Cost Volume Pyramid Based Depth Inference for Multi-View Stereo

Jiayu Yang[1], Wei Mao[1], Jose M. Alvarez[2], Miaomiao Liu[1,3]
[1]Australian National University, [2]NVIDIA, [3]Australian Centre for Robotic Vision
{jiayu.yang, wei.mao, miaomiao.liu}@anu.edu.au, josea@nvidia.com

R-MVSNet [43] Point-MVSNet [5] Ours

Figure 1: Point clouds reconstructed by state-of-the-art methods [5, 43] and our CVP-MVSNet. Best viewed on screen.

Abstract

We propose a cost volume-based neural network for depth inference from multi-view images. We demonstrate that building a cost volume pyramid *in a coarse-to-fine manner instead of constructing a cost volume at a fixed resolution leads to a compact, lightweight network and allows us inferring high resolution depth maps to achieve better reconstruction results. To this end, we first build a cost volume based on uniform sampling of fronto-parallel planes across the entire depth range at the coarsest resolution of an image. Then, given current depth estimate, we construct new cost volumes iteratively on the pixelwise depth residual to perform depth map refinement. While sharing similar insight with* Point-MVSNet *as predicting and refining depth iteratively, we show that working on* cost volume pyramid *can lead to a more compact, yet efficient network structure compared with the* Point-MVSNet *on 3D points. We further provide detailed analyses of the relation between (residual) depth sampling and image resolution, which serves as a principle for building compact* cost volume pyramid*. Experimental results on benchmark datasets show that our model can perform* 6x *faster and has similar performance as state-of-the-art methods. Code is available at* https://github.com/JiayuYANG/CVP-MVSNet.

1. Introduction

Multi-view stereo (MVS) aims to reconstruct the 3D model of a scene from a set of images captured by a camera from multiple viewpoints. It is a fundamental problem for computer vision community and has been studied extensively for decades [35]. While traditional methods before deep learning era have great achievements on the reconstruction of a scene with Lambertian surfaces, they still suffer from illumination changes, low-texture regions, and reflections resulting in unreliable matching correspondences for further reconstruction.

Recent learning-based approaches [5, 42, 43] adopt deep CNNs to infer the depth map for each view followed by a separate multiple-view fusion process for building 3D models. These methods allow the network to extract discriminative features encoding global and local information of a scene to obtain robust feature matching for MVS. In particular, Yao *et al*. propose MVSNet [42] to infer a depth map for each view. An essential step in [42] is to build a cost volume based on a plane sweep process followed by multi-scale 3D CNNs for regularization. While effective in depth inference accuracy, its memory requirement is cubic to the image resolution. To allow handling high resolution images, they then adopt a recurrent cost volume regularization process [43]. However, the reduction in memory requirements involves a longer run-time.

In order to achieve a computationally efficient network, Chen *et al*. [5] work on 3D point clouds to iteratively predict the depth residual along visual rays using *edge convolutions* operating on the k nearest neighbors of each 3D point. While this approach is efficient, its run-time increases almost linearly with the number of iteration levels.

In this work, we propose a *cost volume pyramid* based Multi-View Stereo Network (CVP-MVSNet) for depth inference. In our approach, we first build an image pyramid for each input image. Then, for the coarsest resolution of the reference image, we build a compact cost volume by sampling the depth across the entire depth-range of a scene. After that, at the next pyramid level, we perform residual depth search from the neighbor of the current depth estimate to construct a *partial cost volume* using multi-scale 3D CNNs for regularization. As we build these cost volumes iteratively with a short search range at each level, it leads to a small and compact network. As a result, our network performs 6x faster than current state-of-the-art networks on benchmark datasets.

While it is noteworthy that we share the similar insight with [5] as predicting and refining the depth map in a coarse-to-fine manner, our work differs from theirs in the following four main aspects. First, the approach in [5] performs convolutions on 3D point cloud. Instead, we construct cost volumes on a regular grid defined on the image coordinates, which is shown to be faster in run-time. Second, we provide a principle for building a compact *cost volume pyramid* based on the correlation between depth sampling and image resolution. As third main difference, we use a multi-scale 3D-CNN regularization to cover large receptive field and encourage local smoothness on residual depth estimates which, as shown in Fig. 1, leads to a better accuracy. Finally, in contrast to [5] and other related works, our approach can output depth of small resolution with small resolution image.

In summary, our main contributions are

- We propose a cost-volume based, compact, and computational efficient depth inference network for MVS.
- We build a *cost volume pyramid* in a coarse-to-fine manner based on a detailed analysis of the relation between the depth residual search range and the image resolution,
- Our framework can handle high resolution images with less memory requirement, is 6x faster than the current state-of-the-art framework, i.e. Point-MVSNet [5], and achieves a better accuracy on benchmark datasets.

2. Related Work

Traditional Multi-View Stereo. Multi-view stereo has been extensively studied for decades. We refer to algorithms before deep learning era as traditional MVS methods which represent the 3D geometry of objects or scene using voxels [7, 23], level-sets [9, 31], polygon meshes [8, 10] or depth maps [18, 22]. In the following, we mainly focus on discussions about volumetric and depth-based MVS methods which have been integrated to learning-based framework recently.

Volumetric representations can model most of the objects or scenes. Given a fixed volume of an object or scene, volumetric-based methods first divide the whole volume into small voxels and then use a photometric consistency metric to decide whether the voxel belongs to the surface or not. These methods do not impose constraints on the shape of the objects. However, the space discretisation is memory intensive. By contrast, depth-map based MVS methods have shown more flexibility in modeling the 3D geometry of scene [28]. Readers are referred to [35] for detailed discussions. Similar to other recent learning-based approaches, we adopt depth map representation in our framework.

Deep learning-based MVS. Deep CNNs have significantly advanced the progress of high-level vision tasks, such as image recognition [15, 36], object detection [13, 32], and semantic segmentation [4, 25]. As for 3D vision tasks, learning-based approaches have been widely adopted to solve stereo matching problems and have achieved very promising results [2, 27, 44]. However, these learning-based approaches cannot be easily generalized to solve MVS problems as rectifications are required for the multiple view scenario which may cause the loss of information [42].

More recently, a few approaches have proposed to directly solve MVS problems [14, 24, 29]. For instance, Ji *et al*. [17] introduce the first learning based pipeline for MVS. This approach learns the probability of voxels lying on the surface. Concurrently, Kar *et al*. [19] present a learnable system to up-project pixel features to the 3D volume and classify whether a voxel is occupied or not by the surface. These systems provide promising results. However, they use volumetric representations that are memory expensive and therefore, these algorithms can not handle large-scale scenes.

Large-scale scene reconstruction has been approached by Yao *et al*. in [42]. The authors propose to learn the depth map for each view by constructing a cost volume followed by 3D CNN regularization. Then, they obtain the 3D geometry by fusing the estimated depth maps from multiple views. The algorithm uses cost volume with memory requirements cubic to the image resolution. Thus, it can not leverage all the information available in high-resolution images. To circumvent this problem, the algorithm adopts GRUs [43] to regularize the cost volume in a sequential manner. As a result, the algorithm reduces the memory requirement but leads to increased run-time.

Closely related work to ours is Point-MVSNet [5]. Point-MVSNet is a framework to predict the depth in a coarse-to-fine manner working directly on point cloud. It allows the aggregation of information from its k nearest neighbors in 3D space. Our approach shares similar insight

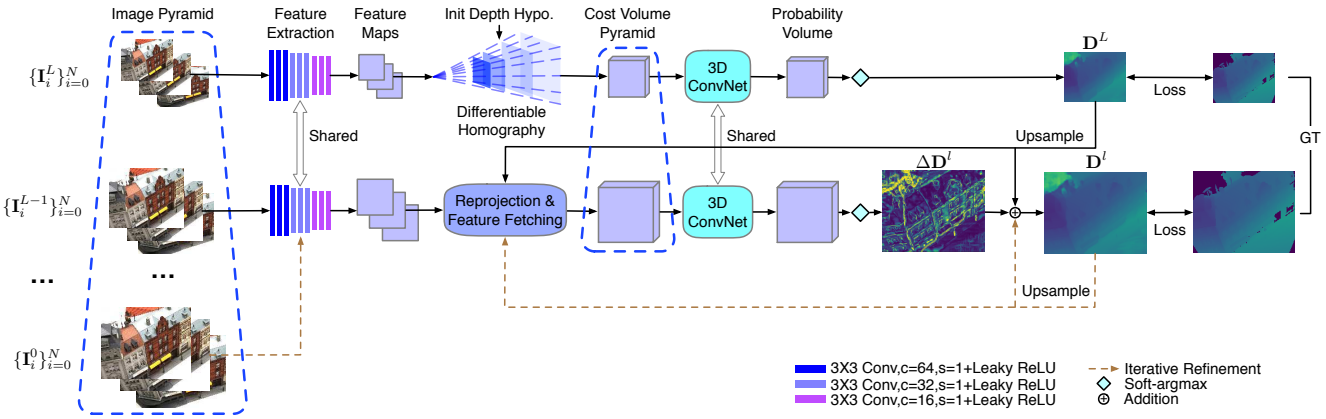


Figure 2: Network Structure. Reference and source images are first downsampled to form an image pyramid. We apply *feature extraction network* to all levels and images to extract feature maps. We then build the *cost volume pyramid* in a coarse-to-fine manner. Specifically, we start with the construction of a cost volume corresponding to coarsest image resolution followed by building *partial cost volumes* iteratively for depth residual estimation in order to achieve depth map $\mathbf{D} = \mathbf{D}^0$ for $\mathbf{I}$. Please refer to Fig. 3 for details about re-projection, feature fetching and building cost volume.

as predicting and refining depth maps iteratively. However, we differ from Point-MVSNet in a few key aspects: Instead of working on 3D, we build the cost volume on the regular image grid. Inspired by the idea of *partial cost volume* used in PWC-Net [37] for optical flow estimation, we build *partial cost volume* to predict depth residuals. We compare the memory and computational efficiency with [5]. It shows that our cost-volume pyramid based network leads to more compact and accurate models that run much faster for a given depth-map resolution.

Cost volume. Cost volume is widely used in traditional methods for dense depth estimation from unrectified images [6, 41]. However, most recent learning-based works build cost volume at a fixed resolution [16, 42, 43], which leads to high memory requirement for handling high resolution images. Recently, Sun *et al*. [37] introduced the idea of *partial cost volume* for optical flow estimation. In short, given an estimate of the current optical flow, the *partial cost volume* is constructed by searching correspondences within a rectangle around its position locally in the *warped* source view image. Inspired by such strategy, in this paper, we propose *cost volume pyramid* as an algorithm to progressively estimate the depth residual for each pixel along its visual ray. As we will demonstrate in our experiments, constructing cost volumes at multiple levels leads to a more effective and efficient framework.

3. Method

Let us now introduce our approach to depth inference for MVS. The overall system is depicted in Fig. 2. As existing works, we assume the reference image is denoted as $\mathbf{I}_0 \in \mathbb{R}^{H \times W}$, where H and W define its dimensions. Let $\{\mathbf{I}_i\}_{i=1}^N$ be its N neighboring source images. Assume $\{\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}_{i=0}^N$ are the corresponding camera intrinsics, rotation matrix, and translation vector for all views. Our goal is to infer the depth map $\mathbf{D}$ for $\mathbf{I}_0$ from $\{\mathbf{I}_i\}_{i=0}^N$. The key novelty of our approach is using a feed-forward deep network on *cost volume pyramid* constructed in a coarse-to-fine manner. Below, we introduce our feature pyramid, the *cost volume pyramid*, depth map inference and finally provide details of the loss function.

3.1. Feature Pyramid

As raw images vary with illumination changes, we adopt learnable features, which has been demonstrated to be crucial step for extracting dense feature correspondences [42, 38]. The general practice in existing works is to make use of high resolution images to extract multi-scale image features even for the output of a low resolution depth map. By contrast, we show that a low resolution image contains enough information useful for estimating a low resolution depth map.

Our feature extraction pipeline consists of two steps, see Fig. 2. First, we build the $(L+1)$-level image pyramid $\{\mathbf{I}_i^j\}_{j=0}^L$ for each input image, $i \in \{0, 1, \cdots, N\}$, where the bottom level of the pyramid corresponds to the input image, $\mathbf{I}_i^0 = \mathbf{I}_i$. Second, we obtain feature representations at the l-th level using a CNN, namely *feature extraction network*. Specifically, it consists of 9 convolutional layers, each of which is followed by a leaky rectified linear unit (LeakyReLU). We use the same CNN to extract features for all levels in all the images. We denote the feature maps for a given level l by $\{\mathbf{f}_i^l\}_{i=0}^N, \mathbf{f}_i^l \in \mathbb{R}^{H/2^l \times W/2^l \times F}$, where $F = 16$ is the number of feature channels used in our experiments.

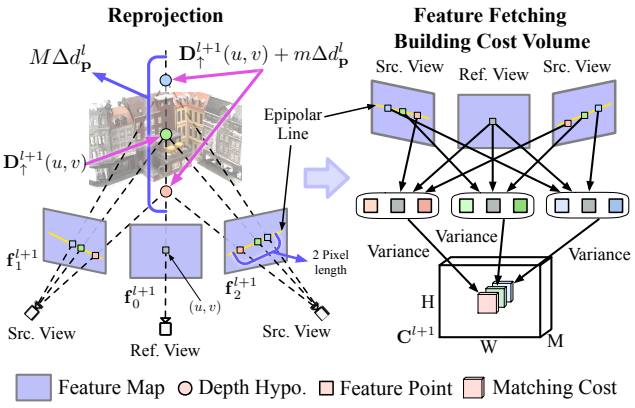


Figure 3: Reprojection, feature fetching and building cost volume. Left: We define M depth hypotheses for each pixel (u, v) in the reference view. By projecting them to each source view, we can fetch M corresponding features. Right: For each depth hypothesis, the matching cost is the variance of fetched features across source views and the reference view. The cost volume $\mathbf{C}^{l+1}$ defines matching costs for all depth hypotheses of all pixels in the reference view.

We will show that, compared to existing works, our feature extraction pipeline leads to significant reduction in memory requirements and, at the same time, improve performance.

3.2. Cost Volume Pyramid

Given the extracted features, the next step is to construct cost volumes for depth inference in the reference view. Common approaches usually build a single cost volume at a fixed resolution [16, 42, 43], which incurs in large memory requirements and thus, limits the use of high-resolution images. Instead, we propose to build a *cost volume pyramid*, a process that iteratively estimates and refines depth maps to achieve high resolution depth inference. More precisely, we first build a cost volume for coarse depth map estimation based on images of the coarsest resolution in image pyramids and uniform sampling of the fronto-parallel planes in the scene. Then, we construct partial cost volumes based on coarse estimation and depth residual hypotheses iteratively to achieve depth maps with higher resolution and accuracy. We provide details about these two steps below.

Cost Volume for Coarse Depth Map Inference. We start building a cost volume at the Lth level corresponding to the lowest image resolution $(H/2^L, W/2^L)$. Assume depth measured at the reference view of a scene ranges from d_{min} to d_{max}. We construct the cost volume for the reference view by sampling M fronto-parallel planes uniformly across entire depth range. A sampled depth $d = d_{\text{min}} + m(d_{\text{max}} - d_{\text{min}})/M, m \in \{0, 1, 2, \cdots, M-1\}$ represents a plane where its normal $\mathbf{n}_0$ is the principal axis of the reference camera.

Similar to [42], we define the differentiable homography $\mathbf{H}_i(d)$ between the ith source view and the reference view at depth d as

$$\mathbf{H}_i(d) = \mathbf{K}_i^L \mathbf{R}_i (\mathbf{I} - \frac{(\mathbf{t}_0 - \mathbf{t}_i)\mathbf{n}_0^T}{d}) \mathbf{R}_0^{-1} (\mathbf{K}_0^L)^{-1}, \quad (1)$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{K}_i^L$ and $\mathbf{K}_0^L$ are the scaled intrinsic matrices of $\mathbf{K}_i$ and $\mathbf{K}_0$ at level L.

Each homography $\mathbf{H}_i(d)$ suggests a possible pixel correspondence between $\tilde{\mathbf{x}}_i$ in source view i and a pixel $\mathbf{x}$ in the reference view. This correspondence is defined as $\lambda_i \tilde{\mathbf{x}}_i = \mathbf{H}_i(d)\mathbf{x}$, where λ_i represents the depth of $\tilde{\mathbf{x}}_i$ in the source view i.

Given $\tilde{\mathbf{x}}_i$ and $\{\mathbf{f}_i^L\}_{i=1}^N$, we use differentiable bilinear interpolation to reconstruct the feature map warped to the reference view $\{\tilde{\mathbf{f}}_{i,d}^L\}_{i=1}^N$. The cost for all pixels at depth d is defined as its variance of features from $N+1$ views,

$$\mathbf{C}_d^L = \frac{1}{(N+1)} \sum_{i=0}^{N} (\tilde{\mathbf{f}}_{i,d}^L - \bar{\mathbf{f}}_d^L)^2, \quad (2)$$

where $\tilde{\mathbf{f}}_{0,d}^L = \mathbf{f}_0^L$ is the feature map of the reference image and $\bar{\mathbf{f}}_d^L$ is the average of feature volumes across all views $(\{\tilde{\mathbf{f}}_{i,d}^L\}_{i=1}^N \cup \mathbf{f}_0^L)$ for each pixel. This metric encourages that the correct depth for each pixel has the smallest feature variance, which corresponds to the photometric consistency constraint. We compute the cost map for each depth hypothesis and concatenate those cost maps to a single cost volume $\mathbf{C}^L \in \mathbb{R}^{W/2^L \times H/2^L \times M \times F}$.

A key parameter to obtain good depth estimation accuracy is the depth sampling resolution M. We will show in Section 3.3 how to determine the interval for depth sampling and coarse depth estimation.

Cost Volume for Multi-scale Depth Residual Inference. Recall that our ultimate goal is to obtain $\mathbf{D} = \mathbf{D}^0$ for $\mathbf{I}_0$. We iterate starting from $\mathbf{D}^{l+1}$, a given depth estimate for the $(l+1)$th level, to obtain a refined depth map for the next level $\mathbf{D}^l$ until reaching the bottom level. More precisely, we first upsample $\mathbf{D}^{l+1}$ to the next level $\mathbf{D}_{\uparrow}^{l+1}$ via bicubic interpolation and then, we build the *partial cost volume* to regress the *residual depth map* defined as $\Delta\mathbf{D}^l$ to obtain a refined depth map $\mathbf{D}^l = \mathbf{D}_{\uparrow}^{l+1} + \Delta\mathbf{D}^l$ at the lth level.

While we share the similar insight with [5] to iteratively predict the depth residual, we argue that instead of performing convolutions on a point cloud [5], building the regular 3D cost volume on the depth residual followed by multi-scale 3D convolution can lead to a more compact, faster, and higher accuracy depth inference. Our motivation is that depth displacements for neighboring pixels are correlated which indicates that regular multi-scale 3D convolution would provide useful contextual information for depth residual estimation. We therefore arrange the depth displacement hypotheses in a regular 3D space and compute the cost volume as follows.

Assume we are given camera parameters $\{\mathbf{K}_i^l, \mathbf{R}_i, \mathbf{t}_i\}_{i=0}^N$ for all camera views and the upsam-

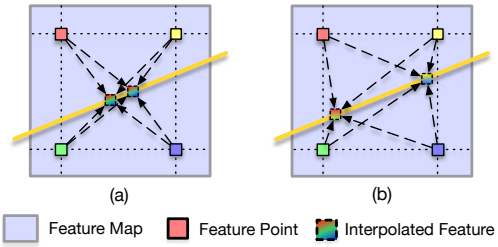


Figure 4: Interpolation of two sampling points from four feature points in source view. (a) Densely sampled depth will result in very close (< 0.5 pixel) locations which have similar feature. (b) Points projected using appropriate depth sampling carry distinguishable information.

pled depth estimate $\mathbf{D}_{\uparrow}^{l+1}$. Current depth estimate for each pixel $\mathbf{p} = (u, v)$ is defined as $d_{\mathbf{p}} = \mathbf{D}_{\uparrow}^{l+1}(u, v)$. Let each depth residual hypothesis interval be $\Delta d_{\mathbf{p}} = s_{\mathbf{p}}/M$, where $s_{\mathbf{p}}$ represents the depth search range at $\mathbf{p}$ and M denotes the number of sampled depth residual. We consider the projection of corresponding hypothesized 3D point with depth $(\mathbf{D}_{\uparrow}^{l+1}(u, v) + m\Delta d_{\mathbf{p}})$ into view i as

$$\lambda_i \mathbf{x}_i' = \mathbf{K}_i^l(\mathbf{R}_i\mathbf{R}_0^{-1}((\mathbf{K}_0^l)^{-1}(u, v, 1)^T(d_{\mathbf{p}} + m\Delta d_{\mathbf{p}}) - \mathbf{t}_0) + \mathbf{t}_i), \tag{3}$$

where λ_i denotes the depth of corresponding pixel in view i, and $m \in \{-M/2, \cdots, M/2 - 1\}$ (see Fig. 3). Then, the cost for that pixel at each depth residual hypothesis is similarly defined based on Eq. 2, which leads to a partial cost volume $\mathbf{C}^l \in \mathbb{R}^{H/2^l \times W/2^l \times M \times F}$.

In the next section, we introduce our solution to determine the depth search intervals and range for all pixels, $s_{\mathbf{p}}$, which is essential to obtain accurate depth estimates.

3.3. Depth Map Inference

In this section, we first provide details to perform depth sampling at the coarsest image resolution and discretisation of the local depth search range at higher image resolution for building the cost volume. Then, we introduce depth map estimators on cost volumes to achieve the depth map inference.

Depth Sampling for Cost Volume Pyramid We observe that the depth sampling for virtual depth planes is related to the image resolution. As shown in Fig. 4, it is not necessary to sample depth planes densely as projections of those sampled 3D points in the image are too close to provide extra information for depth inference. In our experiments, to determine the number of virtual planes, we compute the mean depth sampling interval for a corresponding 0.5 pixel distance in the image.

For determining the local search range for depth residual around the current depth estimate for each pixel, we first project its 3D point into source views, find points that are two pixels away from the its projection along the epipolar line in both directions(see Fig. 3 "2 pixel length"), and back project those two points into 3D rays. The intersection of these two rays with the visual ray in the reference view determines the search range for depth refinement on current level.

Depth Map Estimator Similar to MVSNet [42], we apply 3D convolution to the constructed cost volume pyramid $\{\mathbf{C}^l\}_{l=0}^L$ to aggregate context information and output probability volumes $\{\mathbf{P}^l\}_{l=0}^L$, where $\mathbf{P}^l \in \mathbb{R}^{H/2^l \times W/2^l \times M}$. Detailed 3D convolution network design is in Supp. Mat. Note that $\mathbf{P}^L$ and $\{\mathbf{P}^l\}_{l=0}^{L-1}$ are generated on absolute and residual depth, respectively. We therefore first apply soft-argmax to $\mathbf{P}^L$ to obtain the coarse depth map. Then, we iteratively refine the obtained depth map by applying soft-argmax to $\{\mathbf{P}^l\}_{l=1}^{L-1}$ to obtain the depth residual for higher resolutions.

Recall that sampled depth is $d = d_{\text{min}} + m(d_{\text{max}} - d_{\text{min}})/M, m \in \{0, 1, 2, \cdots, M-1\}$ at level L. Therefore, the depth estimate for each pixel $\mathbf{p}$ is computed as

$$\mathbf{D}^L(\mathbf{p}) = \sum_{m=0}^{M-1} d\mathbf{P}_{\mathbf{p}}^L(d). \tag{4}$$

To further refine the current estimate which is either the coarse depth map or a refined depth at $(l+1)$th level, we estimate the *residual depth*. Assume $r_{\mathbf{p}} = m \cdot \Delta d_{\mathbf{p}}^l$ denotes the depth residual hypothesis. We compute the updated depth at the next level as

$$\mathbf{D}^l(\mathbf{p}) = \mathbf{D}_{\uparrow}^{l+1}(\mathbf{p}) + \sum_{m=-M/2}^{(M-2)/2} r_{\mathbf{p}}\mathbf{P}_{\mathbf{p}}^l(r_{\mathbf{p}}) \tag{5}$$

where $l \in \{L-1, L-2, \cdots, 0\}$. In our experiments, we observe no depth map refinement after our pyramidal depth estimation is further required to obtain good results.

3.4. Loss Function

We adopt a supervised learning strategy and construct the pyramid for ground truth depth $\{\mathbf{D}_{GT}^l\}_{l=0}^L$ as supervisory signal. Similar to existing MVSNet framework [42], we make use of the l_1 norm measuring the absolute difference between the ground truth and the estimated depth. For each training sample, our loss is

$$Loss = \sum_{l=0}^{L} \sum_{\mathbf{p} \in \Omega} \|\mathbf{D}_{GT}^l(\mathbf{p}) - \mathbf{D}^l(\mathbf{p})\|_1, \tag{6}$$

where Ω is the set of valid pixels with ground truth measurements.

4. Experiments

In this section, we demonstrate the performance of our framework for MVS with a comprehensive set of experiments in standard benchmarks. Below, we first describe the datasets and benchmarks and then analyze our results.

4.1. Datasets

DTU Dataset [1] is a large-scale MVS dataset with 124 scenes scanned from 49 or 64 views under 7 different lighting conditions. DTU provides 3D point clouds acquired using structured-light sensors. Each view consists of an image and the calibrated camera parameters. To train our model, we generate a 160×128 depth map for each view by using the method provided by MVSNet [42]. We use the same training, validation and evaluation sets as defined in [42, 43].

Tanks and Temples [21] contains both indoor and outdoor scenes under realistic lighting conditions with large scale variations. For comparison with other approaches, we evaluate our results on the *intermediate set*.

4.2. Implementation

Training We train our CVP-MVSNet on DTU training set. Unlike previous methods [42, 43] that take high resolution image as input but estimate a depth map of smaller size, our method produces the same size depth map as the input image. For training, we match the ground-truth depth map by downsampling the high resolution image into a smaller one of size 160×128. Then, we build the image and ground truth depth pyramid with 2 levels. To construct the *cost volume pyramid*, we uniformly sample $M = 48$ depth hypotheses across entire depth range at the coarsest (2nd) level. Then, each pixel has $M = 8$ depth residual hypotheses at the next level for the refinement of the depth estimation. Following MVSNet [42], we adopt 3 views for training. We implemented our network using Pytorch [30], and we used ADAM [20] to train our model. The batch size is set to 16 and the network is end-to-end trained on a NVIDIA TITAN RTX graphics card for 27 epochs. The learning rate is initially set to 0.001 and divided by 2 iteratively at the 10^{th},12^{th},14^{th} and 20^{th} epoch.

Metrics. We follow the standard evaluation protocol as in [1, 42]. We report the *accuracy*, *completeness* and *overall score* of the reconstructed point clouds. *Accuracy* is measured as the distance from estimated point clouds to the ground truth ones in millimeter and *completeness* is defined as the distance from ground truth point clouds to the estimated ones [1]. The *overall score* is the average of accuracy and completeness [42].

Evaluation As the parameters are shared across the *cost volume pyramid*, we can evaluate our model with different number of cost volumes and input views. For the evaluation, we set the number of depth sampling, $M = 96$ for the coarsest depth estimation (same as [5]. We also provide results of $M = 48$ in the Supp. Mat.) and $M = 8$ for the following depth residual inference levels. Similar to previous methods [5, 42, 43], we use 5 views and apply the same depth map fusion method to obtain the point clouds. We evaluate our model with images of different size and set the pyramid levels accordingly to maintain a similar size as the input image (80×64) at coarsest level. For instance, for an input size of 1600×1184, the pyramid has 5 levels and 4 levels for an input size of 800×576 and 640×480.

	Method	Acc.	Comp.	Overall (*mm*)
Geometic	Furu[11]	0.613	0.941	0.777
	Tola[39]	0.342	1.190	0.766
	Camp[3]	0.835	0.554	0.695
	Gipuma[12]	**0.283**	0.873	0.578
	Colmap[33, 34]	0.400	0.664	0.532
Learning	SurfaceNet[17]	0.450	1.040	0.745
	MVSNet[42]	0.396	0.527	0.462
	P-MVSNet[26]	0.406	0.434	0.420
	R-MVSNet[43]	0.383	0.452	0.417
	MVSCRF[40]	0.371	0.426	0.398
	Point-MVSNet[5]	0.342	<u>0.411</u>	<u>0.376</u>
	Ours	<u>0.296</u>	**0.406**	**0.351**

Table 1: Quantitative results of reconstruction quality on DTU dataset (lower is better). Our method outperforms all methods on Mean Completeness and Overall reconstruction quality and achieved second best on Mean Accuracy.

4.3. Results on DTU dataset

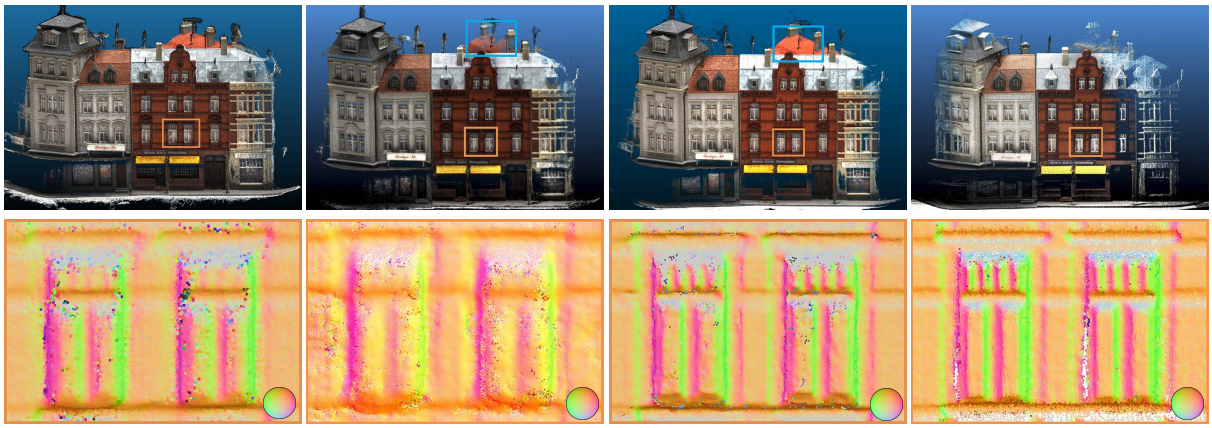
We first compare our results to those reported by traditional geometric-based methods and other learning-based baseline methods. As summarized in Table 1, our method outperforms all current learning-based methods in terms of *accuracy*, *completeness* and *overall score*. Compared to geometric-based approaches, only the method proposed by Galliani *et al*. [12] provides slightly better results in terms of mean *accuracy*.

We now compare our results to related learning based methods in terms of GPU memory usage and runtime for different input resolution. The summary of these results is listed in Table 2. As shown, our network, with a similar memory usage (bottom row), is able to produce better point clouds with lower runtime. In addition, compared to Point-MVSNet [5] on the same size of depth map output (top rows), our approach is six times faster and consumes six times less memory with similar accuracy. We can output high resolution depth map with better accuracy, less memory usage and shorter runtime than Point-MVSNet [5].

Figures 5 and 6 show some qualitative results. As shown, our method is able to reconstruct more details than Point-MVSNet [5], see for instance, the details highlighted in blue box of the roof behind the front building. Compared to R-MVSNet [43] and Point-MVSNet [5], as we can see in the normal maps, our results are smoother on the surfaces while capturing more high-frequency details in edgy areas.

4.4. Results on Tanks and Temples

We now evaluate the generalization ability of our method. To this end, we use the model trained on DTU



R-MVSNet [43] Point-MVSNet [5] Ours Ground truth

Figure 5: Qualitative results of scan 9 of DTU dataset. The upper row shows the point clouds and the bottom row shows the normal map corresponding to the orange rectangle. As highlighted in the blue rectangle, the completeness of our results is better than those provided by Point-MVSNet[5]. The normal map (orange rectangle) further shows that our results are smoother on surfaces while maintaining more high-frequency details.

Method	Input Size	Depth Map Size	Acc.(mm)	Comp.(mm)	Overall(mm)	*f-score*(0.5mm)	GPU Mem(MB)	Runtime(s)
Point-MVSNet[5]	1280x960	640x480	0.361	0.421	0.391	84.27	8989	2.03
Ours-640	640x480	640x480	0.372	0.434	0.403	82.44	**1416**	**0.37**
Point-MVSNet[5]	1600x1152	800x576	0.342	0.411	0.376	-	13081	3.04
Ours-800	800x576	800x576	0.340	0.418	0.379	86.82	**2207**	**0.49**
MVSNet[42]	1600x1152	400x288	0.396	0.527	0.462	78.10	22511	2.76
R-MVSNet[43]	1600x1152	400x288	0.383	0.452	0.417	83.96	**6915**	5.09
Point-MVSNet[5]	1600x1152	800x576	0.342	0.411	0.376	-	13081	3.04
Ours	1600x1152	1600x1152	**0.296**	**0.406**	**0.351**	**88.61**	8795	**1.72**

Table 2: Comparison of reconstruction quality, GPU memory usage and runtime on DTU dataset for different input sizes. GPU memory usage and runtime are obtained by running the official evaluation code of baselines on a same machine with a NVIDIA TITAN RTX graphics card. For the same size of depth maps (Ours-640, Ours-800) and a performance similar to Point-MVSNet [5], our method is 6 times faster and consumes 6 times smaller GPU memory. For the same size of input images (Ours), our method achieves the best reconstruction with the shortest time and a reasonable GPU memory usage.

Method	Rank	Mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
P-MVSNet [26]	**11.72**	**55.62**	70.04	44.64	**40.22**	**65.20**	55.08	**55.17**	**60.37**	**54.29**
Ours	12.75	54.03	**76.5**	**47.74**	36.34	55.12	**57.28**	54.28	57.43	47.54
Point-MVSNet[5]	29.25	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
R-MVSNet[43]	31.75	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
MVSNet[42]	42.75	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69

Table 3: Performance on Tanks and Temples [21] on November 12, 2019. Our results outperform Point-MVSNet [5], which is the strongest baseline on DTU dataset, and are competitive compared to P-MVSNet [26].

without any fine-tuning to reconstruct point clouds for scenes in Tanks and Temples dataset. For fair comparison, we use the same camera parameters, depth range and view selection of MVSNet [42]. For comparison, we consider four baselines [5, 26, 42, 43] and evaluate the *f-score* on Tanks and Temples. Table 3 summarizes these results. As shown, our method yielded a mean *f-score* 5% higher than Point-MVSNet [5], which is the best baseline on DTU dataset, and only 1% lower than P-MVSNet [26]. Note that P-MVSNet [26] applies more depth filtering process for point cloud fusion than ours which just follows the simple fusion process provided by MVSNet [42]. Qualitative results of our point cloud reconstructions are shown in Fig. 7.

4.5. Ablation study

Training pyramid levels. We first analyze the effect of the number of pyramid levels on the quality of the reconstruction. To this end, we downsample the images to form pyramids with four different levels. Results of this analysis are summarized in Table 4a. As shown, the proposed 2-level pyramid is the best. As the level of pyramid increases, the image resolution of the coarsest level decreases. For more than 2-levels, this resolution is too small to produce a good initial depth map to be refined.

R-MVSNet [43] Point-MVSNet [5] Ours Ground truth

Figure 6: Additional results from DTU dataset. Best viewed on screen.

(a) Train (b) Panther

(c) Lighthouse (d) Family (e) Horse

Figure 7: Point cloud reconstruction of Tanks and Temples dataset [21]. Best viewed on screen.

Levels	Coarsest Img. Size	Acc.	Comp.	Overall
2	80x64	**0.296**	**0.406**	**0.351**
3	40x32	0.326	0.407	0.366
4	20x16	0.339	0.411	0.375
5	10x8	0.341	0.412	0.376

(a)

Pixel Interval	Acc. (mm)	Comp. (mm)	Overall (mm)
2	0.299	0.413	0.356
1	0.299	**0.403**	**0.351**
0.5	**0.296**	0.406	**0.351**
0.25	0.313	0.482	0.397

(b)

Table 4: Parameter sensitivity on DTU dataset. a) Accuracy as a function of the number of pyramid levels. b) Accuracy as a function of the interval setting.

Evaluation pixel interval settings. We now analyze the effect of varying the pixel interval setting for depth refinement. As discussed in section 3.3, the depth sampling is determined by the corresponding pixel offset in source views, hence, it is important to set a suitable pixel interval. Table 4b summarizes the effect of varying the interval from depth ranges corresponding to 0.25 pixel to 2 pixels during evaluation. As shown, the performance drops when the interval is too small (0.25 pixel) or too large (2 pixels).

5. Conclusion

In this paper, we proposed CVP-MVSNet, a *cost volume pyramid* based depth inference framework for MVS. CVP-MVSNet is compact, lightweight, fast in runtime and can handle high resolution images to obtain high quality depth map for 3D reconstruction. Our model achieves better performance than state-of-the-art methods by extensive evaluation on benchmark datasets. In the future, we want to explore the integration of our approach into a learning-based structure-from-motion framework to further reduce the memory requirements for different applications.

Acknowledgments

This research is supported by Australian Research Council grants (DE180100628, DP200102274).

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 2016. 6

[2] Konstantinos Batsos, Changjiang Cai, and Philippos Mordohai. CBMV: A coalesced bidirectional matching volume for disparity estimation. *CVPR*, 2018. 2

[3] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *ECCV*, 2008. 6

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 2

[5] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, 2019. 1, 2, 3, 4, 6, 7, 8

[6] Robert T Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996. 3

[7] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *ICCV*, 1999. 2

[8] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 2004. 2

[9] Olivier Faugeras and Renaud Keriven. *Variational principles, surface evolution, PDE's, level set methods and the stereo problem*. 2002. 2

[10] Pascal Fua and Yvan G Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 1995. 2

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *TMAPI*, 2010. 6

[12] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 2016. 6

[13] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[14] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. *ICCV*, 2017. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[16] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-to-end deep plane sweep stereo. In *ICLR*, 2019. 3, 4

[17] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2017. 2, 6

[18] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *CVPR*, 2001. 2

[19] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NeurIPS*, pages 365–376, 2017. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 2017. 6, 7, 8

[22] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, 2002. 2

[23] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *IJCV*, 2000. 2

[24] Vincent Leroy, Jean-Sebastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *ECCV*, 2018. 2

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[26] 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. 6, 7

[27] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *CVPR*, 2016. 2

[28] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007. 2

[29] Despoina Paschalidou, Ali Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. *CVPR*, 2019. 2

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[31] J-P Pons, Renaud Keriven, O Faugeras, and Gerardo Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. In *ICCV*, 2003. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[33] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6

[34] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 6

[35] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 1, 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[37] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3

[38] Chengzhou Tang and Ping Tan. BA-net: Dense bundle adjustment networks. In *ICLR*, 2019. 3

[39] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 2012. 6

[40] 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 *ICCV*, 2019. 6

[41] Ruigang Yang and Marc Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *CVPR*, 2003. 3

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

[43] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019. 1, 2, 3, 4, 6, 7, 8

[44] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 2016. 2