

# Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching

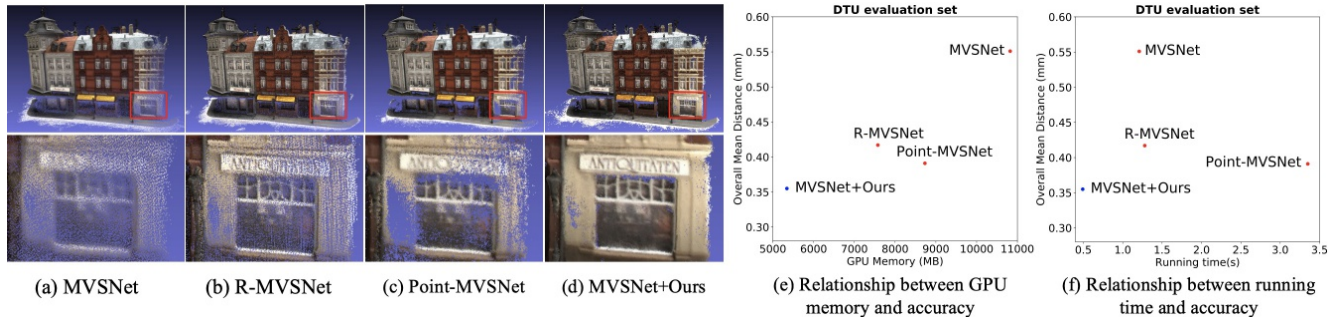
Xiaodong Gu<sup>1\*</sup>Zhiwen Fan<sup>1\*</sup>Siyu Zhu<sup>1</sup>Zuozhuo Dai<sup>1</sup>Feitong Tan<sup>12†</sup>Ping Tan<sup>12</sup><sup>1</sup>Alibaba A.I. Labs<sup>2</sup>Simon Fraser University

Figure 1: Comparison between the state-of-the-art learning-based multi-view stereo approaches [4, 52, 53] and MVSNet+Ours. (a)-(d): Reconstructed point clouds of MVSNet [52], R-MVSNet [53], Point-MVSNet [4] and MVSNet+Ours. (e) and (f): The relationship between reconstruction accuracy and GPU memory or run-time. The resolution of input images is  $1152 \times 864$ .

## Abstract

The deep multi-view stereo (MVS) and stereo matching approaches generally construct 3D cost volumes to regularize and regress the depth or disparity. These methods are limited with high-resolution outputs since the memory and time costs grow cubically as the volume resolution increases. In this paper, we propose a memory and time efficient cost volume formulation complementary to existing multi-view stereo and stereo matching approaches based on 3D cost volumes. First, the proposed cost volume is built upon a feature pyramid encoding geometry and context at gradually finer scales. Then, we can narrow the depth (or disparity) range of each stage by the prediction from the previous stage. With gradually higher cost volume resolution and adaptive adjustment of depth (or disparity) intervals, the output is recovered in a coarser to fine manner.

We apply the cascade cost volume to the representative MVS-Net, and obtain a 35.6% improvement on DTU benchmark (1st place), with 50.6% and 59.3% reduction in GPU memory and run-time. It is also rank first on Tanks and Temples benchmark of all deep models. The statistics of accuracy, run-time and GPU memory on other representative stereo CNNs also validate the effectiveness of our proposed method. Our source code is available at <https://github.com/alibaba/cascade-stereo>.

\*Equal contribution.

†This work was done during an internship at Alibaba A.I. Labs.

## 1. Introduction

Convolutional neural networks (CNNs) have been widely adopted in 3D reconstruction and broader computer vision tasks. State-of-the-art multi-view stereo [19, 29, 52, 53] and stereo matching algorithms [3, 15, 22, 33, 46, 56] often compute a 3D cost volume according to a set of hypothesized depth (or disparity) and warped features. 3D convolutions are applied to this cost volume to regularize and regress the final scene depth (or disparity).

Compared with the methods based on 2D CNNs [30, 55], the 3D cost volume can capture better geometry structures, perform photometric matching in 3D space, and alleviate the influence of image distortion caused by perspective transformation and occlusions [4]. However, methods relying on 3D cost volumes are often limited to low-resolution input images (and results), because 3D CNNs are generally time and GPU memory consuming. Typically, these methods downsample the feature maps to formulate the cost volumes at a lower resolution [3, 4, 15, 19, 22, 29, 33, 46, 52, 53, 56], and adopt upsampling [3, 15, 22, 33, 42, 46, 49, 56] or post-refinement [4, 29] to output the final high-resolution result.

Inspired by the previous coarse-to-fine learning-based stereo approaches [8, 9, 11], we present a novel cascade formulation of 3D cost volumes. We start from a feature pyramid to extract multi-scale features which are commonly used in standard multi-view stereo [52] and stereo match-

ing [3, 15] networks. In a coarse-to-fine manner, the cost volume at the early stages is built upon larger scale semantic 2D features with sparsely sampled depth hypotheses, which lead to a relatively lower volume resolution. Subsequently, the later stages use the estimated depth (or disparity) maps from the earlier stages to adaptively adjust the sampling range of depth (or disparity) hypotheses and construct new cost volumes where finer semantic features are applied. This adaptive depth sampling and adjustment of feature resolution ensures the computation and memory resources are spent on more meaningful regions. In this way, our cascade structure can remarkably decrease computation time and GPU memory consumption. The effectiveness of our method can be seen in Figure 1.

We validate our method on both multi-view stereo and stereo matching on various benchmark datasets. For multi-view stereo, our cascade structure achieves the best performance on the DTU dataset [1] at the submission time of this paper, when combined with MVSNNet [52]. It is also the state-of-the-art learning-based method on Tanks and Temples benchmark [24]. For stereo matching, our method reduces the end-point-error (EPE) and GPU memory consumption of GwcNet [15] by about 15.2% and 36.9% respectively.

## 2. Related Work

**Stereo Matching** According to the survey by Scharstein *et al.* [38], a typical stereo matching algorithm contains four steps: matching cost calculation, matching cost aggregation, disparity calculation, and disparity refinement. Local methods [31, 50, 57] aggregate matching costs with neighboring pixels and usually utilize the winner-take-all strategy to choose the optimal disparity. Global methods [17, 23, 43] construct an energy function and try to minimize it to find the optimal disparity. More specifically, works in [23, 43] use belief propagation and semi-global matching [17] to approximate the global optimization with dynamic programming.

In the context of deep neural networks, CNNs based stereo matching methods are first introduced by Zbontar and LeCun [54], in which a convolutional neural network is introduced to learn the similarity measure of small patch pairs. The introduction of the widely used 3D cost volume in stereo is first proposed in GCNet [22], in which the disparity regression step uses the soft argmin operation to figure out the best matching results. PSMNet [3] further introduces pyramid spatial pooling and 3D hourglass networks for cost volume regularization and yields better results. GwcNet [15] modifies the structure of 3D hourglass and introduces group wise correlation to form a group based 3D cost volume. HSM [48] builds a light model for high-resolution images with a hierarchical design. EMCUA [33] introduces an approach for multi-

level context ultra-aggregation. GANet [56] constructs several semi-global aggregation layers and local guided aggregation layers to further improve the accuracy. DeepPruner [5] is a coarse to fine method which proposes a differentiable PatchMatch-based module to predict the pruned search range for each pixel.

Although methods based on 3D cost-volume remarkably boost the performance, they are limited to downsampled cost volumes and rely on interpolation operations to generate high-resolution disparity. Our cascade cost volumes can be combined with these methods to improve the disparity accuracy and GPU memory efficiency.

**Multi-View Stereo** According to the comprehensive survey [12], works in traditional multi-view stereo can be roughly categorised into volumetric methods [20, 21, 25, 41], which estimate the relationship between each voxel and surfaces; point cloud based methods [13, 26], which directly process 3D points to iteratively densify the results; and depth map reconstruction methods [2, 7, 14, 40, 44, 51], which use only one reference and a few source images for single depth map estimation. For large-scale Structure-from-Motion, works in [58, 59] use distributed methods based on distributed motion averaging and global camera consensus.

Recently, learning-based approaches also demonstrate superior performance on multi-view stereo. Multi-patch similarity [16] introduces a learned cost metric. SurfaceNet [20] and DeepMVS [18] pre-warp the multi-view images to 3D space and use deep networks for regularization and aggregation. Most recently, multi-view stereo based on 3D cost volumes have been proposed in [4, 6, 10, 19, 29, 52, 53]. A 3D cost volume is built based on warped 2D image features from multiple views and 3D CNNs are applied for cost regularization and depth regression. Because the 3D CNNs require large GPU memory, these methods generally use downsampled cost volumes. Our cascade cost volume can be easily integrated into these methods to enable high-resolution cost volumes and further boosts accuracy, computational speed, and GPU memory efficiency.

**High-Resolution Output in Stereo and MVS** Recently, some learning-based methods try to reduce the memory requirement in order to generate high resolution outputs. Instead of using voxel grids, Point MVSNNet [4] proposes to use a small cost volume to generate the coarse depth and uses a point-based iterative refinement network to output the full resolution depth. In comparison, a standard MVSNNet combined with our cascade cost volume can output full resolution depth with superior accuracy using less runtime and GPU memory than Point MVSNNet [4]. Works in [35, 45] partition advanced space to reduce memory consumption and construct a fixed cost volume representation which lacks flexibility. Works in [29, 42, 49] build extra refinement module by 2D CNNs and output a high resolution

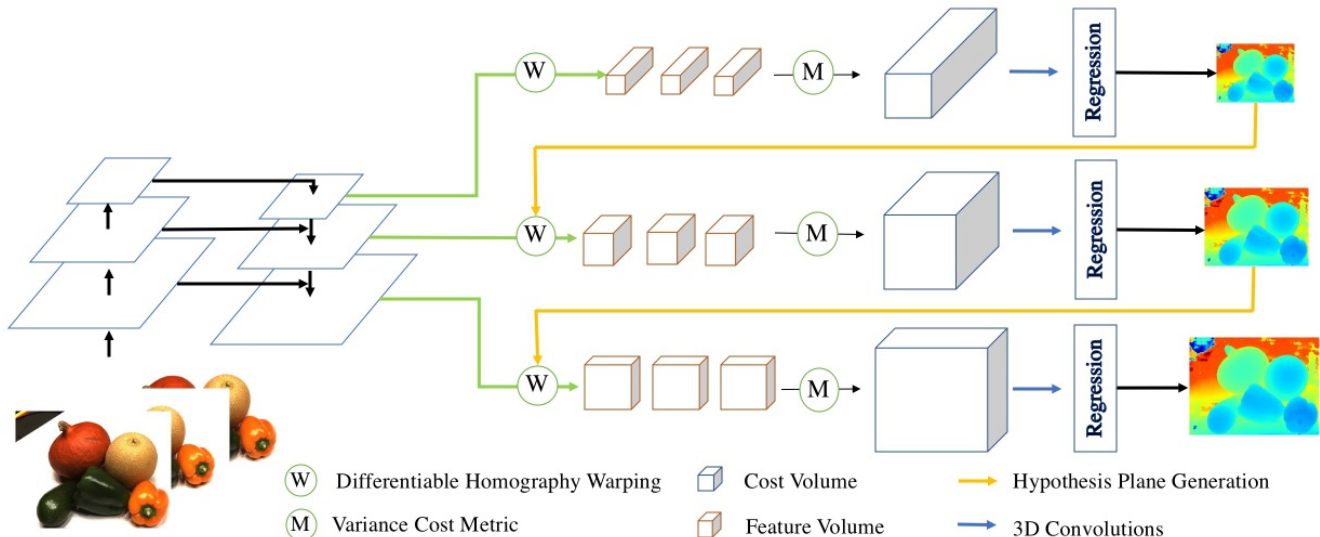


Figure 2: Network architecture of the proposed cascade cost volume on MVSNet [52], denoted as MVSNet+Ours.

prediction. Notably, such refinement modules can be utilized jointly with our proposed cascade cost volume.

### 3. Methodology

This section describes the detailed architecture of the proposed cascade cost volume which is complementary to the existing 3D cost volume based methods in multi-view stereo and stereo matching. Here, we use the representative MVSNet [52] and PSMNet [3] as the backbone networks to demonstrate the application of the cascade cost volume in multi-view stereo and stereo matching tasks respectively. Figure 2 shows the architecture of MVSNet+Ours.

#### 3.1. Cost volume Formulation

Learning-based multi-view stereo [4, 52, 53] and stereo matching [3, 15, 22, 54, 56] construct 3D cost volumes to measure the similarity between corresponding image patches and determine whether they are matched. Constructing 3D cost volume requires three major steps in both multi-view stereo and stereo matching. First, the discrete hypothesis depth (or disparity) planes are determined. Then, we warp the extracted 2D features of each view to the hypothesis planes and construct the feature volumes, which are finally fused together to build the 3D cost volume. Pixel-wise cost calculation is generally ambiguous in inherently ill-posed regions such as occlusion areas, repeated patterns, textureless regions, and reflective surfaces. To solve this, 3D CNNs at multiple scales are generally introduced to aggregate contextual information and regularize the possibly noise-contaminated cost volumes.

**3D Cost Volumes in Multi-View Stereo** MVSNet [52] proposes to use fronto-parallel planes at different depth as

hypothesis planes and the depth range is generally determined by the sparse reconstruction. The coordinate mapping is determined by the homography:

$$H_i(d) = K_i \cdot R_i \cdot \left( I - \frac{(t_1 - t_i) \cdot n_1^T}{d} \right) \cdot R_1^T \cdot K_1^{-1} \quad (1)$$

where  $H_i(d)$  refers to the homography between the feature maps of the  $i^{th}$  view and the reference feature maps at depth  $d$ . Moreover,  $K_i, R_i, t_i$  refers to the camera intrinsics, rotations and translations of the  $i^{th}$  view respectively, and  $n_1$  denotes the principle axis of the reference camera. Then differentiable homography is used to warp 2D feature maps into hypothesis planes of the reference camera to form feature volumes. To aggregate multiple feature volumes to one cost volume, the variance-based cost metric is proposed to adapt an arbitrary number of input feature volumes.

**3D Cost Volumes in Stereo Matching** PSMNet [3] uses disparity levels as hypothesis planes and the range of disparity is designed according to specific scenes. Since the left and right images have been rectified, the coordinate mapping is determined by the offset in the x-axis direction:

$$C_r(d) = X_l - d \quad (2)$$

where  $C_r(d)$  refers to the transformed x-axis coordinate of the right view at disparity  $d$ , and  $X_l$  is the source x-axis coordinate of the left view. To build feature volumes, we warp the feature maps of the right view to the left view using the translation along the x-axis. There are multiple ways to build the final cost volume. GCNet [22] and PSMNet [3] concatenate the left feature volume and the right feature volume without decreasing the feature dimension. The work [55] uses the sum of absolute differences to compute matching cost. DispNetC [30] computes full correlation about the left feature volume and right feature volume

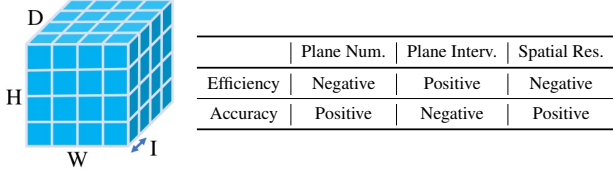


Figure 3: **Left:** the standard cost volume.  $D$  is the number of hypothesis planes,  $W \times H$  is the spatial resolution and  $I$  is the plane interval. **Right:** The influence factors of efficiency (run-time and GPU memory) and accuracy.

and produces only a single-channel correlation map for each disparity level. GwcNet [15] proposes group-wise correlation by splitting the features into groups and computing correlation maps in each group.

### 3.2. Cascade Cost Volume

Figure 3 shows a standard cost volume of a resolution of  $W \times H \times D \times F$ , where  $W \times H$  denotes the spatial resolution,  $D$  is the number of plane hypothesis, and  $F$  is the channel number of feature maps. As mentioned in [4, 52, 53], an increased number of plane hypothesis  $D$ , a larger spatial resolution  $W \times H$ , and a finer plane interval are likely to improve the reconstruction accuracy. However, the GPU memory and run-time grow cubically as the resolution of the cost volume increases. As demonstrated in RMVSNet [53], MVSNet [52] is able to process a maximum cost volume of  $H \times W \times D \times F = 1600 \times 1184 \times 256 \times 32$  on a 16 GB Tesla P100 GPU. To resolve the problems above, we propose a cascade cost volume formulation and predict the output in a coarse-to-fine manner.

**Hypothesis Range** As shown in Figure 4, the depth (or disparity) range of the first stage denoted by  $R_1$  covers the entire depth (or disparity) range of the input scene. In the following stages, we can base on the predicted output from the previous stage, and narrow the hypothesis range. Consequently, we have  $R_{k+1} = R_k \cdot w_k$ , where  $R_k$  is the hypothesis range at the  $k^{th}$  stage and  $w_k < 1$  is the reducing factor of hypothesis range.

**Hypothesis Plane Interval** We also denote the depth (or disparity) interval at the first stage as  $I_1$ . Compared with the commonly adopted single cost volume formulation [3, 52], the initial hypothesis plane interval is comparatively larger to generate a coarse depth (or disparity) estimation. In the following stages, finer hypothesis plane intervals are applied to recover more detailed outputs. Therefore, we have:  $I_{k+1} = I_k \cdot p_k$ , where  $I_k$  is the hypothesis plane interval at the  $k^{th}$  stage and  $p_k < 1$  is the reducing factor of hypothesis plane interval.

**Number of Hypothesis Planes** At the  $k^{th}$  stage, given the hypothesis range  $R_k$  and hypothesis plane interval  $I_k$ , the corresponding number of hypothesis planes  $D_k$  is determined by the equation:  $D_k = R_k / I_k$ . When the spatial

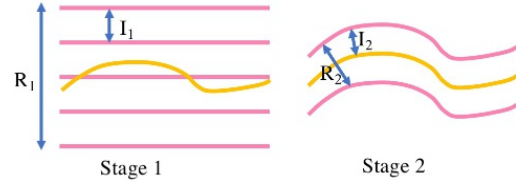


Figure 4: Illustration of hypothesis plane generation.  $R_k$  and  $I_k$  are respectively the hypothesis range and the hypothesis plane number at the  $k^{th}$  stage. Pink lines are hypothesis planes. Yellow line indicates the predicted depth (or disparity) map from stage 1, which is used to determine the hypothesis range and hypothesis plane intervals at stage 2.

resolution of a cost volume is fixed, a larger  $D_k$  generates more hypothesis planes and correspondingly more accurate results while leads to increased GPU memory and run-time. Based on the cascade formulation, we can effectively reduce the total number of hypothesis planes since the hypothesis range is remarkably reduced stage by stage while still covering the entire output range.

**Spatial Resolution** Following the practices of Feature Pyramid Network [28], we double the spatial resolution of the cost volume at every stage along with the doubled resolution of the input feature maps. We define  $N$  as the total stage number of cascade cost volume, then the spatial resolution of cost volume at the  $k^{th}$  stage is defined as  $\frac{W}{2^{N-k}} \times \frac{H}{2^{N-k}}$ . We set  $N = 3$  in multi-view stereo tasks and  $N = 2$  in stereo matching tasks.

**Warping Operation** Applying the cascade cost volume formulation to multi-view stereo, we base on Equation 1 and rewrite the homography warping function at the  $(k+1)^{th}$  stage as:

$$H_i(d_k^m + \Delta_{k+1}^m) = K_i \cdot R_i \cdot \left( I - \frac{(t_1 - t_i) \cdot n_1^T}{d_k^m + \Delta_{k+1}^m} \right) \cdot R_1^T \cdot K_1^{-1} \quad (3)$$

where  $d_k^m$  denotes the predicted depth of the  $m^{th}$  pixel at the  $k^{th}$  stage, and  $\Delta_{k+1}^m$  is the residual depth of the  $m^{th}$  pixel to be learned at the  $k+1^{th}$  stage.

Similarly in stereo matching, we reformulate Equation 2 based on our cascade cost volume. The  $m^{th}$  pixel coordinate mapping at the  $k+1^{th}$  stage is expressed as:

$$C_r(d_k^m + \Delta_{k+1}^m) = X_l - (d_k^m + \Delta_{k+1}^m) \quad (4)$$

where  $d_k^m$  denotes the predicted disparity of the  $m^{th}$  pixel at the  $k^{th}$  stage, and  $\Delta_{k+1}^m$  denotes the residual disparity of the  $m^{th}$  pixel to be learned at the  $k+1^{th}$  stage.

### 3.3. Feature Pyramid

In order to obtain high-resolution depth (or disparity) maps, previous works [29, 33, 46, 56] generally generate a comparatively low-resolution depth (or disparity) map using the standard cost volume and then upsample and refine

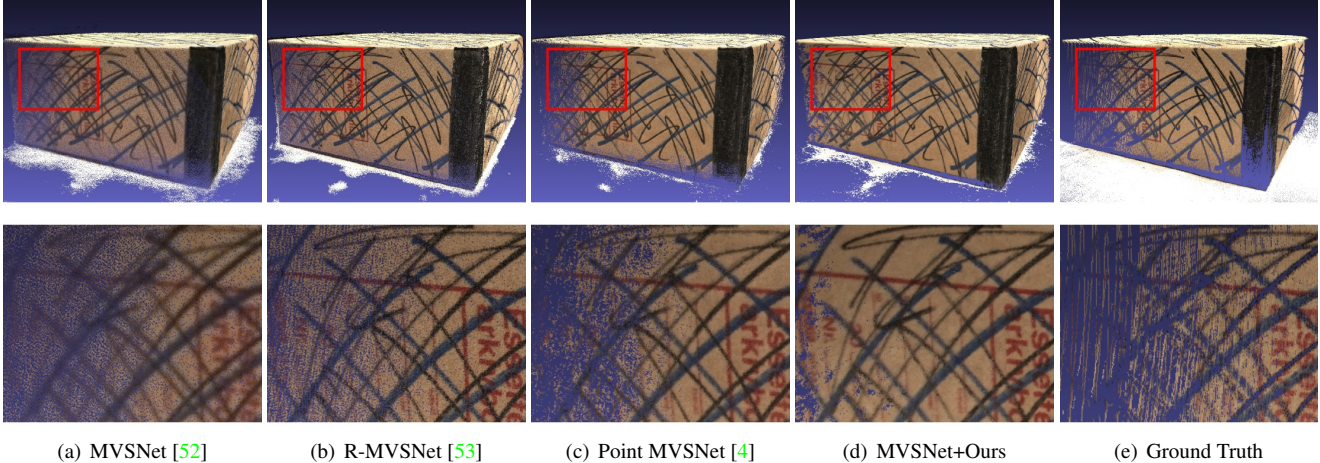


Figure 5: Multi-view stereo qualitative results of scan 10 on DTU dataset [1]. **Top row:** Generated point clouds of different methods and ground truth point clouds. **Bottom row:** Zoomed local areas.

it with 2D CNNs. The standard cost volume is constructed using the top level feature maps which contains high-level semantic features but lacks low-level finer representations. Here, we refer to Feature Pyramid Network [28] and adopt its feature maps with increased spatial resolutions to build the cost volumes of higher resolutions. For example, when applying cascade cost volume to MVSNet [52], we build three cost volumes from the feature maps {P1, P2, P3} of Feature Pyramid Network [28]. Their corresponding spatial resolutions are {1/16, 1/4, 1} of the input image size.

### 3.4. Loss Function

The cascade cost volume with  $N$  stages produces  $N - 1$  intermediate outputs and a final prediction. We apply the supervision to all the outputs and the total loss is defined as:

$$Loss = \sum_{k=1}^N \lambda^k \cdot L^k \quad (5)$$

where  $L^k$  refers to the loss at the  $k^{th}$  stage and  $\lambda^k$  refers to its corresponding loss weight. We adopt the same loss function  $L^k$  as the baseline networks in our experiments.

## 4. Experiments

We evaluate the proposed cascade cost volume on multi-view stereo and stereo matching tasks.

### 4.1. Multi-view stereo

**Datasets** DTU [1] is a large-scale MVS dataset consisting of 124 different scenes scanned in 7 different lighting conditions at 49 or 64 positions. Tanks and Temples dataset [24] contains realistic scenes with small depth ranges. More specifically, its intermediate set is consisted of 8 scenes including Family, Francis, Horse, Lighthouse, M60, Panther, Playground, and Train. Following the work [53], we

Methods	Acc.(mm)	Comp.(mm)	Overall(mm)	GPU Mem(MB)	Run-time(s)
Camp [2]	0.835	0.554	0.695	-	-
Furu [13]	0.613	0.941	0.777	-	-
Tola [44]	0.342	1.190	0.766	-	-
Gipuma [14]	<b>0.283</b>	0.873	0.578	-	-
SurfaceNet [20]	0.450	1.040	0.745	-	-
R-MVSNet [53]	0.383	0.452	0.417	7577	1.28
P-MVSNet [29]	0.406	0.434	0.420	-	-
Point-MVSNet [4]	0.342	0.411	0.376	8731	3.35
MVSNet(D=192) [52]	0.456	0.646	0.551	10823	1.210
MVSNet+Ours	0.325	0.385	0.355	5345	0.492
Comp. with MVSNet	28.7%	40.4%	35.6%	50.6%	59.3%

Table 1: Multi-view stereo quantitative results of different methods on DTU dataset [1] (lower is better). We conduct this experiment using two resolution settings according to PointMVSNet [4] where MVSNet+Ours uses resolution of  $1152 \times 864$ .

use DTU training set [1] to train our method, and test on DTU evaluation set. To validate the generalization of our approach, we also test it on the intermediate set of Tanks and Temples dataset [24] using the model trained on DTU dataset without fine-tuning.

**Implementation** We apply the proposed cascade cost volume to the representative MVSNet [52] and denote the network as **MVSNet+Ours**. During training, we set the number of input images to  $N=3$  and image resolution to  $640 \times 512$ . After balancing accuracy and efficiency, we adopt a three-stage cascade cost volume. From the first to the third stage, the number of depth hypothesis is 48, 32 and 8, and the corresponding depth interval is set to 4, 2 and 1 times as the interval of MVSNet [52] respectively. Accordingly, the spatial resolution of feature maps gradually increases and is set to 1/16, 1/4 and 1 of the original input image size. We follow the same input view selection and data pre-processing strategies as MVSNet [52] in both training and evaluation. During training, we use Adam optimizer with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . The training is done for 16 epochs with an initial learning rate of 0.001, which is downscaled by a factor of 2 after 10, 12, and 14 epochs. We

	Rank	Mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
COLMAP [39, 40]	54.62	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
R-MVSNet [53]	40.12	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
Point-MVSNet [4]	38.12	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
ACMH [47]	15.00	54.82	69.99	49.45	45.12	59.04	52.64	52.37	58.34	51.61
P-MVSNet [29]	12.25	55.62	70.04	44.64	40.22	<b>65.20</b>	55.08	<b>55.17</b>	<b>60.37</b>	<b>54.29</b>
MVSNet [52]	52.00	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
MVSNet+Ours	<b>9.50</b>	<b>56.42</b>	<b>76.36</b>	<b>58.45</b>	<b>46.20</b>	55.53	<b>56.11</b>	54.02	58.17	46.56

Table 2: Statistical results on the Tanks and Temples dataset [24] of state-of-the-art multi-view stereo and our methods.



Figure 6: Point cloud results of MVSNet+Ours on the intermediate set of Tanks and Temples dataset [24].

Stages	Resosution	>2mm(%)	>8mm(%)	Overall (mm)	GPU Mem. (MB)	Run-time (s)
1	1/4 × 1/4	0.310	0.163	0.602	2373	0.081
2	1/2 × 1/2	0.208	0.084	0.401	4093	0.243
3	1	0.174	0.077	0.355	5345	0.492

Table 3: The statistical results of different stages in cascade cost volume. The statistics are collected on the DTU evaluation set [1] using MVSNet+Ours. The run-time is the sum of the current and previous stages. The base of resolution of input images in this experiment is  $1152 \times 864$ .

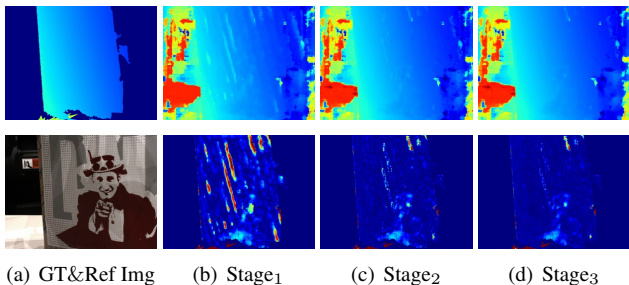


Figure 7: Reconstruction results of each stage. **Top row:** Ground truth depth map and intermediate reconstructions. **Bottom row:** Error maps of intermediate reconstructions.

train our method with 8 Nvidia GTX 1080Ti GPUs with 2 training samples on each GPU.

For quantitative evaluation on DTU dataset [1], we calculate the accuracy and the completeness by the MATLAB code provided by DTU dataset [1]. The percentage evaluation is implemented following MVSNet [52]. The F-score is used as the evaluation metric for Tanks and Temples dataset [24] to measure the accuracy and completeness

of the reconstructed point clouds. We use fusibile [36] as our post-processing consisting of three steps: photometric filtering, geometric consistency filtering, and depth fusion.

**Benchmark Performance** Quantitative results on DTU evaluation set [1] are shown in Table 1. We can see that MVSNet [52] with cascade cost volume outperforms other methods [4, 29, 52, 53] in both completeness and overall quality and rank the 1st place on DTU dataset [1], with the improvement of 35.6%, and the decrease of memory, run-time reduction of 50.6% and 59.3%. The qualitative results are shown in Figure 5. We can see that MVSNet+Ours generates more complete point clouds with finer details. Besides, we demonstrate the generalization ability of our trained model by testing on Tanks and Temples dataset [24]. The corresponding quantitative results are reported in Table 2, and MVSNet+Ours achieves the state-of-the-art performance among the learning-based multi-view stereo methods. The qualitative point cloud results of the intermediate set of Tanks and Temples benchmark [24] are visualized in Figure 6. Note that, we get the results of above mentioned methods by running their provided pre-trained model and code except R-MVSNet [53] which provides point cloud results with their post-processing method.

To analyse the accuracy, GPU memory and run-time at each stage, we evaluate the MVSNet+Ours method on the DTU dataset [1]. We provide comprehensive statistics in Table 3 and visualization results in Figure 7. In a coarse-to-fine manner, the overall quality is improved from 0.602 to

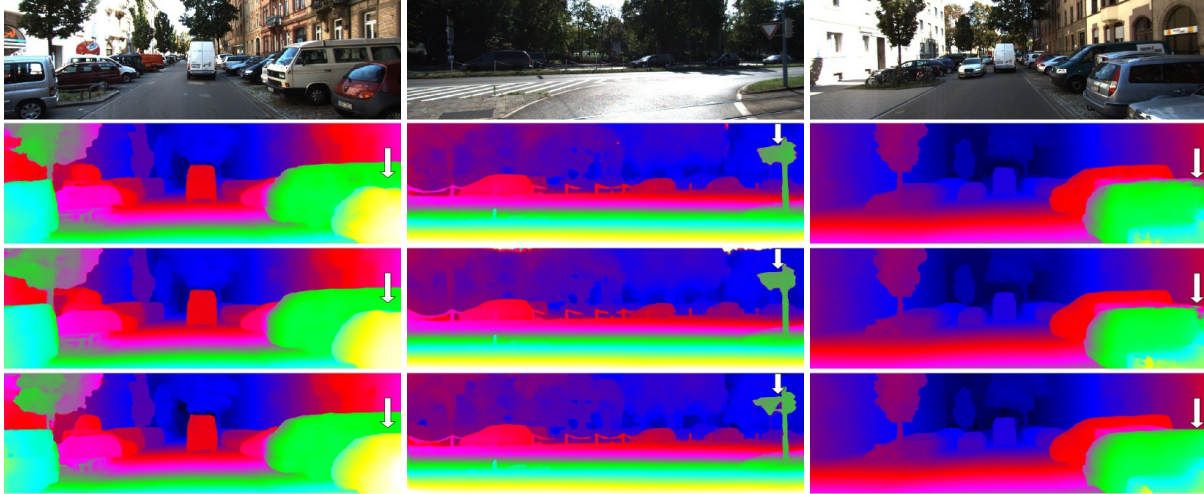


Figure 8: Qualitative results on the test set of KITTI2015 [32]. **Top row:** Input images, **Second row:** Results of PSMNet [3]. **Third row:** Results of GwcNet [15]. **Bottom row:** Results of GwcNet with cascade cost volume (GwcNet+Ours).

0.355. Accordingly, the GPU memory increases from 2,373 MB to 4,093 MB and 5,345 MB, and run-time increases from 0.081 s to 0.243 s and 0.492 s.

## 4.2. Stereo Matching

**Datasets** Scene Flow dataset [30] is a large scale-dataset containing 35,454 training and 4,370 testing stereo pairs of size  $960 \times 540$ . It contains accurate ground truth disparity maps. We use the Finalpass of the Scene Flow dataset [30] since it contains more motion blur and defocus and is more like a real-world environment. KITTI 2015 [32] is a real-world dataset with dynamic street views. It contains 200 training pairs and 200 testing pairs. Middlebury [37] is the publicly available dataset for high-resolution stereo matching contains 60 pairs under imperfect calibration, different exposures, and different lighting conditions.

**Implementation** In Scene Flow dataset, we extend PSMNet [3], GwcNet [15] and GANet11 [56] with our proposed cascade cost volume and denote them as **PSMNet+Ours**, **GwcNet+Ours** and **GANet11+Ours**. Balancing the trade-off between accuracy and efficiency, a two-stage cascade cost volume is applied, and the number of disparity hypothesis is 12. The corresponding disparity interval is set to 4 and 1 pixels respectively. The spatial resolution of feature maps increases from  $1/16$  to  $1/4$  of the original input image size. The maximum disparity is set to 192.

In KITTI 2015 benchmark [32], we mainly compare GwcNet [15] and GwcNet+Ours. For a fair comparison, we follow the training details of the original networks. The evaluation metric in Scene Flow dataset [30] is end-point-error (EPE), which is the mean absolute disparity error in pixels. For KITTI 2015 [32], the percentage of disparity outliers  $D1$  is used to evaluate disparity error larger than

	>1px	>2px.	>3px	EPE	Mem.
PSMNet [3]	9.46	5.19	3.80	0.887	6871
PSMNet+Ours	7.44	4.61	3.50	0.721	4124
GwcNet [15]	8.03	4.47	3.30	0.765	7277
GwcNet+Ours	7.46	4.16	3.04	0.649	4585
GANet11 [56]	-	-	-	0.95	6631
GANet11+Ours	11.0	5.97	4.28	0.90	5032

Table 4: Quantitative results of different stereo matching methods with and without cascade cost volume on Scene Flow dataset [30]. Accuracy, GPU memory consumption and run-time are included for comparisons.

Methods	All (%)			Noc (%)		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
DispNetC [30]	4.32	4.41	4.34	4.11	3.72	4.05
GC-Net [22]	2.21	6.16	2.87	2.02	5.58	2.61
CRL [34]	2.48	<b>3.59</b>	2.67	2.32	<b>3.12</b>	2.45
iResNet-i2e2 [27]	2.14	3.45	2.36	1.94	3.20	2.15
SegStereo [49]	1.88	4.07	2.25	1.76	3.70	2.08
PSMNet [3]	1.86	4.62	2.32	1.71	4.31	2.14
GwcNet [15]	1.74	3.93	2.11	1.61	3.49	1.92
GwcNet+Ours	<b>1.59</b>	4.03	<b>2.00</b>	<b>1.43</b>	3.55	<b>1.78</b>

Table 5: Comparison of different stereo matching methods on KITTI2015 benchmark [32].

$\max(3px, 0.05d^*)$ , where  $d^*$  denotes the ground-truth disparity.

**Benchmark Performance** Quantitative results of different stereo methods on Scene Flow dataset [30] is shown in Table 4. By applying the cascade 3D cost volume, we boost the accuracy in all the metrics and less memory is required owing to the cascade design with smaller number of disparity hypothesis. Our method reduces the end-point-error by 0.166, 0.116 and 0.050 on PSMNet [3] (0.887 vs. 0.721), GwcNet [15] (0.765 vs. 0.649) and GANet11 [56] (0.950 vs. 0.900) respectively. The obvious improvement on  $>1px$  indicates that small errors are suppressed with the introduction of high-resolution cost volumes. In KITTI 2015 [32],

	Depth Num.	Depth Interv.	Acc.	Comp.	Overall
MVSNet	192	1	0.4560	0.6460	0.5510
MVSNet-Cas <sub>2</sub>	96, 96	2, 1	<b>0.4352</b>	0.4275	0.4314
MVSNet-Cas <sub>3</sub>	96, 48, 48	2, 2, 1	0.4479	<b>0.4141</b>	<b>0.4310</b>
MVSNet-Cas <sub>4</sub>	96, 48, 24, 24	2, 2, 2, 1	0.4354	0.4374	0.4364
MVSNet-Cas <sub>3</sub> -share	96, 48, 48	2, 2, 1	0.4741	0.4282	0.4512

Table 6: Comparisons between MVSNet [52] and MVSNet using our cascade cost volume with different setting of depth hypothesis numbers and depth intervals. The statistics are collected on DTU dataset [1].

Methods	cascade?	upsample?	feature pyramid?	Acc. (mm)	Comp. (mm)	Overall (mm)
MVSNet	×	×	×	0.456	0.646	0.551
MVSNet-Cas <sub>3</sub>	✓	×	×	0.450	0.455	0.453
MVSNet-Cas <sub>3</sub> -Ups	✓	✓	×	0.419	0.338	0.379
MVSNet+Ours	✓	×	✓	0.325	0.385	0.355

Table 7: The quantitative comparison between MVSNet and MVSNet with different settings of the cascade cost volumes. Specifically, "cascade" denotes that the original cost volume is divided into three cascade cost volumes, "upsample" denotes cost volumes with increased spatial resolutions by bilinear upsampling corresponding feature maps, and "feature pyramid" denotes cost volumes with higher spatial resolutions built on pyramid feature maps. The statistics are evaluated on the DTU dataset.

Table 5 shows the percentage of disparity outliers  $D1$  evaluated for background, foreground, and all pixels. Compared with the original GwcNet [15], the rank of GwcNet+Ours rises from 29<sup>th</sup> to 17<sup>th</sup> (date: Nov.5, 2019). Several disparity estimation on KITTI 2015 test set [32] is shown in Figure 8. In Middlebury benchmark, PSMNet+Ours ranks 37th for the avgerr metric(date: Feb.7, 2020).

### 4.3. Ablation Study

Extensive ablation studies are performed to validate the improved accuracy and efficiency of our approach. All results are obtained by the three-stage model on DTU validation set [1] unless otherwise stated.

**Cascade Stage Number** The quantitative results with different stage numbers are summarized in Table 6. In our implementation, we use MVSNet [52] with 192 depth hypothesis as the baseline model, and replace its cost volume with our cascade design which is also consisted of 192 depth hypothesis. Note that the spatial resolution of different stages are the same as that of the original MVSNet [52]. This extended MVSNet is denoted as **MVSNet-Cas<sub>i</sub>** where  $i$  indicates the total stage number. We find that as the number of stages increases, the overall quality first remarkably increases and then stabilizes.

**Spatial Resolution** Then, we study how the spatial resolution of a cost volume  $W \times H$  affects the reconstruction performance. Here, we compare MVSNet-Cas<sub>3</sub>, which contains 3 stages and all the stages share the same spatial

resolution, and MVSNet-Cas<sub>3</sub>-Ups where the spatial resolution increases from 1/16 to 1 of the original image size and bilinear interpolation is used to upsample feature maps. As shown in Table 7, the overall quality of MVSNet+Ours is obviously superior to those of MVSNet-Cas<sub>3</sub> (0.453 vs. 0.355). Accordingly, a higher spatial resolution also leads to increased GPU memory (2373 vs. 5345 MB) and run-time (0.322 vs. 0.492 seconds).

**Feature Pyramid** As shown in Table 7, the cost volume constructed from Feature Pyramid Network [28] denoted by MVSNet+Ours can slightly improve the overall quality from 0.379 to 0.355. The GPU memory (6227 vs. 5345 MB) and run-time (0.676 vs. 0.492 seconds) are also decreased. Compared with the improvement between MVSNet-Cas<sub>3</sub> and MVSNet-Cas<sub>3</sub>-Ups, the increased spatial resolution is still more critical to the improvement of reconstruction accuracy.

**Parameter Sharing in Cost Volume Regularization** We also analyze the effect of weight sharing in 3D cost volume regularization across all the stages. As is shown in Table 6, the shared parameters cascade cost volume denoted by MVSNet-Cas<sub>3</sub>-share achieves worse performance than MVSNet-Cas<sub>3</sub>. It indicates that separate parameter learning of the cascade cost volumes at different stages further improves the accuracy.

### 4.4. Run-time and GPU Memory

Table 1 shows the comparison of GPU memory and run-time between MVSNet [52] with and without cascade cost volume. Given the remarkable accuracy improvement, the GPU memory decreases from 10,823 to 5,345 MB, and the run-time drops from 1.210 to 0.492 seconds. In Table 4, we compare the GPU memory between PSMNet [3], GwcNet [15] and GANet11 [56] with and without the proposed cascade cost volume. The GPU memory of PSMNet [3], GwcNet [15] and GANet11 [56] decreases by 39.97%, 36.99% and 24.11% respectively.

## 5. Conclusion

In this paper, we present a both GPU memory and computationally efficient cascade cost volume formulation for high-resolution multi-view stereo and stereo matching. First, we decompose the single cost volume into a cascade formulation of multiple stages. Then, we can narrow the depth (or disparity) range of each stage and reduce the total number of hypothesis planes by utilizing the depth (or disparity) map from the previous stage. Next, we use the cost volumes of higher spatial resolution to generate the outputs with finer details. The proposed cost volume is complementary to existing 3D cost-volume-based multi-view stereo and stereo matching approaches.



## References

- [1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 2016, 120(2):153–168, 2016. [2](#), [5](#), [6](#), [8](#)
- [2] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008, pages 766–779. Springer, 2008. [2](#), [5](#)
- [3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018, pages 5410–5418, 2018. [1](#), [2](#), [3](#), [4](#), [7](#), [8](#)
- [4] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, 2019, 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#)
- [5] Duggal et al. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *ICCV*, 2019, pages 4384–4393, 2019. [2](#)
- [6] Hou et al. Multi-view stereo by temporal nonparametric fusion. In *ICCV2019*, pages 2651–2660, 2019. [2](#)
- [7] Romanoni et al. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *ICCV2019*, pages 10413–10422, 2019. [2](#)
- [8] Tonioni et al. Real-time self-adaptive deep stereo. In *CVPR2019*, pages 195–204, 2019. [1](#)
- [9] Wang et al. Anytime stereo image depth estimation on mobile devices. In *ICRA2019*, pages 5893–5900. IEEE, 2019. [1](#)
- [10] Xue et al. Myscrf: Learning multi-view stereo with conditional random fields. In *ICCV2019*, pages 4312–4321, 2019. [2](#)
- [11] Yin et al. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR2019*, pages 6044–6053, 2019. [1](#)
- [12] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *CGV*, 9(1-2):1–148, 2015. [2](#)
- [13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32(8):1362–1376, 2009. [2](#), [5](#)
- [14] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015, pages 873–881, 2015. [2](#), [5](#)
- [15] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019, pages 3273–3282, 2019. [1](#), [2](#), [3](#), [4](#), [7](#), [8](#)
- [16] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *ICCV*, 2017, pages 1586–1594, 2017. [2](#)
- [17] Heiko Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005, volume 2, pages 807–814. IEEE, 2005. [2](#)
- [18] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018, pages 2821–2830, 2018. [2](#)
- [19] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: end-to-end deep plane sweep stereo. *arXiv:1905.00538*, 2019. [1](#), [2](#)
- [20] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfnet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2017, pages 2307–2315, 2017. [2](#), [5](#)
- [21] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NeurIPS*, 2017, pages 365–376, 2017. [2](#)
- [22] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017, pages 66–75, 2017. [1](#), [2](#), [3](#), [7](#)
- [23] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006, volume 3, pages 15–18. IEEE, 2006. [2](#)
- [24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *TOG*, 36(4):78, 2017. [2](#), [5](#), [6](#)
- [25] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000. [2](#)
- [26] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *TPAMI*, 27(3):418–433, 2005. [2](#)
- [27] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *CVPR*, 2018, pages 2811–2820, 2018. [7](#)
- [28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017, pages 2117–2125, 2017. [4](#), [5](#), [8](#)
- [29] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *ICCV*, 2019, October 2019. [1](#), [2](#), [4](#), [5](#), [6](#)
- [30] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016, pages 4040–4048, 2016. [1](#), [3](#), [7](#)
- [31] Xing Mei, Xun Sun, Weiming Dong, Haitao Wang, and Xiaopeng Zhang. Segment-tree based cost aggregation for stereo matching. In *CVPR*, 2013, pages 313–320, 2013. [2](#)
- [32] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015, pages 3061–3070, 2015. [7](#), [8](#)
- [33] Guang-Yu Nie, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang. Multi-level context ultra-aggregation for stereo matching. In *CVPR*, 2019, pages 3283–3291, 2019. [1](#), [2](#), [4](#)
- [34] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *ICCV*, 2017, pages 887–895, 2017. [7](#)
- [35] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, 2017, pages 3577–3586, 2017. [2](#)

- [36] K. Lasinger S. Galliani and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. <https://github.com/kysucix/fusibile/>. 6
- [37] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 7
- [38] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002. 2
- [39] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR, 2016*, pages 4104–4113, 2016. 6
- [40] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV, 2016*, pages 501–518. Springer, 2016. 2, 6
- [41] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):151–173, 1999. 2
- [42] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *ACCV, 2018*, pages 20–35. Springer, 2018. 1, 2
- [43] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *TPAMI*, (7):787–800, 2003. 2
- [44] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *MVA*, 23(5):903–920, 2012. 2, 5
- [45] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *TOG*, 36(4):72, 2017. 2
- [46] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *ICCV, 2019*, October 2019. 1, 4
- [47] Qingshan Xu and Wenbing Tao. Multi-view stereo with asymmetric checkerboard propagation and multi-hypothesis joint view selection. *arXiv:1805.07920*, 2018. 6
- [48] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *CVPR, 2019*, pages 5515–5524, 2019. 2
- [49] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV, 2018*, pages 636–651, 2018. 1, 2, 7
- [50] Qingxiong Yang. A non-local cost aggregation method for stereo matching. In *CVPR, 2012*, pages 1402–1409. IEEE, 2012. 2
- [51] Yao Yao, Shiwei Li, Siyu Zhu, Hanyu Deng, Tian Fang, and Long Quan. Relative camera refinement for accurate dense reconstruction. In *3DV, 2017*, pages 185–194. IEEE, 2017. 2
- [52] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV, 2018*, pages 767–783, 2018. 1, 2, 3, 4, 5, 6, 8
- [53] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR, 2019*, pages 5525–5534, 2019. 1, 2, 3, 4, 5, 6
- [54] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR, 2015*, pages 1592–1599, 2015. 2, 3
- [55] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 17:1–32, 2016. 1, 3
- [56] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR, 2019*, pages 185–194, 2019. 1, 2, 3, 4, 7, 8
- [57] Ke Zhang, Jiangbo Lu, and Gauthier Lafreuit. Cross-based local stereo matching using orthogonal integral images. *TCSVT*, 19(7):1073–1079, 2009. 2
- [58] Runze Zhang, Siyu Zhu, Tian Fang, and Long Quan. Distributed very large scale bundle adjustment by global camera consensus. In *ICCV, 2017*, pages 29–38, 2017. 2
- [59] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *CVPR, 2018*, pages 4568–4577, 2018. 2