Self-supervised Learning of Depth Inference for Multi-view Stereo

Jiayu Yang\textsuperscript{1}, Jose M. Alvarez\textsuperscript{2}, Miaomiao Liu\textsuperscript{1}
\textsuperscript{1}Australian National University, \textsuperscript{2}NVIDIA
\{jiayu.yang, miaomiao.liu\}@anu.edu.au, josea@nvidia.com

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figures/fig1.png}
\caption{Point cloud reconstructed by existing unsupervised MVS network [13] and our methods. Best view on screen.}
\end{figure}

Abstract

Recent supervised multi-view depth estimation networks have achieved promising results. Similar to all supervised approaches, these networks require ground-truth data during training. However, collecting a large amount of multi-view depth data is very challenging. Here, we propose a self-supervised learning framework for multi-view stereo that exploit pseudo labels from the input data. We start by learning to estimate depth maps as initial pseudo labels under an unsupervised learning framework relying on image reconstruction loss as supervision. We then refine the initial pseudo labels using a carefully designed pipeline leveraging depth information inferred from higher resolution images and neighboring views. We use these high-quality pseudo labels as the supervision signal to train the network and improve, iteratively, its performance by self-training. Extensive experiments on the DTU dataset show that our proposed self-supervised learning framework outperforms existing unsupervised multi-view stereo networks by a large margin and performs on par compared to the supervised counterpart. Code is available at https://github.com/JiayuYANG/Self-supervised-CVP-MVSNet.

1. Introduction

The goal of Multi-view Stereo (MVS) is to reconstruct the 3D model of a scene from a set of images captured at multiple viewpoints. While this problem has been studied for decades [19], the current best performance is achieved by cost-volume based supervised deep neural networks for MVS [26, 27, 25, 9, 5, 24]. The success of these networks mainly relies on large amount of ground truth depth as training data, which is generally captured by expensive and multiple synchronized image and depth sensors.

The use of synthetic data is considered a good alternative to handle the main challenges in collecting training data for MVS [28]. Given a set of 3D scene models with a proper setting of lighting conditions, we can obtain a large number of synthetic multiple view images with ground truth depths [28]. While it is possible to train the network using this synthetic data, for successfully deploying the model in real scenes, we still require to fine-tune the model using data from the target domain [16]. Another alternative is adopting an unsupervised learning strategy [6, 13]. In this case, the few existing unsupervised MVS approaches use an image reconstruction loss to supervise the training process. This training strategy heavily relies on image colors’ photometric consistency for multiple views images, which is sensitive to illumination changes. While both alternatives remove the dependency on depth labels, their performance is far inferior compared to their corresponding supervised counterparts on the target domain.

In this paper, we propose a self-supervised learning framework for depth inference from multi-view images. Our goal is to generate high-quality depth maps as pseudo labels for training the network only from multiple view images. To this end, we first rely on an image recon-
construction loss to supervise the training of a cost-volume based depth inference network. We then use this unsupervised network to infer depth maps as pseudo labels for self-supervision [15]. While our unsupervised network can estimate accurate depth for pixels with rich textures and satisfying color consistency across views, these pseudo depth labels still contain a large amount of noise.

To refine the pseudo labels, we first propose to infer depth from a higher resolution image than the required training image to obtain depth estimates of higher accuracy for trustworthy pixels. Then, we filter depth with large errors by leveraging depth information from neighboring views and, finally, use multi-view depth fusion, mesh generation, and depth rendering to fill in the incomplete pseudo depth labels. With our carefully designed pipeline, we improve the pseudo labels’ quality; and use them for training the network, improving its performance within a few iterations.

Our contributions can be summarized as follows:

• We propose a self-supervised learning framework for multi-view depth estimation.

• We generate an initial set of pseudo depth labels from an unsupervised learning network and then improve their quality with a carefully designed pipeline to use them to supervise the network yielding performance improvements.

Our extensive set of experiments demonstrate that the proposed self-supervised framework outperforms existing unsupervised MVS networks by a large margin and performs on par compared to the supervised counterpart.

2. Related Works

Supervised Multi-view Stereo. Recent supervised learning-based multi-view depth estimation networks have shown great potential to replace traditional optimization-based MVS pipelines [7, 21, 2, 18]. In particular, cost-volume-based networks have achieved impressive results for depth inference from multi-view images. For instance, Yao et al. in [26] propose MVSNet to learn the depth map for each view by constructing a cost volume followed by 3D CNN regularization. While effective in inferring depth for low-resolution images, their framework cannot scale to handle high-resolution images. Follow-up works have focused on reducing the memory requirements of cost volume-based methods. Yao et al. use a recurrent network to regularize the cost volume in a sequential manner [27], and a few other approaches integrate a coarse-to-fine strategy to construct partial cost volumes [9, 25, 5] resulting not only in memory reductions but also achieving higher resolution estimation. All these works, however, focus on designing effective backbones.

Another line of research [30, 29, 3, 24] explore multi-view aggregation to further leverage information from multiple view images and improve the performance of the network. Unlike these works mainly focusing on the backbone design or improving the view-aggregation strategy, we focus on self-supervised learning for depth inference. We adopt the backbone network in CVP-MVSNet [25], which is compact and flexible in handling high-resolution images, and leave as future work the introduction of view-aggregation into our framework.

Synthetic Datasets for Multi-view Stereo. Existing supervised methods rely on ground-truth depth maps for supervision. However, collecting a large amount of high-quality multi-view ground-truth depth data is very challenging. One solution is to use synthetic data for training. For instance, Yao et al. created BlendedMVS [28], a synthetic dataset based on the rendered depth maps and blended images of meshes generated by existing MVS algorithms. This synthetic data is potentially enough for training MVS algorithms; however, algorithms trained on synthetic data inherently suffer from domain differences with real data. To bridge this domain gap, Mallick et al. [16] introduce a self-supervised domain adaptation method for multi-view stereo. This approach improves the model’s performance over the model without using domain adaptation; however, its performance on the target domain is still far inferior to its supervised counterpart.

Unsupervised Multi-view Stereo Networks. Unsupervised learning-based methods have emerged as an alternative to reduce the requirement of ground-truth data [6, 13, 10]. For instance, Dai et al. propose the first unsupervised MVS network with a symmetric unsupervised network that enforces cross-view consistency of multi-view depth maps during both training and testing [6]. They use a view synthesis loss and a cross-view consistency loss to minimize the discrepancy between the source image and the reconstructed image and encourage cross-view consistency. Concurrently, Khot et al. propose to utilize photometric consistency for unsupervised training [13]. They also adopt a similar loss function including a $L_1$ loss between image intensity, a structure similarity (SSIM) loss, and a depth smoothness loss. Very recently, Huang et al. propose the M$^3$VSNet consisting of a multi-metric unsupervised network and a multi-metric loss function that provides comparable performance with the original supervised MVSNet [26].

While these unsupervised MVS methods do not require ground-truth depth training data, their training strategy relies heavily on the color consistency across multiple views, which is sensitive to environmental lighting changes. As a result, their performance is still compromised compared to their supervised counterpart [26]. By contrast, we focus on self-supervised learning and generating pseudo depth labels from input image data to supervise the network’s training.
3. Method

We aim to generate high-quality pseudo depth labels from multi-view images for self-supervised learning of depth inference. To this end, we design a framework consisting of two stages: unsupervised learning for initial pseudo label estimation and iterative pseudo label refinement for self-training. Below, we first introduce the overall network structure in Section 3.1, and then, in Sections 3.2 and 3.3, the stages of our framework. The overall framework is depicted in Fig 2 where we ignore camera parameters for simplicity.

3.1. Network Structure

We apply the recent CVP-MVSNet [25] as the backbone network in our framework. Specifically, CVP-MVSNet takes as input a reference image $I_0 \in \mathbb{R}^{h \times w}$, source images $\{I_i\}_{i=1}^N$ and the corresponding camera intrinsics and extrinsics parameters for all views $\{K_i, R_i, t_i\}_{i=1}^N$ and infers the depth map $D_0$ for $I_0$. Unlike other cost-volume-based MVS networks, CVP-MVSNet adopts a cost-volume pyramid structure with weight sharing across levels, which can be trained with low-resolution images and still handle any high-resolution image during inference. We follow the same network design as in [25] and build a cost-volume pyramid of $(L + 1)$ levels.

We formulate our self-training loss as

$$l_{\text{pseudo}} = \sum_{l=0}^{L} \sum_{p \in \Omega} \| D_{\text{pseudo}}^l(p) - D^l(p) \|_1, \quad (1)$$

where $\Omega$ is the set of valid pixels associated to pseudo depth labels, $D_{\text{pseudo}}^l$ and $D^l$ denote the pseudo depth label and depth estimate at the $l$th level of the cost volume pyramid, respectively. The quality of the pseudo depth label is crucial for achieving good performance. Next, we introduce the proposed unsupervised learning method to generate initial pseudo-labels, then the pseudo-label refinement process, and the overall self-supervised learning pipeline.

3.2. Unsupervised Learning for Pseudo Depth Label

In the first stage, we learn to estimate depth based on photometric consistency from multi-view images (see Fig. 2). We adopt the CVP-MVSNet as the backbone network and use an image reconstruction loss as supervision signal to train the network. Unlike recent unsupervised MVS method [6] that uses the estimated depth map for image synthesis, we, inspired by networks designed for view synthesis [31], directly synthesize image from the probability distribution of depth hypothesis. To leverage the cost-volume
A level L pyramid network structure in [25], we build image intensity volume pyramid, \( \{B_l^i\}_{l=0}^{L} \), based on the warped pixel intensity of each depth hypothesis at each level \( l \). See Fig. 3 for an illustration for one source view \( i \) and pyramid level \( l \).

Specifically, we adopt the differentiable homography defined in [25] at each depth hypothesis for image warping at level \( L \), and the perspective projection defined in [25] for the other levels. Given the image intensity volume \( \{B_l^i\}_{l=0}^{L} \) and depth hypothesis probability volumes \( \{P_l^i\}_{l=0}^{L} \), we can obtain the synthesized image from source view \( i \) as the expectation of warped image intensity based on all depth hypotheses,

\[
I_{l \to 0}^i (x) = \sum_d B_{i \times}^l (d) P_{l}^i (d)
\]

where \( x = (u, v) \) is a pixel in the reference view, \( B_{i \times}^l (d) \in \mathbb{R}^3 \) is the intensity of the warped image at pixel \( x \) with depth \( d \), and \( P_{l}^i (d) \in [0, 1] \) is the probability of pixel \( x \) with depth \( d \) predicted by the model.

We explore a view synthesis loss functions similar to [6] to encourage depth smoothness and enforce consistency between the synthesized image and the reference image. We also adopt the perceptual loss proposed in [11] to enforce high-level contextual similarity between the synthesized images and the reference image. Specifically, we use a weighted combination of four loss functions:

\[
l_{syn} = \alpha_1 l_g + \alpha_2 l_{ssim} + \alpha_3 l_p + \alpha_4 l_s,
\]

where \( l_g \) is the image gradient loss, \( l_{ssim} \) is the structure similarity loss, \( l_p \) is the perceptual loss, \( l_s \) is the depth smoothness loss, and \( \alpha_i \) sets the influence of each loss – see supplemental material for details of each loss function.

### 3.3. Iterative Self-training

Given the network initially trained in an unsupervised manner, we create an initial set of pseudo depth labels by inferring depth maps for the images in the training set (see Fig. 2). Specifically, the network takes a reference image \( I_0 \), and the source images \( \{I_i \mid I_i \in \mathbb{R}^{h \times w \times 3}, i = 1, \ldots, N \} \), as input to learn from the cost volume pyramid and estimate the depth map \( D_0 \in \mathbb{R}^{h \times w} \). We obtain the initial pseudo depth labels for the training set as \( \{D_m \mid D_m \in \mathbb{R}^{h \times w}, m = 1, \ldots, M \} \).

As unsupervised learning relies on the image reconstruction loss, which is sensitive to illumination changes, the initial pseudo depth label is subject to a certain noise level. Next, we describe three stages, refinement from a high-resolution image, pseudo depth filtering by consistency check, and multiple view fusion to improve the quality of the initial set of pseudo depth labels.

#### Pseudo Label Refinement from High Resolution Image

Recall that CVP-MVSNet is a coarse-to-fine depth estimation network with parameter sharing across pyramid levels. Thus, we can evaluate a model trained on low-resolution images and depth pairs on higher resolution ones.

To improve the quality of pseudo labels, we propose to refine the initial pseudo depth label by using information from a higher resolution training image \( \{I_m^{\prime} \mid m = 1, \ldots, M \} \in \mathbb{R}^{h \times w \times 3} \). As evidenced in CVP-MVSNet [25], a higher resolution image carries more discriminative features. Therefore, we can build a cost volume with a smaller depth search interval to further refine the depth map. As shown in Fig. 2, we extend the coarse network (2 levels) to a fine network (5 levels) to further refine the pseudo label and use the refined pseudo label to supervise the original coarse network itself as self-training. As we will show, this process improves the accuracy of depth estimate for pixels with rich features and satisfying photometric consistency across views.

However, depth estimates for pixels sensitive to illumination changes or in textureless regions still have large errors. In the following, we detail our approach to filter noise and improve performance.

#### Pseudo Label Filtering

Assume the refined pseudo depth label obtained from high-resolution images is \( \{D_m^{\prime} \mid D_m^{\prime} \in \mathbb{R}^{h \times w}, m = 1, \ldots, M \} \). To select reliable depth labels for self-training, we apply a cross-view depth consistency check utilizing depth re-projection error to measure the pseudo depth labels’ consistency. To refine the pseudo depth label for each view \( D_m^{\prime} \), we form pairs of views between the reference view \( i \) and any other view for the same scene to calculate depth re-projection errors.

Here, we provide the calculation of the depth re-projection error between a reference view \( D_i^{\prime} \) and a source view \( D_j^{\prime} \). Fig. 4 shows the visualization of the depth re-projection error. Assume the camera calibration matrices for view \( i \) and \( j \) are \( \mathbf{K}_i \) and \( \mathbf{K}_j \), the relative rotation matrix and the translation vector between this pair of views are defined as \( \mathbf{R}_{ij} \) and \( \mathbf{T}_{ij} \), respectively. For each pixel \( x = (u, v)^T \) in the \( i \)-th view, its corresponding 3D point defined in the camera coordinate system for that view is defined as \( \mathbf{X} = D_i^{\prime} (x) \mathbf{K}_i^{-1} \bar{x} \), where \( \bar{x} \) is the homogeneous coordinate of \( x \). Its projection to the \( j \)-th view is defined as \( \lambda_x \bar{x}_j = \mathbf{K}_j (\mathbf{R}_{ij} \mathbf{X} + \mathbf{T}_{ij}) \), where \( \bar{x}_j \) is the homoge-
neous coordinate of $x_j$. As $x_j$ might not be integer, we obtain the depth for pixel $x_j$, namely $D_j'(x_j)$, by bilinear interpolation. We then obtain the 3D point in the $j$-th view based on $D_j'(x_j)$ as $X_j = D_j'(x_j)K_j^{-1}x_j$. Therefore, by re-projecting this point back to view $i$, we can obtain $X_{ji} = R_{ij}^{-1}(X_j - T_{ij})$. Its depth in the $i$-th view is defined as $d_{ji}(x) = X_{ji}$. Finally, the depth re-projection error for pixel $x$ computed from the $j$-th view is defined as $r_{ji}(x) = |D_j'(x) - d_{ji}(x)|$.

Assume there are $(M - 1)$ source views for the current reference view $i$. We compute the set of depth re-projection errors $\{r_{ji}(x)\}_{j=1}^{M-1}$ from $M - 1$ source views and then define a criterion to filter noisy ones and obtain a refined pseudo depth map $\{D_j''(x)\}_{j=1}^{M}$ with sparse but accurate depth values. More precisely, $D_j''(x) = D_j'(x)$ iff $\sum_{j=1}^{M-1} q_{ji}(x) > n_{\text{min}}$, where $n_{\text{min}}$ is the minimum number of depth consistency, and $q_{ji}(x)$ is the filtering criterion defined as:

$$ q_{ji}(x) = \begin{cases} 1, & \text{if } r_{ji}(x) \leq r_{\text{max}} \\ 0, & \text{otherwise.} \end{cases} $$

**Multi-view Pseudo Label Fusion.** We now focus on completing the sparse pseudo depth map resulting from the filtering process. Each view provides a different set of sparse points; therefore, we can combine them to generate a more complete point cloud. To this end, we first project points defined by depth maps from multiple views into the world coordinate system to form a point cloud (see Fig. 2). Specifically, given $M$ filtered high-quality pseudo depth map $\{D_m''\}_{m=1}^{M} \in \mathbb{R}^{H \times W}$ of the same scene, we project them into 3D space using their corresponding camera parameters $\{K_m, R_m, t_m\}_{m=1}^{M}$ to form a pseudo point cloud $\mathcal{X} \in \mathbb{R}^{\Phi \times 3}$, where $\Phi$ is the number of pseudo-3D points corresponding to the aggregation of valid pseudo labels from all views.

Formally, we define the pseudo point cloud from fusing multiple views as

$$ \mathcal{X} = \{P_m^x | m \in \{1, 2, \cdots, M\}, x \in \Omega_m\}, \quad (4) $$

where $\Omega_m$ is the set of pixel coordinates with high quality depth values per image, and $P_m^x = R_m^{-1}(D_m''(x)K_m^{-1}x - t_m)$ is the pseudo-3D point for each pixel $x$ in the $m$-th view. We then use the Screened Poisson Surface Reconstruction method [12] to filter out noisy pseudo labels, improve the completeness of $\mathcal{X}$, and generate a mesh $S = \text{SPSR}(\mathcal{X})$.

Finally, we render this mesh $S$ into image coordinate of each view as a complete pseudo depth map $\{D_m''\}_{m=1}^{M} \in \mathbb{R}^{H \times W}$ and use them as supervision signal for each view.

**Overall self-supervised learning pipeline.** Our self-supervised learning framework can be summarised in Algorithm 1 where we ignore camera parameters for simplicity. Note that we trained the model on low resolution images and depth map pairs. In particular, we render the 3D model to small resolution depth map after each iteration. Such process guarantees the supervision signal for low resolution depth training is of high quality and the performance will not deteriorate rapidly after iterative self-training.

### 4. Experiments

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

#### 4.1. Dataset

**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. We only use the

---

**Algorithm 1** Self-supervised Learning Framework

**Input:** $\{I_m\}_{m=1}^{M} \in \mathbb{R}^{H \times W \times 3}$, $\{T_m\}_{m=1}^{M} \in \mathbb{R}^{H \times W \times 3}$

**Output:** Trained model parameters $\hat{\theta}_T$.

**Unsupervised Initialization:**

1. Train $\hat{\theta}_0$ using $\{I_m\}_{m=1}^{M}$ and $l_{\text{syn}}$.

**Iterative Self-training:**

2. for $t = 1$ to $T$

3. Inference $\{D_m'''\}_{m=1}^{M}$ from $\{I_m\}_{m=1}^{M}$ using $\hat{\theta}_{t-1}$

4. Refine $\{D_m'''\}_{m=1}^{M}$ to $\{D_m''\}_{m=1}^{M}$ using $\{I_m\}_{m=1}^{M}$ and $\hat{\theta}_{t-1}$.

5. Filter $\{D_m''\}_{m=1}^{M}$ to $\{D_m''\}_{m=1}^{M}$ by $r_{\text{max}}$.

6. Project $\{D_m''\}_{m=1}^{M}$ to $\mathcal{X}'$.

7. Interpolate $\mathcal{X}'$ to $S'$ using SPSR.

8. Render $S'$ to $\{D_m'''\}_{m=1}^{M} \in \mathbb{R}^{H \times W}$.

9. Train $\hat{\theta}_t$ using $\{I_m\}_{m=1}^{M}$, $\{D_m'''\}_{m=1}^{M}$ and $l_{\text{pseudo}}$.

10. end for

11. return $\hat{\theta}_T$.
Figure 5: DTU Dataset. Representative point cloud results. Best viewed on screen.

Figure 6: Tanks and Temples. Representative point cloud results. Best viewed on screen.

Figure 7: Pseudo depth labels generated using different methods. (a) Ground-truth depth map. (b) Reference image. Top row, Columns 2-5: Pseudo depth label generated from different combination of following methods: Low Resolution Inference (LRI), High Resolution pseudo label Refinement (HRR), Pseudo label filtering (Filtering) and Multi-view pseudo Label Fusion (MLF). Bottom row is the error visualization of corresponding pseudo depth label. Areas with no pseudo depth labels are marked as blue in the error visualization. Best viewed on screen.
provided images and camera parameters for the proposed self-supervised learning framework. For the unsupervised initialization, we downsample the images in the training set into $512 \times 640$. For generating pseudo depth labels in the iterative self-training, we use the original $1600 \times 1200$ training images. For iterative self-training, we downsample the images in the training set into $160 \times 128$. We use the same training, validation and evaluation sets as defined in [26, 27]. We report the mean accuracy [1], mean completeness[1] and overall score [26].

**Tanks and Temples** [14] contains both indoor and outdoor scenes under realistic lighting conditions with large scale variations. We evaluate the generalization ability of our proposed self-supervised learning framework on the intermediate set. We report the mean $f$-score and the $f$-score for each scene in that set.

### 4.2. Self-supervised Learning

To demonstrate the performance of the incremental self-supervised learning framework, we use our approach to train a CVP-MVSNet [25] network on the DTU training dataset. No ground-truth depth training data is used. Tab. 1 shows the summary of performance of our results and existing unsupervised MVS networks. As shown, our method outperforms existing unsupervised MVS networks by a large margin. We perform qualitative comparison with supervised and unsupervised methods in Fig. 5 to further demonstrate the performance of our approach.

We also compare our self-supervised results obtained without any ground-truth training data to traditional geometric-based MVS supervised networks and supervised MVS networks, including recent methods with learning based view aggregation [29, 3, 30, 24] that outperform our backbone network[25] in supervised scenario. Tab. 3 summarizes this comparison. As shown, our approach provides competitive results compared to traditional and supervised networks. Tab. 2 shows a more detailed comparison between our approach and its supervised counter part [25]. Overall, the supervised approach achieves a slightly better $f$-score (+0.21%) and slight reduction in reconstruction error (0.012mm) at the expense, however, of needing ground-truth data.

### 4.3. Generalization Ability

To evaluate the generalization ability of the proposed self-supervised learning framework, we firstly train the CVP-MVSNet with self-supervised learning on DTU training dataset and directly test the model on Tanks and Temples dataset without any fine tuning. Further more, since our method does not rely on ground truth labels, we can apply the method on the training images of Tanks and Temples dataset. Results are listed in Tab. 4 and Fig. 6. Our results clearly outperform existing unsupervised MVS networks. Fine-tuning on Tanks and Temples training data can further boost performance (See first row of Tab. 4).

### 4.4. Ablation Study

Hereby we provide ablation studies analysis by evaluating the contribution of each part of our self-supervised approach to the final reconstruction quality. We also evaluate the ability of self-improving on reconstruction quality and discuss about the limitation of proposed method on texture-less areas. More ablation experiments and discussions can be found in supplementary material.

**Probability based image synthesis.** We first analyze the effect of probability based image synthesis by comparing our approach to directly warp image base on the estimated depth. Tab. 5 summarizes the results for this experiment. As shown, there is a significant performance improvement when using probability based image synthesis for unsupervised learning.

**Each part of the self-supervised learning framework.** In this experiment, we analyze the contribution of each part of the proposed self-supervised learning framework to the model performance. For the contribution of high resolu-

---

**Table 1: DTU Dataset.** Quantitative reconstruction results of unsupervised and self-supervised MVS networks

| Method       | Acc.↓ | Comp.↓ | Overall↓ (mm) |
|--------------|-------|--------|---------------|
| Khot et al. [13] | 0.881 | 1.073  | 0.977         |
| MVS² [6]     | 0.760 | 0.515  | 0.637         |
| M²VSNets [10] | 0.636 | 0.531  | 0.583         |
| Ours (self-sup.) | 0.308 | 0.418  | 0.363         |

**Table 2: DTU Dataset.** Quantitative reconstruction results of traditional, supervised MVS networks, and our self-supervised approach.

| Method       | Acc.↓ | Comp.↓ | Overall↓ (mm) |
|--------------|-------|--------|---------------|
| Furu [7]     | 0.613 | 0.941  | 0.777         |
| Tola [21]    | 0.342 | 1.190  | 0.766         |
| Camp [3]     | 0.835 | 0.554  | 0.695         |
| Gipuma [8]   | 0.283 | 0.873  | 0.578         |
| Colmap [17, 18] | 0.400 | 0.664  | 0.532         |
| MVSNet [26]  | 0.396 | 0.527  | 0.462         |
| Point-MVSNet [4] | 0.342 | 0.411  | 0.376         |
| Vis-MVSNet [9] | 0.325 | 0.385  | 0.355         |
| CVP-MVSNet [25] | 0.296 | 0.406  | 0.351         |
| UCSNet [5]   | 0.338 | 0.349  | 0.344         |
| PVA-MVSNet [29] | 0.379 | 0.336  | 0.357         |
| Vis-MVSNet [30] | 0.359 | 0.358  | 0.359         |
| PVSNets [24] | 0.337 | 0.315  | 0.326         |
| Ours (self-supervised) | 0.308 | 0.418  | 0.363         |

**Table 3: DTU dataset.** Quantitative reconstruction results of traditional, supervised MVS networks, and our self-supervised approach.
Table 4: **Tanks and Temples**. Performance as November 16, 2020. Our results clearly outperform existing unsupervised MVS networks.

| Model            | Rank ↓ | Mean↑ | Family↑ | Francis↑ | Horse↑ | M60↑ | Panther↑ | Playground↑ | Train↑ |
|------------------|--------|-------|---------|----------|--------|------|----------|-------------|--------|
| Ours (Self-sup. T&T) | 42.00  | 56.54 | 76.55   | 49.06    | 43.04  | 57.35| 60.64    | 57.35       | 58.47  | 50.06  |
| Ours (Self-sup. DTU) | 70.62  | 46.71 | 64.95   | 38.79    | 24.98  | 49.73| 52.57    | 44.95       | 47.39  | 30.31  |
| M³VSNet [10]   | 100.38 | 37.67 | 47.74   | 24.38    | 18.74  | 44.42| 43.45    | 44.95       | 47.39  | 30.31  |
| MVS² [6]       | 100.38 | 37.21 | 47.74   | 21.55    | 19.50  | 44.54| 44.86    | 46.32       | 43.48  | 29.72  |

Table 5: **DTU dataset**. Effect of the probability based image synthesis.

| Model                  | Acc↓ | Comp↓ | Overall↓ | f-score↑ |
|------------------------|------|-------|----------|----------|
| Depth Warping          | 0.447| 0.773 | 0.610    | 75.29%   |
| Probability Based      | **0.415** | **0.720** | **0.567** | **77.06%** |

Table 6: **DTU dataset**. Contribution of each part of the proposed incremental self-supervision framework on final reconstruction quality. Filtering: Pseudo label filtering. LRI: Low Resolution Inference. HRR: High Resolution pseudo label Refinement. MLF: Multi-view pseudo Label Fusion. HRR + MLF: Combining high resolution refinement and multi-view pseudo label fusion. LRI + MLF: Combining low resolution refinement and multi-view pseudo label fusion. LRI + Filtering: Combining low resolution refinement and low resolution pseudo label filtering. LRI + Filtered: Combining low resolution refinement and filtered pseudo label filtering. Filtered: Filtering pseudo labels.

| Model                          | Acc↓ | Comp↓ | Overall↓ | f-score↑ |
|--------------------------------|------|-------|----------|----------|
| Unsupervised                   | 0.415| 0.720 | 0.567    | 77.06%   |
| LRI + Filtering                | 0.322| 0.434 | 0.378    | 87.75%   |
| HRR + Filtering                | 0.316| 0.428 | 0.372    | 88.01%   |
| LRI + MLF + Filtering          | 0.325| 0.429 | 0.376    | 87.91%   |
| HRR + MLF + Filtering          | 0.308| 0.418 | 0.363    | 88.42%   |
| supervised                     | **0.296** | **0.406** | **0.351** | **88.61%** |

Table 7: **DTU dataset**. Performance at different self-supervised iterations.

| Self-supervision Itr. | Acc↓ | Comp↓ | Overall↓ | f-score↑ |
|-----------------------|------|-------|----------|----------|
| init                  | 0.415| 0.720 | 0.567    | 77.06%   |
| 1                     | 0.306| 0.431 | 0.368    | 88.16%   |
| 2                     | 0.308| 0.418 | 0.363    | 88.42%   |
| 3                     | 0.309| 0.420 | 0.364    | 88.49%   |
| 4                     | 0.309| 0.421 | 0.365    | 88.47%   |

**Figure 8:** Generating pseudo labels for severely texture-less areas is the current main limitation of our method. Best viewed on screen.

5. Conclusion

We proposed a self-supervised learning framework for depth inference from multiple view images. Given initial pseudo depth labels generated from a network under unsupervised learning process, we refine the flawed initial pseudo depth labels using a carefully designed pipeline. Using these refined pseudo depth labels as supervision signal, we achieve significantly better performance than state-of-the-art unsupervised networks and achieve similar performance compared to supervised learning frameworks. One current limitation of our approach is handling texture-less areas as the unsupervised stage fails to extract information to generate the initial pseudo labels. We will address this in our future work.

Acknowledgments

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

[1] Henrik Aanæs, Rasmus Ramsbol Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multi-view stereopsis. *IJCV*, 2016. 5, 7

[2] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008. 2, 7

[3] Rui Chen, Songfang Han, Jing Xu, et al. Visibility-aware point-based multi-view stereo network. *TPAMI*, 2020. 2, 7

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

[5] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *CVPR*, 2020. 1, 2, 7

[6] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry. In *3DV*, 2019. 1, 2, 3, 4, 7, 8

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *TPAMI*, 2010. 2, 7

[8] 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. 7

[9] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020. 1, 2, 7

[10] Baichuan Huang, Hongwei Yi, Can Huang, Yijia He, Jingbin Liu, and Xin Liu. M+3VSNet: Unsupervised multi-metric multi-view stereo network. *ArXiv*, 2020. 2, 7, 8

[11] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, 2016. 4, 10

[12] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ToG*, 2013. 5

[13] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *ArXiv*, 2019. 1, 2, 6, 7, 11

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

[15] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 2

[16] Arijit Mallick, Jörg Stückler, and Hendrik Lensch. Learning to adapt multi-view stereo by self-supervision. In *BMVC*, 2020. 1, 2

[17] Johannes Lutz Schönbberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 7

[18] Johannes Lutz Schönbberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2, 7

[19] 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

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 10

[21] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 2012. 2, 7

[22] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018. 10

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 2004. 10

[24] Qingshan Xu and Wenbing Tao. Pvsnet: Pixelwise visibility-aware multi-view stereo network. *ArXiv*, 2020. 1, 2, 7

[25] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *CVPR*, 2020. 1, 2, 3, 4, 6, 7, 11

[26] Yao Yao, Zixin Luo, Shihui Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 1, 2, 7

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

[28] Yao Yao, Zixin Luo, Shihui Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *CVPR*, 2020. 1, 2

[29] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In *ECCV*, 2020. 2, 7

[30] Jingyang Zhang, Yao Yao, Shihui Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *BMVC*, 2020. 2, 7

[31] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *Proc. SIGGRAPH*, 2018. 3
Supplementary Material

In this supplementary material, we first provide details of the image synthesis loss functions in section A. In section B, we show additional qualitative reconstruction results by our method for different scans on the DTU dataset. In section C, we provide visualization of the pseudo depth labels generated by our self-supervised learning framework. In section D, we show the pseudo depth labels generated by each iteration of our self-supervised learning framework. In section E, we provide more ablation experiments. In section F, we provide more discussions regarding the proposed method.

A. Image synthesis loss

In our approach, as introduced in section 3.2, we use a weighted combination of four loss functions,

\[ l_{syn} = \alpha_1 l_g + \alpha_2 l_{ssim} + \alpha_3 l_p + \alpha_4 l_s, \tag{5} \]

where \( l_g \) is the image gradient loss, \( l_{ssim} \) is the structure similarity loss, \( l_p \) is the perceptual loss, \( l_s \) is the depth smoothness loss, and \( \alpha_i \) sets the influence of each loss.

**Image gradient loss** is defined as the L1 distance between the gradient of input reference image \( \nabla I_0(x) \) and the synthesized image \( \nabla I_{l \rightarrow 0}(x) \) for each source view \( i \) and each pyramid level \( l \),

\[ l_g = \sum_{i=0}^{L} \sum_{x \in \Omega} \sum_{i=1}^{N} ||\nabla I_{l \rightarrow 0}(x) - \nabla I_0(x)||_1. \tag{6} \]

where \( \Omega \) is the set of valid pixels of the synthesized image.

**Structure similarity loss** enforces the contextual similarity between a synthesized image and the input reference image. Specifically, we use the Structure Similarity Index [23] to measure the contextual similarity. This index increases as the structure similarity between the images increases, with a range \([-1, 1]\). We formulate the loss as the negative of \( SSIM \) between each synthesized image and input reference image,

\[ l_{ssim} = \sum_{l=0}^{L} \sum_{i=1}^{N} 1 - SSIM(I_{l \rightarrow 0}, I_0). \tag{7} \]

**Perceptual loss** also encourages high-level contextual similarity between images [11]. This loss is defined as the L1 distance in the feature space of a shared weight perceptual network taking each image as input [11]. In our experiments, we use a VGG model [20] and extract features from \( 3_{th}, 8_{th}, 15_{th} \) and \( 22_{th} \) layers. Therefore, we formulate the loss as follows,

\[ l_p = \sum_{l=0}^{L} \sum_{i=1}^{N} \sum_{j \in [3, 8, 15, 22]} ||VGG(I_{l \rightarrow 0}, j) - VGG(I_0, j)||_1. \tag{8} \]

**Depth smoothness loss** encourages local depth smoothness. This term encourages depth smoothness with respect to the alignment of image and depth discontinuities, which is measured by the gradient of color intensity with respect to the alignment of image and depth discontinuities. We define this loss as follows,

\[ l_{sm} = \sum_{l=0}^{L} \sum_{x \in \Omega} ||\nabla_u I_{l \rightarrow 0}(x)|| + ||\nabla_v I_{l \rightarrow 0}(x)|| \tag{9} \]

where \( \nabla_u \) and \( \nabla_v \) refer to the gradient on \( x \) and \( y \) direction, and \( D = D/D \) is the mean-normalized inverse depth [22].

B. Qualitative results on DTU dataset

Fig. 9 shows additional reconstruction results by our self-supervised method on the DTU dataset. As shown, our self-supervised model can achieve similar reconstruction results comparing with the supervised model.

C. Pseudo depth labels visualization

Fig. 10 and Fig. 11 provide visualization of pseudo depth labels generated by our self-supervised learning framework. As shown, our method can generate high quality pseudo depth labels for rich-texture areas. However, our method cannot generate pseudo depth labels for severely textureless regions.

D. Pseudo labels generated on each iteration

Fig. 12 shows visualization of pseudo depth labels generated by the different iterations of the self-supervised learning framework. As shown, pseudo depth labels tend to become stable after the second iteration.

E. Additional Ablation Experiments

Use geometric MVS methods as initialization.

We use a traditional MVS method OpenMVS to generate the pseudo labels to replace our unsupervised learning process on DTU dataset. Specifically, we render a depth map from the mesh generated by OpenMVS and treat it as the label for the first iteration of the iterative training. As shown in Tab. 8, using OpenMVS outputs as pseudo labels can achieve similar performance as our proposed initialization method after 3 iterations of self-supervised learning.

| Method/init | Iter. 1 | Iter. 2 | Iter. 3 |
|-------------|---------|---------|---------|
| OpenMVS Init | 73.70%  | 76.86%  | 87.95%  | 88.02%  |
| Ours Init.   | 77.06%  | 88.16%  | 88.42%  | 88.49%  |

Table 8: Performance with different initialization method.
Figure 9: DTU Dataset. Representative point cloud results. Best viewed on screen.
Figure 10: Pseudo depth labels generated by the self-supervised learning framework. Areas with no pseudo depth labels or no ground-truth depth are marked as blue in the error visualization. Best viewed on screen.
Figure 11: Pseudo depth labels generated by the self-supervised learning framework. Areas with no pseudo depth labels or no ground-truth depth are marked as blue in the error visualization. Best viewed on screen.
Figure 12: Pseudo depth labels generated by each iteration of the iterative self-supervised learning framework. (a) Ground-truth depth map. (b-e) Pseudo depth labels generated by 1-4 iteration of the self-supervised learning framework. (f) Reference image. (g-j) Error of each iteration of the pseudo depth labels. Areas with no pseudo depth labels are marked as blue in the error visualization. Best viewed on screen.

F. Discussions

Runtime Despite the limitation on texture-less area mentioned in main paper, another limitation appears on the runtime of proposed self-supervised learning method. Each iteration of the self-training process takes around 15 hours on our machine, which adds up to days for several iterations of self-training or fine-tuning on novel data. For comparison, the supervised CVP-MVSNet takes around 10 hours. A classical MVS method such as the OpenMVS even does not need any training to achieve compromised results. Improving the efficiency of the proposed learning method can be an direction of future research.

Limitation of geometric processing Another concern appears on the geometric filtering and fusion methods we used to refine the pseudo depth label. The traditional geometric methods such as consistency check and Screened Poisson Surface Reconstruction have their limitations. Specifically, the SPSR is a non-learning, indifferentiable and time-consuming step, which might be too heavy for fine-tuning on novel data. It also have limited performance on very complex scenes. Improving pseudo label processing methods can be a direction of future research.