Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness

Shuo Cheng\textsuperscript{1}\textsuperscript{*} Zexiang Xu\textsuperscript{1}\textsuperscript{*} Shilin Zhu\textsuperscript{1}
Zhuwen Li\textsuperscript{2} Li Erran Li\textsuperscript{3,4} Ravi Ramamoorthi\textsuperscript{1} Hao Su\textsuperscript{1}
\textsuperscript{1}University of California, San Diego \textsuperscript{2}Nuro Inc. \textsuperscript{3}Scale AI \textsuperscript{4}Columbia University

Abstract

We present Uncertainty-aware Cascaded Stereo Network (UCS-Net) for 3D reconstruction from multiple RGB images. Multi-view stereo (MVS) aims to reconstruct fine-grained scene geometry from multi-view images. Previous learning-based MVS methods estimate per-view depth using plane sweep volumes (PSVs) with a fixed depth hypothesis at each plane; this requires densely sampled planes for high accuracy, which is impractical for high-resolution depth because of limited memory. In contrast, we propose adaptive thin volumes (ATVs); in an ATV, the depth hypothesis of each plane is spatially varying, which adapts to the uncertainties of previous per-pixel depth predictions. Our UCS-Net has three stages: the first stage processes a small PSV to predict low-resolution depth; two ATVs are then used in the following stages to refine the depth with higher resolution and higher accuracy. Our ATV consists of only a small number of planes with low memory and computation costs; yet, it efficiently partitions local depth ranges within learned small uncertainty intervals. We propose to use variance-based uncertainty estimates to adaptively construct ATVs; this differentiable process leads to reasonable and fine-grained spatial partitioning. Our multi-stage framework progressively sub-divides the vast scene space with increasing depth resolution and precision, which enables reconstruction with high completeness and accuracy in a coarse-to-fine fashion. We demonstrate that our method achieves superior performance compared with other learning-based MVS methods on various challenging datasets.

1. Introduction

Inferring 3D scene geometry from captured images is a core problem in computer vision and graphics with applications in 3D visualization, scene understanding, robotics and autonomous driving. Multi-view stereo (MVS) aims to reconstruct dense 3D representations from multiple images with calibrated cameras. Inspired by the success of deep convolutional neural networks (CNN), several learning-based MVS methods have been presented [23, 27, 54, 20, 47]; the most recent work leverages cost volumes in a learning pipeline [58, 21], and outperforms many traditional MVS methods [13].

At the core of the recent success on MVS [58, 21] is the application of 3D CNNs on plane sweep cost volumes to effectively infer multi-view correspondence. However, such 3D CNNs involve massive memory usage for depth estimation with high accuracy and completeness. In particular, for a large scene, high accuracy requires sampling a large number of sweeping planes and high completeness requires reconstructing high-resolution depth maps. In general, given limited memory, there is an undesired trade-off between accuracy (more planes) and completeness (more pixels) in previous work [58, 21].
Our goal is to achieve highly accurate and highly complete reconstruction with low memory and computation consumption at the same time. To do so, we propose a novel learning-based uncertainty-aware multi-view stereo framework, which utilizes multiple small volumes, instead of a large standard plane sweep volume, to progressively regress high-quality depth in a coarse-to-fine fashion. A key in our method is that we propose novel adaptive thin volumes (ATVs, see Fig. 1) to achieve efficient spatial partitioning.

Specifically, we propose a novel cascaded network with three stages (see Fig. 2): each stage of the cascade predicts a depth map with a different size; each following stage constructs an ATV to refine the predicted depth from the previous stage with higher pixel resolution and finer depth partitioning. The first stage uses a small standard plane sweep volume with low image resolution and relatively sparse depth planes – 64 planes that are fewer than the number of planes (256 or 512) in previous work [58, 59]; the following two stages use ATVs with higher image resolutions and significantly fewer depth planes – only 32 and 8 planes. While consisting of a very small number of planes, our ATVs are constructed within learned local depth ranges, which enables efficient and fine-grained spatial partitioning for accurate and complete depth reconstruction.

This is made possible by the novel uncertainty-aware construction of an ATV. In particular, we leverage the variances of the predicted per-pixel depth probabilities, and infer the uncertainty intervals (as shown in Fig. 1) by calculating variance-based confidence intervals of the per-pixel probability distributions for the ATV construction. Specifically, we apply the previously predicted depth map as a central curved plane, and construct an ATV around the central plane within local per-pixel uncertainty intervals. In this way, we explicitly express the uncertainty of the depth prediction at one stage, and embed this knowledge into the input volume for the next stage.

Our variance-based uncertainty estimation is differentiable and we train our UCSNet from end to end with depth supervision for the predicted depths from all three stages. Our network can thus learn to optimize the estimated uncertainty intervals, to make sure that an ATV is constructed with proper depth coverage that is both large enough – to try to cover ground truth depth – and small enough – to enable accurate reconstruction for the following stages. Overall, our multi-stage framework can progressively sub-divide the local space at a finer scale in a reasonable way, which leads to high-quality depth reconstruction. We demonstrate that our novel UCS-Net outperforms the state-of-the-art learning-based MVS methods on various datasets.

2. Related Work

Multi-view stereo is a long-studied vision problem with many traditional approaches [44, 39, 33, 32, 26, 10, 8, 13, 43]. Our learning-based framework leverages the novel spatial representation, ATV to reconstruct high-quality depth for fine-grain scene reconstruction. In this work, we mainly discuss spatial representation for 3D reconstruction and deep learning based multi-view stereo.

Spatial Representation for 3D Reconstruction. Existing methods can be categorized based on learned 3D representations. Volumetric based approaches partition the space into a regular 3D volume with millions of small voxels [23, 27, 54, 55, 60, 40], and the network predicts if a voxel is on the surface or not. Ray tracing can be incorporated into this voxelized structure [49, 38, 50]. The main drawback of these methods is computation and memory inefficiency, given that most voxels are not on the surface. Researchers have also tried to reconstruct point clouds [22, 13, 35, 52, 36, 2], however the high dimensionality of a point cloud often results in noisy outliers since a point cloud does not efficiently encode connectivity between points. Some recent works utilize single or multiple images to reconstruct a point cloud given strong shape priors [11, 22, 36], which cannot be directly extended to large-scale scene reconstruction. Recent work also tried to directly reconstruct surface meshes [34, 25, 53, 19, 46, 28], deformable shapes [24, 25], and some learned implicit distance functions [7, 41, 37, 6]. These reconstructed surfaces often look smoother than point-cloud-based approaches, but often lack high-frequency details. A depth map represents dense 3D information that is perfectly aligned with a reference view; depth reconstruction has been demonstrated in many previous works on reconstruction with both single view [9, 51, 16, 17, 62] and multiple views [4, 48, 18, 14, 43, 57, 43]. Some of them leverage normal information as well [14, 15]. In this paper, we present ATV, a novel spatial representation for depth estimation; we use two ATVs to progressively partition local space, which is the key to achieve coarse-to-fine reconstruction.

Deep Multi-View Stereo (MVS). The traditional MVS pipeline mainly relies on photo-consistency constraints to infer the underlying 3D geometry, but usually performs poorly on texture-less or occluded areas, or under complex lighting environments. To overcome such limitations, many deep learning-based MVS methods have emerged in the last two years, including regression-based approaches [58, 21], classification-based approaches [20] and approaches based on recurrent- or iterative- style architectures [59, 61, 5] and many other approaches [30, 38, 3, 45]. Most of these methods build a single cost volume with uniformly sampled depth hypotheses by projecting 2D image features into 3D space, and then use a stack of either 2D or 3D CNNs to infer the final depth [58, 12, 56]. However, a single cost volume often requires a large number of depth planes to achieve enough reconstruction accuracy, and it is difficult to reconstruct high-resolution depth, limited by the memory bottle-
3. Method

Some recent works aim to improve learning-based MVS methods. Recurrent networks [59] have been utilized to achieve fine depth-wise partitioning for high accuracy; a PointNet-based method [5] is also presented to densify the reconstruction for high completeness. Our goal is to reconstruct high-quality 3D geometry with both high accuracy and high completeness. To this end, we propose a novel uncertainty-aware cascaded network (UCS-Net) to reconstruct highly accurate per-view depth with high resolution.

Given a reference image $I_1$ and $N-1$ source images $\{I_i\}_{i=2}^N$, our UCS-Net progressively regresses a fine-grained depth map at the same resolution as the reference image. We show the architecture of the UCS-Net in Fig. 2. Our UCS-Net first leverages a 2D CNN to extract multi-scale deep image features at three resolutions (Sec. 3.1). Our depth prediction is achieved through three stages, which leverage multi-scale image features to predict multi-resolution depth maps. In these stages, we construct multi-scale cost volumes (Sec. 3.2), where each volume is a plane sweep volume or an adaptive thin volume (ATV). We then apply 3D CNNs to process the cost volumes to predict per-pixel depth probability distributions, and a depth map is reconstructed from the expectations of the distributions (Sec. 3.3). To achieve efficient spatial partitioning, we utilize the uncertainty of the depth prediction to construct ATVs as cost volumes for the last two stages (Sec. 3.4). Our multi-stage network effectively reconstructs depth in a coarse-to-fine fashion (Sec. 3.5).

3.1. Multi-scale feature extractor

Previous methods use downsampling layers [58, 59] or a UNet [56] to extract deep features and build a plane sweep volume at a single resolution. To reconstruct high-resolution depth, we introduce a multi-scale feature extractor, which enables constructing multiple cost volumes at different scales for multi-resolution depth prediction. As schematically shown in Fig. 2, our feature extractor is a small 2D UNet [42], which has an encoder and a decoder with skip connections. The encoder consists of a set of convolutional layers followed by BN (batch normalization) and ReLu activation layers; we use stride = 2 convolutions to
downsample the original image size twice. The decoder upsamples the feature maps, convolves the upsampled features and the concatenated features from skip links, and also applies BN and Relu layers. Given each input image $I$, the feature extractor provides three scale feature maps, $F_{i,1}$, $F_{i,2}$, $F_{i,3}$, from the decoder for the following cost volume construction. We represent the original image size as $W \times H$, where $W$ and $H$ denote the image width and height; correspondingly, $F_{i,1}$, $F_{i,2}$ and $F_{i,3}$ have resolutions of $W \times \frac{W}{2}$, $W \times \frac{W}{4}$ and $W \times H$, and their numbers of channels are 32, 16 and 8 respectively. Our multi-scale feature extractor allows for the high-resolution features to properly incorporate the information at lower resolutions through the learned upsampling process; thus in the multi-stage depth prediction, each stage is aware of the meaningful feature knowledge used in previous stages, which leads to reasonable high-frequency feature extraction.

3.2. Cost volume construction

We construct multiple cost volumes at multiple scales by warping the extracted feature maps, $F_{i,1}$, $F_{i,2}$, $F_{i,3}$ from source views to a reference view. Similar to previous work, this process is achieved through differentiable unprojection and projection. In particular, given camera intrinsic and extrinsic matrices $\{K_i, T_i\}$ for each view $i$, the $4 \times 4$ warping matrix at depth $d$ at the reference view is given by:

$$H_i(d) = K_i T_i T_i^{-1} K_i^{-1}. \quad (1)$$

In particular, when warping to a pixel in the reference image $I_i$ at location $(x, y)$ and depth $d$, $H_i(d)$ multiplies the homogeneous vector $(xd, yd, d, 1)$ to finds its corresponding pixel location in each $I_i$ in homogeneous coordinates.

Each cost volume consists of multiple planes; we use $L_{k,j}$ to denote the depth hypothesis of the $j$th plane at the $k$th stage, and $L_{k,j}(x)$ represents its value at pixel $x$. At stage $k$, once we warp per-view feature maps $F_{i,k}$ at all depth planes with corresponding hypotheses $L_{k,j}$, we calculate the variance of the warped feature maps across views at each plane to construct a cost volume. We use $D_k$ to represent the number of planes for stage $k$. For the first stage, we build a standard plane sweep volume, whose depth hypotheses are of constant values, i.e. $L_{1,j}(x) = d_j$. We uniformly sample $\{d_j\}_{j=1}^{D_1}$ from a pre-defined depth interval $[d_{\text{min}}, d_{\text{max}}]$ to construct the volume, in which each plane is constructed using $H_i(d_j)$ to warp multi-view images. For the second and third stages, we build novel adaptive thin volumes, whose depth hypotheses have spatially-varying depth values according to pixel-wise uncertainty estimates of the previous depth prediction. In this case, we calculate per-pixel per-plane warping matrices by setting $d = L_{k,j}(x)$ in Eqn. 1 to warp images and construct cost volumes. Please refer to Sec. 3.4 for uncertainty estimation.

3.3. Depth prediction and probability distribution

At each stage, we apply a 3D CNN to process the cost volume, infer multi-view correspondence and predict depth probability distributions. In particular, we use a 3D UNet similar to [58], which has multiple downsampling and upsampling 3D convolutional layers to reason about scene geometry at multiple scales. We apply depth-wise softmax at the end of the 3D CNNs to predict per-pixel depth probabilities. Our three stages use the same network architecture without sharing weights, so that each stage learns to process its information at a different scale. Please refer to the supplemental material for details of our 3D CNN architecture.

The 3D CNN at each stage predicts a depth probability volume that consists of $D_k$ depth probability maps $P_{k,j}$ associated with the depth hypotheses $L_{k,j}$. $P_{k,j}$ expresses per-pixel depth probability distributions, where $P_{k,j}(x)$ represents how probable the depth at pixel $x$ is $L_{k,j}(x)$. A depth map $\hat{L}_k$ at stage $k$ is reconstructed by weighted sum:

$$\hat{L}_k(x) = \sum_{j=1}^{D_k} L_{k,j}(x) \cdot P_{k,j}(x). \quad (2)$$

3.4. Uncertainty estimation and ATV

The key for our framework is to progressively sub-partition the local space and refine the depth prediction with increasing resolution and accuracy. To do so, we construct novel ATVs for the last two stages, which have curved sweeping planes with spatially-varying depth hypotheses (as illustrated in Fig. 1 and Fig. 2), based on uncertainty inference of the predicted depth in its previous stage.

Given a set of depth probability maps, previous work only utilizes the expectation of the per-pixel distributions (using Eqn. (2)) to determine an estimated depth map. For the first time, we leverage the variance of the distribution for uncertainty estimation, and construct ATVs using the uncertainty. In particular, the variance $V_k(x)$ of the probability distribution at pixel $x$ and stage $k$ is calculated as:

$$V_k(x) = \sum_{j=1}^{D_k} P_{k,j}(x) \cdot (L_{k,j}(x) - \hat{L}_k(x))^2, \quad (3)$$

and the corresponding standard deviation is $\hat{\sigma}_k(x) = \sqrt{V_k}$. Given the depth prediction $L_k(x)$ and its variance $\hat{\sigma}_k(x)^2$ at pixel $x$, we propose to use a variance-based confidence interval to measure the uncertainty of the prediction:

$$C_k(x) = [L_k(x) - \lambda \hat{\sigma}_k(x), L_k(x) + \lambda \hat{\sigma}_k(x)], \quad (4)$$

where $\lambda$ is a scalar parameter that determines how large the confidence interval is. For each pixel $x$, we uniformly sample $D_{k+1}$ depth values from $C_k(x)$ of the $k$th stage, to get its depth values $L_{k+1,1}(x)$, $L_{k+1,2}(x)$, ... $L_{k+1,D_{k+1}}(x)$
of the depth planes for stage \((k + 1)\). In this way, we construct \(D_{k+1}\) spatially-varying depth hypotheses \(L_{k+1,j}\), which form the ATV for stage \((k + 1)\).

The estimated \(C_k(x)\) expresses the uncertainty interval of the prediction \(L_k(x)\), which determines the physical thickness of an ATV at each pixel. In Fig. 3, we show two actual examples with two pixels and their estimated uncertainty intervals \(C_k(x)\) around the predictions (red dash line). The \(C_k\) essentially depicts a probabilistic local space around the ground truth surface, and the ground truth depth is located in the uncertainty interval with a very high confidence. Note that, our variance-based uncertainty estimation is differentiable, which enables our UCS-Net to learn to adjust the probability prediction at each stage to achieve optimized intervals and corresponding ATVs for following stages in an end-to-end training process. As a result, the spatially varying depth hypotheses in ATVs naturally adapt to the uncertainty of depth predictions, which leads to highly efficient spatial partitioning.

3.5. Coarse-to-fine prediction

Our UCS-Net leverages three stages to reconstruct depth at multiple scales from coarse to fine, which generally supports different numbers \((D_k)\) of planes in each stage. In practice, we use \(D_1 = 64\), \(D_2 = 32\) and \(D_3 = 8\) to construct a plane sweep volume and two ATVs with sizes of \(\frac{W}{2} \times \frac{H}{2} \times 64\), \(\frac{W}{4} \times \frac{H}{4} \times 32\) and \(H \times W \times 8\) to estimate depth at corresponding resolutions. While our two ATVs have small numbers \((32 and 8)\) of depth planes, they in fact partition local depth ranges at finer scales than the first stage volume; this is achieved by our novel uncertainty-aware volume construction process which adaptively controls local depth intervals. This efficient usage of a small number of depth planes enables the last two stages to deal with higher pixel-wise resolutions given the limited memory, which makes fine-grained depth reconstruction possible. Our novel ATV effectively expresses the locality and uncertainty in the depth prediction, which enables state-of-the-art depth reconstruction results with high accuracy and high completeness through a coarse-to-fine framework.

3.6. Training details

Training set. We train our network on the DTU dataset [1]. We split the dataset into training, validate and testing set, and create ground truth depth similar to [58]. In particular, we apply Poisson reconstruction [29] on the point clouds in DTU, and render the surface at the captured views with three resolutions, \(\frac{W}{4} \times \frac{H}{4}, \frac{W}{2} \times \frac{H}{2}\) and the original \(W \times H\). In particular, we use \(W \times H = 640 \times 512\) for training.

Loss function. Our UCS-Net predicts depth at three resolutions; we apply \(L1\) loss on depth prediction at each resolution with the rendered ground truth at the same resolution. Our final loss is the combination of the three \(L1\) losses.

Training policy. We train our full three-stage network from end to end for 60 epochs. We use Adam optimizer with an initial learning rate of 0.0016. We use 8 NVIDIA GTX 1080Ti GPUs to train the network with a batch size of 16 (mini-batch size of 2 per GPU).

4. Experiments

We now evaluate our UCS-Net. We do benchmarking on the DTU and Tanks and Temple datasets. We then justify the effectiveness of the designs of our network, in terms of uncertainty estimation and multi-stage prediction.

Evaluation on the DTU dataset [1]. We evaluate our method on the DTU testing set. To reconstruct the final point cloud, we follow [14] to fuse the depth from multiple views; we use this fusion method for all our exper-

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
<th>Comp.</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camp [4]</td>
<td>0.835</td>
<td>0.554</td>
<td>0.695</td>
</tr>
<tr>
<td>Furu [13]</td>
<td>0.613</td>
<td>0.941</td>
<td>0.777</td>
</tr>
<tr>
<td>Tola [48]</td>
<td>0.342</td>
<td>1.190</td>
<td>0.766</td>
</tr>
<tr>
<td>Gipuma [14]</td>
<td>0.283</td>
<td>0.873</td>
<td>0.578</td>
</tr>
<tr>
<td>SurfaceNet [46]</td>
<td>0.450</td>
<td>1.040</td>
<td>0.745</td>
</tr>
<tr>
<td>MVSNet [58]</td>
<td>0.396</td>
<td>0.527</td>
<td>0.462</td>
</tr>
<tr>
<td>R-MVSNet [59]</td>
<td>0.383</td>
<td>0.452</td>
<td>0.417</td>
</tr>
<tr>
<td>Point-MVSNet [5]</td>
<td>0.342</td>
<td>0.411</td>
<td>0.376</td>
</tr>
<tr>
<td>Our 1st stage</td>
<td>0.548</td>
<td>0.529</td>
<td>0.539</td>
</tr>
<tr>
<td>Our 2nd stage</td>
<td>0.401</td>
<td>0.397</td>
<td>0.399</td>
</tr>
<tr>
<td>Our full model</td>
<td>0.338</td>
<td>0.349</td>
<td>0.344</td>
</tr>
</tbody>
</table>
Table 2: Quantitative results of F-scores (higher means better) on Tanks and Temples.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Family</th>
<th>Francis</th>
<th>Horse</th>
<th>Lighthouse M60</th>
<th>Panther</th>
<th>Playground</th>
<th>Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSNet[58]</td>
<td>43.48</td>
<td>55.99</td>
<td>28.55</td>
<td>25.07</td>
<td>50.79</td>
<td>53.96</td>
<td>50.86</td>
<td>47.90</td>
</tr>
<tr>
<td>R-MVSNet[59]</td>
<td>48.40</td>
<td>69.96</td>
<td>46.65</td>
<td>32.59</td>
<td>42.95</td>
<td>51.88</td>
<td>48.80</td>
<td>52.00</td>
</tr>
<tr>
<td>Dense-R-MVSNet[59]</td>
<td>50.55</td>
<td>73.01</td>
<td>54.46</td>
<td>43.42</td>
<td>43.88</td>
<td>46.80</td>
<td>46.69</td>
<td>50.87</td>
</tr>
<tr>
<td>Point-MVSNet[5]</td>
<td>48.27</td>
<td>61.79</td>
<td>41.15</td>
<td>34.20</td>
<td>50.79</td>
<td>51.97</td>
<td>50.85</td>
<td>52.38</td>
</tr>
<tr>
<td>Our full model</td>
<td>54.83</td>
<td>76.09</td>
<td>53.16</td>
<td>43.03</td>
<td>54.00</td>
<td>55.60</td>
<td>51.49</td>
<td>57.38</td>
</tr>
</tbody>
</table>

Figure 4: Comparisons with R-MVSNet on an example in the DTU dataset. We show rendered images of the point clouds of our method, R-MVSNet and the ground truth. In this example, the ground truth from scanning is incomplete. We also show insets for detailed comparisons marked as a blue box in the ground truth. Note that our result is smoother and has fewer outliers than R-MVSNet’s result.

Evaluation on Tanks and Temple dataset [31]. We now evaluate the generalization of our model by testing our network trained with the DTU dataset on complex outdoor scenes in the Tanks and Temple intermediate dataset. We use $N = 5$ and $W \times H = 1920 \times 1056$ for this experiment. Our method outperforms most published methods, and to the best of our knowledge, when comparing with all published learning-based methods, we achieve the best average F-score (54.83) as shown in Tab. 2. In particular, our method obtains higher F-scores than MVSNet[58] and Point-MVSNet[5] in all nine testing scenes. Dense-R-MVSNet leverages a well-designed post-processing method and achieves slightly better performance than ours on two of the scenes, whereas our work is focused on high-quality per-view depth reconstruction, and we use a traditional fu-
variance-based uncertainty estimation is equivalent to approximating a depth probability distribution as a Gaussian distribution and then computing its confidence interval with a specified scale on its standard deviation as in Eqn. 4.

We note that our variance-based uncertainty estimation is not only valid for single-mode Gaussian-like distributions as in Fig. 3.a, but also valid for many multi-mode cases as in Fig. 3.b, which shows a challenging example near object boundary. In Fig. 3.b, the predicted first-stage depth distribution has multiple modes; yet, it correspondingly has large variance and a large enough uncertainty interval. Our network predicts reasonable uncertainty intervals that are able to cover the ground truth depth in most cases, which allows for increasingly accurate reconstruction in the following stages at finer local spatial scales. This is made possible by the differentiable uncertainty estimation and the end-to-end training process, from which the network learns to control per-stage probability estimation to obtain proper uncertainty intervals for ATV construction. Because of this, we observe that our network is not very sensitive to different λ, and learns to predict similar uncertainty. Our uncertainty-aware volume construction process enables highly efficient spatial partitioning, which further allows for the final reconstruction to be of high accuracy and high completeness.

Evaluation of multi-stage depth prediction. We have quantitatively demonstrated that our multi-stage framework reconstructs scene geometry with increasing accuracy and completeness in every stage (see Fig. 1). We now further evaluate our network and do ablation studies about different stages on the DTU testing set with detailed quantitative and qualitative comparisons. We compare with naive upsampling to justify the effectiveness of our uncertainty-aware coarse-to-fine framework. In particular, we compare the results from our full model and the results from the first two stages with naive bilinear upsampling using a scale of 2 (for both height and width) in Tab. 4. We can see that upsampling does improve the reconstruction, which benefits from denser geometry and using our high-quality low-resolution results as input. However, the improvement made by naive upsampling is very limited, which is much lower than our improvement from our ATV-based upsampling. Our UCS-Net makes use of the ATV – a learned local spatial repre-

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Interval</th>
<th>$D_b$</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSV</td>
<td>100%</td>
<td>508.8mm</td>
<td>64</td>
</tr>
<tr>
<td>1st ATV</td>
<td>94.72%</td>
<td>13.88mm</td>
<td>32</td>
</tr>
<tr>
<td>2st ATV</td>
<td>85.22%</td>
<td>3.83mm</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 3: Evaluation of uncertainty estimation. The PSV is the first-stage plane sweep volume; the 1st ATV is constructed after the first stage and used in the second stage; the 2nd ATV is used in the third stage. We show the percentages of uncertainty intervals that cover the ground truth depth. We also show the average length of the intervals, the number of depth planes and the unit sampling distance.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Scale</th>
<th>Size</th>
<th>Acc.</th>
<th>Comp.</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>×1</td>
<td>400x296</td>
<td>0.548</td>
<td>0.529</td>
<td>0.539</td>
</tr>
<tr>
<td>1</td>
<td>×2</td>
<td>800x592</td>
<td>0.411</td>
<td>0.535</td>
<td>0.473</td>
</tr>
<tr>
<td>2</td>
<td>×1</td>
<td>800x592</td>
<td>0.401</td>
<td>0.397</td>
<td>0.399</td>
</tr>
<tr>
<td>2</td>
<td>×2</td>
<td>1600x1184</td>
<td>0.342</td>
<td>0.386</td>
<td>0.364</td>
</tr>
<tr>
<td>3</td>
<td>×1</td>
<td>1600x1184</td>
<td>0.338</td>
<td>0.349</td>
<td>0.344</td>
</tr>
</tbody>
</table>

Table 4: Ablation study on the DTU testing set with different stages and upsampling scales (a scale of 1 represents the original result at the stage). The quantitative results represent average distances in mm (lower is better).
Our first stage  Our second stage  Our full model  Ground truth

Figure 5: Qualitative comparisons between multi-stage point clouds and the ground truth point cloud on a scene in the DTU validate set. We show zoom-out (top) and zoom-in (bottom) rendered point clouds; the corresponding zoom-in region is marked in the ground truth as a green box. Our UCS-Net achieves increasingly dense and accurate reconstruction through the multiple stages. Note that, the ground truth point cloud is obtained by scanning, which is even of lower quality than our reconstructions in this example.

<table>
<thead>
<tr>
<th>Method</th>
<th>Running time (s)</th>
<th>Memory (MB)</th>
<th>Input size</th>
<th>Prediction size</th>
</tr>
</thead>
<tbody>
<tr>
<td>One stage</td>
<td>0.065</td>
<td>1309</td>
<td>160x120</td>
<td></td>
</tr>
<tr>
<td>Two stages</td>
<td>0.114</td>
<td>1607</td>
<td>640x480</td>
<td>320x240</td>
</tr>
<tr>
<td>Our full model</td>
<td>0.257</td>
<td>1647</td>
<td>640x480</td>
<td></td>
</tr>
<tr>
<td>MVSNet [58]</td>
<td>1.049</td>
<td>4511</td>
<td>640x480</td>
<td>160x120</td>
</tr>
<tr>
<td>R-MVSNet [59]</td>
<td>1.421</td>
<td>4261</td>
<td>640x480</td>
<td>160x120</td>
</tr>
</tbody>
</table>

Table 5: Performance comparisons. We show the running time and memory of our method by running the first stage, the first two stages and our full model.

Figure 5 shows qualitative comparisons between our reconstructed point clouds and the ground truth point cloud. Our UCS-Net is able to effectively refine and densify the reconstruction through multiple stages. Note that, our MVS-based reconstruction is even more complete than the ground truth point cloud that is obtained by scanning, which shows the high quality of our reconstruction.

Compared runtime performance. We now evaluate the timing and memory usage of our method. We run our model on the DTU validate set with an input image resolution of $W \times H = 640 \times 480$; We compare performance with MVSNet and R-MVSNet with 256 depth planes using the same inputs. Table 5 shows the performance comparisons including running time and memory. Note that, our full model is the only one that reconstructs the depth at the original image resolution that is much higher than the comparison methods. However, this hasn’t introduced any higher computation or memory consumption. In fact, our method requires significantly less memory and shorter running time, which are only about a quarter of the memory and time used in other methods. This demonstrates the benefits of our coarse-to-fine framework with fewer depth planes (104 in total), in terms of system resource usage. Our UCS-Net with ATVs achieves high-quality reconstruction with high computation and memory efficiency.

5. Conclusion

In this paper, we present a novel deep learning-based approach for multi-view stereo. We propose the novel uncertainty-aware cascaded stereo network (UCS-Net), which utilizes the adaptive thin volume (ATV), a novel spatial representation. For the first time, we make use of the uncertainty of the prediction in a learning-based MVS system. Specifically, we leverage variance-based uncertainty intervals at one cascade stage to construct an ATV for its following stage. The ATVs are able to progressively subpartition the local space at a finer scale, and ensure that the smaller volume still surrounds the actual surface with a high probability. Our novel UCS-Net achieves highly accurate and highly complete scene reconstruction in a coarse-to-fine fashion. We compare our method with various state-of-the-art benchmarks; we demonstrate that our method is able to achieve the qualitatively and quantitatively best performance with high computation- and memory- efficiency. Our novel UCS-Net takes a step towards making the learning-based MVS method more reliable and efficient.

Acknowledgements This work was funded in part by Kuaishou Technology, NSF grant IIS-1764078, NSF grant 1703957, the Ronald L. Graham chair and the UC San Diego Center for Visual Computing.
References


Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh


