.5. Untextured Region Detection

Geometric consistency could lead to blurred details and ACMM (combining ACMH with geometric consistency guidance) [37] is then proposed, which contains a detail restorer to detect thin structures and boundaries and only use

Table 1. Performance comparison results of our method with its model variants. The three values are accuracy / completeness / F$_1$ score (in %). We present results under tolerances of $1cm$ and $2cm$ since small tolerances reflect the changes in accuracy better.

DataSets	Tolerance	Baseline	Without MC	Without URD	Ours
Office	$1cm$	**82.14** / 37.18 / 51.19	73.33 / **46.53** / 56.93	81.99 / 37.40 / 51.36	79.19 / 45.83 / **58.06**
	$2cm$	**89.58** / 47.56 / 62.14	82.39 / **62.40** / 71.02	89.54 / 47.77 / 62.30	87.72 / 60.15 / **71.37**
Electro.	$1cm$	80.52 / 56.80 / 66.62	84.79 / 67.74 / 75.31	**87.60** / 67.16 / 76.03	86.62 / **67.83** / **76.08**
	$2cm$	88.13 / 76.18 / 81.72	90.55 / **80.43** / 85.19	**93.29** / 79.36 / 85.76	92.33 / 80.35 / **85.93**
Avg	$1cm$	81.33 / 46.99 / 58.91	79.06 / **57.14** / 66.12	**84.80** / 52.28 / 63.70	82.91 / 56.83 / **67.07**
	$2cm$	88.86 / 61.87 / 71.93	86.47 / **71.42** / 78.11	**91.42** / 63.57 / 74.03	90.03 / 70.25 / **78.65**


(a) (b) (c)

(d) (e) (f)

Figure 7. (a) The input image; (d) the initial depth map; (b) the details map produced by the detail restorer (pixels with white color are details); (c) the final depth map guided by details map; (e) the untextured region map (untextured regions are in black color); (f) the final depth map of our method.

photometric consistency in these specific regions. The detail restorer may work well in ACMM , but sometimes it mistakes textureless regions as details when applying it to the proposed method (Figure 7(b)) and correct estimates in these regions could easily be impaired without geometric constraint (Figure 7(c)). This is because some depth values in textureless regions are from the mesh in our method and the costs of these depths are often not locally optimal. After the execution of ACMH at the finer scale, the values in textureless regions are trapped into wrong local optimums in despite of the correct initial values. If the differences between the costs of initial depths and the local optimums are large, the corresponding pixels are marked as details by the detail restorer.

The purpose of enforcing geometric consistency is to prevent damage to those depth estimates which could be easily impaired at the finer scale. In other words, we need to seek depths with costs that are indistinctive. Noticing that the purpose of adding cost maps into MG-Conf is to take the distinctiveness of the estimates into account, we also can locate the indistinctive values in the same way. Similarly, we perturb the current depths and normals for N times and calculate their costs. Then we compute the average of the absolute differences:

$$f_{avg} = \frac{\sum_{i=1}^{N} \min(f_{max}, |c_{now} - c_i|)}{N}, \tag{3}$$

where c_{now} is the matching cost of the current estimate, and c_i is the cost of the i-th perturbed result. If $f_{avg} < f_{thresh}$, we consider it indistinctive. We set $f_{thresh} = 0.3$ and $f_{max} = 0.6$ in all experiments.

Table 2. Ablation study for the ASPP with attention module (ATT-ASPP). We consider both the accuracy of the predictions and the recall rate of erroneous depths (in %).

	Accuracy	Recall Rate
Without ATT-ASPP	**0.8609**	0.5851
With ATT-ASPP	0.8119	**0.6919**

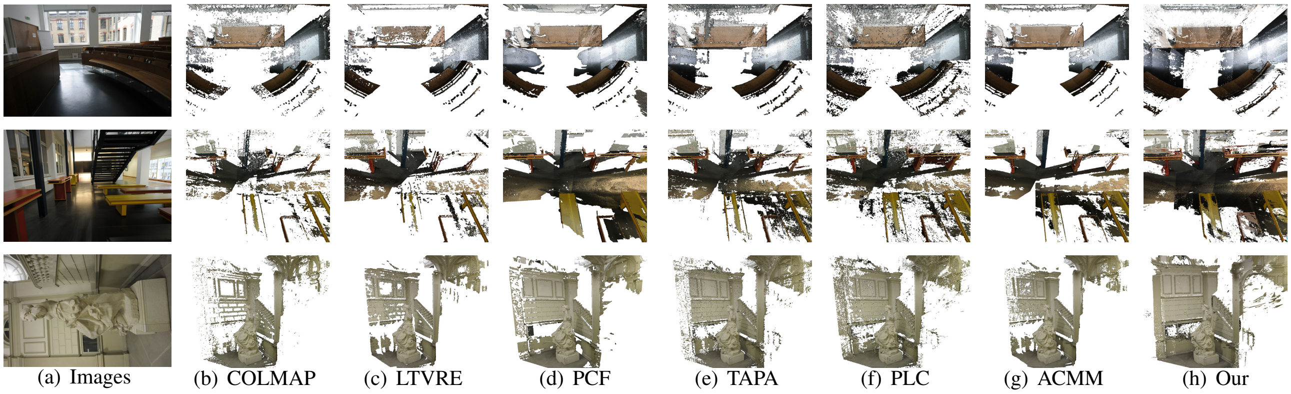
Table 3. Evaluation of point clouds obtained by our method using MG-Conf with ATT-ASPP / without ATT-ASPP, where the three values correspond to accuracy/completeness/F$_1$ score (in %) respectively.

DataSets	Tolerance	Without ATT-ASPP	With ATT-ASPP
Office	$1cm$	77.42 / **46.08** / 57.77	**79.19** / 45.83 / **58.06**
	$2cm$	85.27 / **60.64** / 70.88	**87.72** / 60.15 / **71.37**
Electro.	$1cm$	85.12 / 67.80 / 75.48	**86.62** / **67.83** / **76.08**
	$2cm$	90.62 / **80.41** / 85.21	**92.33** / 80.35 / **85.93**
Avg	$1cm$	81.27 / **56.94** / 66.63	**82.91** / 56.83 / **67.07**
	$2cm$	87.95 / **70.53** / 78.05	**90.03** / 70.25 / **78.65**

These indistinctive values are more likely to appear in untextured regions as shown in Figure 7(e), so we call the resulted map the untextured region map. We calculate the untextured region maps after obtaining the depth maps at each scale of the pyramid. We then retain the correct estimates in untextured regions by enforcing geometric consistency while using only photometric consistency for distinctive regions to avoid decreasing accuracy. In addition, our untextured region detector is more time saving compared to the detail restorer which needs to execute ACMH twice to get the details.

4. Experimental Results

We implement and test our method “MG-MVS” on a computer with an Intel E5-1650 CPU and a GTX 1080Ti GPU. We mainly focus on demonstrating the ability of our method in improving the completeness and overall reconstruction quality of MVS for high-resolution images. The dataset for evaluation is the ETH3D dataset [31], which provides multi-view stereo scans with high-resolution images. It is worth noting that CNN-based MVS methods usually couldn’t deal with such large-scale images with satisfactory accuracy due to memory restriction. Moreover, images in the ETH3D dataset do not overlap with each other much



(a) Images (b) COLMAP (c) LTVRE (d) PCF (e) TAPA (f) PLC (g) ACMM (h) Our

Figure 8. Point cloud comparisons on some test scans (lectur., lounge, statue).

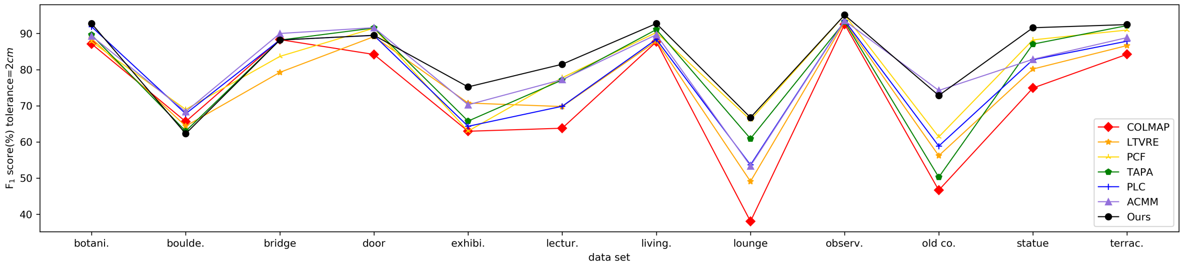


Figure 9. F_1 scores on all scans of the ETH3D with tolerance=$2cm$.

and often contain lots of untextured regions, which makes it difficult to produce satisfactory MVS results with high completeness. The training set of ETH3D provides ground-truth (GT) depth maps and point clouds while the GT data for the test set are not publicly available. In addition to completeness, the ETH3D benchmark also evaluates the accuracy and F_1 score of combining both completeness and accuracy.

Limited by the device memory, we could not directly feed the images of original resolution to MG-Conf. For the training of MG-Conf, we downsample images from the training set to the coarsest scale and execute basic MVS and mesh projection to generate initial depth maps, which are then used to obtain the confidence maps. We continue to downsample the input data of MG-Conf to 300 × 200. Different from LAF-Net, we use Adam optimizer and train for a total of 256 epochs. The GT confidence map we use is simply the binary map that indicates whether a depth estimate is right or wrong at each pixel. If the relative difference of the estimated depth and the corresponding GT depth is larger than a threshold (similar to Eq.(1)), this depth is marked as a wrong depth. The threshold is set to ϵ_3. After we obtain the prediction from MG-Conf, we upsample it to the resolution of the input data of the second fine scale in our pyramid for errors removing. For the depth map fusion at the finest scale, we adopt the fusion step of [29] and set the relative depth difference threshold ϵ_1 to 0.01, the angle between normals threshold to 20°, the reprojection error threshold to 2, and the minimum matching pixels to 3.

4.1. Ablation study

Since we have no access to the GT depths of the test set, we randomly choose two samples from the training set for validation (in our experiments the selected test set consists of office and electro) for the ablation study of our method.

Effects of mesh guidance, MG-Conf and untextured region detector – We remove MG-Conf (MC) and the untextured region detector (URD) in turn and compare the results with those from the full version of MG-MVS. To prove the effectiveness of mesh guidance, we also compare them with the baseline method which uses ACMH only to build the three-level pyramid. The results are reported in Table 1. The mesh guidance can improve the completeness dramatically while it induces only a little decrease in accuracy, which results in an increase of F_1 score. MG-Conf helps to eliminate the negative effects of erroneous faces in the mesh but could also cause loss of completeness due to some mistaken removal. However, the decrease of completeness is acceptable since F_1 score still increases. The untextured region detector is also vital to MG-MVS since it can preserve those indistinctive values to maintain high completeness with only a small loss of accuracy.

Effect of ASPP with attention module in MG-Conf – Since the purpose of MG-Conf is to remove the erroneous depths, we take the recall rate of errors and the accuracy of the confidence maps as evaluation metrics. We remove ASPP with attention module from MG-Conf and execute the same training process, then we compare the perfor-

Table 4. Evaluation of completeness on the high-resolution multi-view test set of ETH3D at different thresholds ($2cm$ and $10cm$).

Tol.	Method	Botani.	Boulde.	Bridge	Door	Exhibi.	Lectur.	Living	Lounge	Observ.	Old co.	Statue	Terrac.
$2cm$	COLMAP	81.44	53.00	83.75	75.53	48.34	48.81	81.40	24.14	90.30	32.95	60.52	75.65
	LTVRE	84.00	51.55	67.08	83.53	58.78	54.94	80.31	34.54	93.23	40.64	68.03	78.58
	PCF	86.06	65.96	87.03	91.10	69.86	69.79	90.64	57.54	96.59	64.37	82.29	90.32
	TAPA	90.40	52.07	89.19	89.61	62.69	70.07	92.67	48.89	94.32	38.26	79.97	91.12
	PLC	92.93	58.68	90.64	88.30	60.90	64.43	90.53	42.38	94.57	56.52	76.54	86.33
	ACMM	84.25	57.87	90.36	89.93	63.17	65.64	86.04	37.79	94.10	65.89	71.52	85.55
	Ours	**97.79**	**76.27**	**94.96**	**94.49**	**78.74**	**83.87**	**92.94**	**64.34**	**96.94**	**75.19**	**94.09**	**95.66**
$10cm$	COLMAP	96.72	78.85	95.89	92.83	71.55	77.54	95.89	59.40	99.14	67.26	88.26	91.11
	LTVRE	92.89	83.18	93.15	94.74	78.99	75.14	92.69	65.64	97.63	69.46	81.82	89.74
	PCF	95.44	91.58	95.29	96.10	85.07	82.75	97.45	78.28	99.78	80.60	95.65	97.12
	TAPA	98.48	85.17	98.42	96.58	82.44	87.24	98.53	82.77	98.84	59.09	97.88	98.72
	PLC	99.17	87.49	**99.25**	96.55	88.43	91.26	98.21	85.92	**99.86**	89.77	97.57	97.24
	ACMM	93.61	81.34	98.50	95.92	87.88	83.85	95.20	67.20	97.89	86.90	84.90	92.06
	Ours	**99.79**	**95.71**	99.07	**99.06**	**94.45**	**96.42**	**98.70**	**88.81**	99.74	**95.35**	**99.86**	**99.97**

Table 5. Evaluation on the high-resolution multi-view test set of ETH3D at different tolerances ($2cm$ and $10cm$), where the three values correspond to accuracy/completeness/F_1 score (in %) respectively.

Method	$2cm$	$10cm$
COLMAP	91.97 / 62.98 / 73.01	98.25 / 84.54 / 90.40
LTVRE	**93.04** / 66.27 / 76.25	**99.18** / 84.59 / 90.99
PCF	82.15 / 79.29 / 80.38	92.12 / 91.26 / 91.56
TAPA	85.71 / 74.94 / 79.15	94.93 / 90.35 / 92.30
PLC	82.09 / 75.23 / 78.05	94.05 / 94.23 / 94.11
ACMM	90.65 / 74.34 / 80.78	98.05 / 88.77 / 92.96
Ours	80.32 / **87.11 / 83.41**	94.08/ **97.24 / 95.61**

mance on the validation set with the full version of MG-Conf. The results are reported in Table 2, which shows that although MG-Conf without ATT-ASPP can achieve higher accuracy, the price is that it labels significantly more erroneous depths as correct. Some of the mistakenly removed depths can be regained by the propagation of ACMH if there exist correct depths nearby, but it is not easy for ACMH to jump out of the traps brought by erroneous depths. Thus the recall rate of errors is more important for our method when the difference of accuracies is small, as demonstrated by the results shown in Table 3.

4.2. Evaluation on the ETH3D dataset

We next compare our method with many state-of-the-art methods (namely COLMAP [29], LTVRE [18], PCF-MVS [19], TAPA-MVS [28], PLC [22], ACMM [37]) on the test set of ETH3D dataset. Figure 8 shows point clouds of some sample scans produced by these methods. To further quantify our completeness, we report the evaluation results from ETH3D benchmark website in Table 4. As we can see, our method outperforms all other methods in completeness, even though PCF-MVS, TAPA-MVS and PLC also aim at improving completeness. These methods assume that the pixels with similar colors may come from the same surface, however, they only use the information from one view to estimate the surface while our method fuses the information from all neighbor views together.

Although we already feed some good initial values to MVS, it still may fail to find out the precise depths in some textureless regions since the matching costs of the accurate depths may not be the locally optimal solution at all. Existing methods with high completeness are all suffered from this deficit as demonstrated in Table 5. An outstanding MVS algorithm should achieve a trade-off between completeness and accuracy and that is why F_1 score is also used for the overall measurement. We present the accuracy, completeness and F_1 scores of our method and its competitors in Table 5. It is observed that our method has no obvious advantage of accuracy under the $2cm$ tolerance, but we achieve competitive accuracy under the $10cm$ tolerance. Considering both accuracy and completeness, our method outperforms other state-of-the-art methods in terms of F_1 score, where Figure 9 shows the F_1 score for each sample of the test set.

5. Conclusion

In this paper, we have presented a mesh-guided MVS method that can handle textureless regions well and achieve high completeness without much loss of accuracy. We adopt a pyramid structure and treat the depth map estimation as the process of sculpturing. To maintain the completeness, we first reconstruct the surface mesh using the depth maps at the coarsest scale and then utilize this mesh to guide the MVS process. Specifically, we project the mesh onto each view to complete and refine the corresponding depth map. To avoid misguidance of wrong depths brought by erroneous faces in the mesh, we also design a network "MG-Conf" to predict the confidence of the depth map. We remove the depth values with bad confidence and feed the rest to the finer scale to get estimates with higher accuracy. In addition, an untextured region detector and enforcing geometric consistency are used to help avoid tremendous shifts of depths in textureless regions. Experimental results show that our method significantly promotes the performance of MVS on the ETH3D dataset.

References

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[2] Caroline Baillard and Andrew Zisserman. A plane-sweep strategy for the 3d reconstruction of buildings from multiple images. *International Archives of Photogrammetry and Remote Sensing*, 33(B2; PART 2):56–62, 2000.

[3] D Cernea. Openmvs: Open multiple view stereovision. `https://github.com/cdcseacave/openMVS`, 2015.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Robert 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. IEEE, 1996.

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[7] 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.

[8] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[9] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[10] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1586–1594, 2017.

[11] 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.

[12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014.

[13] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *arXiv preprint arXiv:1905.02706*, 2019.

[14] Sunok Kim, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. Laf-net: Locally adaptive fusion networks for stereo confidence estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 205–214, 2019.

[15] Sunok Kim, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Unified confidence estimation networks for robust stereo matching. *IEEE Transactions on Image Processing*, 28(3):1299–1313, 2018.

[16] 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.

[17] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (ToG)*, volume 26, page 96. ACM, 2007.

[18] Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, and Helmut Mayer. A tv prior for high-quality scalable multi-view stereo reconstruction. *International Journal of Computer Vision*, 124(1):1–16, 2017.

[19] Andreas Kuhn, Shan Lin, and Oliver Erdler. Plane completion and filtering for multi-view stereo reconstruction. In *German Conference on Pattern Recognition*, pages 18–32. Springer, 2019.

[20] Patrick Labatut, J-P Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. In *Computer graphics forum*, volume 28, pages 2275–2290, 2009.

[21] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 781–796, 2018.

[22] Jie Liao, Yanping Fu, Qingan Yan, and Chunxia Xiao. Pyramid multi-view stereo with local consistency. In *Computer Graphics Forum*, volume 38, pages 335–346. Wiley Online Library, 2019.

[23] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10452–10461, 2019.

[24] 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*, pages 6778–6787, 2019.

[25] 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.

[26] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[27] Despoina Paschalidou, 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.

[28] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *Pro-*

ceedings of the IEEE International Conference on Computer Vision, pages 10413–10422, 2019.

[29] 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.

[30] Thomas Schops, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. Eth3d benchmark. `https://www.eth3d.net`.

[31] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[32] Shuhan Shen. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE transactions on image processing*, 22(5):1901–1914, 2013.

[33] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[34] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017.

[35] Fabio Tosi, Matteo Poggi, Antonio Benincasa, and Stefano Mattoccia. Beyond local reasoning for stereo confidence estimation with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 319–334, 2018.

[36] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7484–7493, 2019.

[37] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5483–5492, 2019.

[38] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4312–4321, 2019.

[39] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[40] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[41] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4595–4603, 2017.

[42] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014.