Pattern Recognition 136 (2023) 109198

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Prior depth-based multi-view stereo network for online 3D model reconstruction

Soohwan Song[a,1], Khang Giang Truong[b,1], Daekyum Kim[c], Sungho Jo[b,*]

[a] Intelligent Robotics Research Division, ETRI, Daejeon 34129, Republic of Korea
[b] School of Computing, KAIST, Daejeon 34141, Republic of Korea
[c] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA

ARTICLE INFO

Article history:
Received 27 December 2021
Revised 7 May 2022
Accepted 20 November 2022
Available online 22 November 2022

Keywords:
Multi-view stereo
Deep learning
Online 3D reconstruction

ABSTRACT

This study addresses the online multi-view stereo (MVS) problem when reconstructing precise 3D models in real time. To solve this problem, most previous studies adopted a motion stereo approach that sequentially estimates depth maps from multiple localized images captured in a local time window. To compute the depth maps quickly, the motion stereo methods process down-sampled images or use a simplified algorithm for cost volume regularization; therefore, they generally produce reconstructed 3D models that are inaccurate. In this paper, we propose a novel online MVS method that accurately reconstructs high-resolution 3D models. This method infers prior depth information based on sequentially estimated depths and leverages it to estimate depth maps more precisely. The method constructs a cost volume by using the prior-depth-based visibility information and then fuses the prior depths into the cost volume. This approach significantly improves the stereo matching performance and completeness of the estimated depths. Extensive experiments showed that the proposed method outperforms other state-of-the-art MVS and motion stereo methods. In particular, it significantly improves the completeness of 3D models.

1. Introduction

Many computer vision and robotics applications require the precise 3D reconstruction of environments. Multi-view stereo (MVS) [1–3] is one of the most widely used approaches in 3D reconstruction. MVS reconstructs the structures in 3D by finding dense correspondences across a collection of calibrated images captured from multiple viewpoints. MVS is particularly effective when reconstructing large-scale scenes because it can accurately estimate a wide range of depths from various baseline observations. However, general MVS approaches [1–4] run offline; they require several hours to process all the acquired images in a batch. Therefore, MVS has received less attention in applications that require real-time scanning like robot perception, compared to other approaches, such as LiDAR or depth-camera-based reconstruction [5,6].

Motion stereo [7,8] addresses the problem of online MVS. Motion stereo estimates depth maps in real time using multiple images captured by a localized, moving monocular camera. It uses a set of sequential images in a local time window as the source images to solve the MVS problem. However, since existing studies aim at collision detection for mobile robots or augmented reality applications, they focused only on estimating coarse 3D scenes instead of performing a detailed reconstruction. Most approaches [7,9] processed down-sampled images or cost volumes to reduce computation times, which as a result decreases 3D reconstruction performance. Furthermore, for these approaches, it is difficult to apply precise cost volume regularization or exhaustive noise filtering due to their time constraints. Therefore, existing motion stereo methods are limited to perform precise 3D reconstruction compared to original MVS methods [1–3].

One way to overcome such issue is to use *Convolutional Neural Networks* (CNNs) for online MVS. It has been shown that using CNNs for MVS [10–13] improves the overall reconstruction quality while significantly reducing the computation time compared to conventional methods [2,3]. A previous study [14] proposed an online MVS system by applying the CNN-based method [11] to motion stereo. This system focused on accurate and detailed reconstruction instead of the fast depth computation considered in conventional motion stereo [7,8]. The system computes camera poses using a keyframe-based SLAM [15] and estimates

[*] Corresponding author.
E-mail addresses: soohwansong@etri.re.kr (S. Song), khangtg@kaist.ac.kr (K.G. Truong), daekyum@seas.harvard.edu (D. Kim), shjo@kaist.ac.kr (S. Jo).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.patcog.2022.109198
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

depth maps of the keyframes based on MVS. The CNN-based MVS method [11] can rapidly compute a depth map within the time interval of keyframe extraction. Therefore, the system not only enabled to reconstruct dense 3D models online but also achieved a remarkable modeling performance compared to existing motion stereo methods [7,8].

However, for this method [14], the online modeling performance is still considerably lower compared to existing offline modeling. This method uses insufficient source views in a restricted time window. The source views may produce incomplete coverage of a reference view, which causes many stereo matching errors from invisible pixels and temporally inconsistent depths to be created. Furthermore, the restricted views make it difficult to completely filter out outliers in the background and occluded regions. In contrast, the offline modeling method can exhaustively check the consistency of depths from all available viewpoints for outlier filtering.

This study proposes a novel deep-learning-based MVS method that performs accurate online 3D modeling based on integrated prior information. This approach consists of 1) Prior depth estimation network and 2) Prior depth-based MVS network. Prior depth estimation network predicts a prior depth map based on the previously estimated depths. Unlike existing sequential depth propagation [8,9,16], it considers geometrically consistent depths from source views to produce noise-suppressed prior depths. Then, MVS network constructs a cost volume using the prior-depth-based visibility information and probabilistically integrates the prior depths into the cost volume. This approach significantly improves depth estimation accuracy by reducing stereo matching errors that caused by invisible pixels. It also estimates temporally consistent depths without sequential error propagation, which is mainly occurred in the existing motion stereo methods [8,9,16]. Furthermore, our method predicts the confidence of depth estimates by directly learning true depth errors. Based on the predicted confidences, the proposed method can effectively remove outliers even with restricted views. Finally, the proposed method greatly improves the completeness and accuracy of 3D modeling when integrating with online MVS system as well as offline MVS.

The contributions of this work are:

- A novel MVS framework for online 3D modeling that predicts prior depth information based on the depth maps of source views and effectively utilizes the prior information to estimate a current depth map. The proposed online approach makes it possible to reconstruct a precise 3D model with high completeness and accuracy using restricted source views.
- A network structure that predicts a noise-suppressed prior depth map by fusing the geometrically consistent depths of source views via *a normalized CNN* (NCNN) [17].
- An MVS network for prior depth integration. The network formulates a cost volume based on visibility information and integrates the prior depth values into the cost volume probabilistically. This method generates accurate and temporally consistent depth maps.
- A confidence prediction network that learns true depth errors based on aleatoric uncertainty loss [18]. This network estimates confidences more accurately than the cost-distribution-based estimation [10].
- The proposed method is evaluated on two MVS benchmarks: the *DTU dataset* [19] and the *Tanks and Temples dataset* [20]. The performance of online 3D modeling was also evaluated using two aerial scenes.

2. Related work

2.1. Multi-view stereo

Traditional MVS studies can be classified into three categories: volumetric [4,21], point cloud-based [1], and depth map-based [2,3] methods. These methods rely on handcrafted features for stereo matching, and generally underperform on scenes that contain untextured or specularly reflected surfaces.

Recently, many studies have adopted deep-learning approaches for MVS, which has gained considerable performance improvements. Many studies used 2D CNNs to improve the performance of stereo matching or depth map fusion. Hartmann et al. [22] trained 2D CNNs from matching and non-matching patches to measure a similarity of multiple image patches. Several researchers [23,24] have used 3D CNNs instead of 2D features to take advantage of 3D volume information. Ji et al. [24] also trained 3D CNNs that represent the volume-wise geometric context to regularize and classify surfaces in a voxel space. Although these methods [23,24] explicitly predict global 3D surfaces, they may suffer from precision deficiency and huge memory requirements for volume representations.

To overcome the limitations of volumetric methods [23,24], Yao et al. [10] proposed a depth-map-based method, MVSNet. It generates the matching cost volume based on the extracted 2D deep features and then applies 3D CNNs for cost-volume regularization. This approach can successfully estimate accurate depth maps using end-to-end deep learning architecture. However, because the cost volume requires large memory consumption cubic to the image resolution, high-resolution images cannot be processed. To address this issue, several studies [11–13] applied a cascade formulation of multi-stage cost volumes. Gu et al. [11] proposed the *cascade MVS network* (CasMVSNet) which uses multiple small cost volumes instead of a single large cost volume to reduce the computation time and GPU memory required. It progressively estimates depths in a coarse-to-fine fashion by restricting the range and number of hypothesis planes at each stage. In this work, we extend CasMVSNet to also account for the sequential depth information. Our method first estimates the prior depth information from previously estimated depth maps and then integrates the prior depth into each cascade stage to compute more accurate and complete depths. Our method also estimates pixel-wise visibility information to perform more accurate stereo matching at each stage.

2.2. Sequential depth propagation

Propagating sequential depth information can provide stable and temporally consistent results. Sequential depth propagation has popularly been applied in video depth estimation methods, such as depth prediction [25] and motion stereo [9,16]. Depth prediction can estimate the depth information in a single image by directly training the correlations between visual cues and absolute depths. Yang et al. [25] integrated sequential depth and uncertainty predictions using a Bayesian fusion approach.

Motion stereo solves a sub-problem of MVS by estimating depth maps online using a localized moving monocular camera. Given a captured current image, it uses a set of sequential images in a local time window as source images and solves the independent MVS problem. Several motion stereo methods [9,16] have propagated sequential depth information to estimate stable depth maps. Hou et al. [9] used Gaussian process models to propagate the previously estimated depths through a probabilistic prior in a latent space. Liu et al. [16] sequentially propagated the depth-probability distribution estimated from a cost volume.

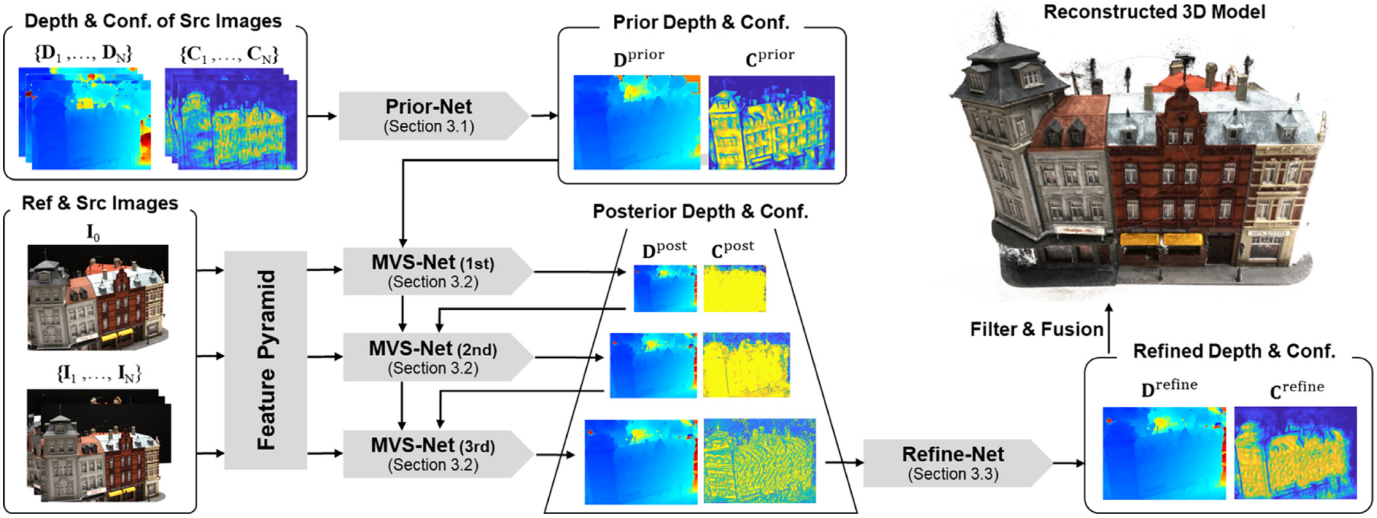


Fig. 1. Architecture of the proposed method for MVS depth estimation. It is composed of three main networks: (a) Prior-Net: predicting the prior depth and confidence from the previously estimated depths and confidences of the source views (Section 3.1). (b) MVS-Net: performing MVS depth estimation by integrating the prior depth information in the cascade cost-volume structure [11] (Section 3.2). (c) Refine-Net: refining the depth and confidence maps estimated by MVS-Net (Section 3.3).

As mentioned earlier, these methods [9,16] focus on fast depth computation instead of precise 3D model reconstruction. Therefore, their depth estimates may produce incomplete 3D models containing many outliers, and the errors are propagated throughout consecutive frames. On the other hand, our method aims to reconstruct a precise online 3D model. Noise-suppressed prior depths are predicted using geometrically consistent depths of multiple source views. Our method also predicts the confidence of prior depths to determine their reliability. The prior depths and confidences are then probabilistically propagated into a cost volume to estimate globally consistent depths.

2.3. Visibility estimation

Visibility information helps reduce errors induced by invisible pixels during stereo matching. Many visibility estimation approaches have been applied in traditional MVS methods [3]. On the other hand, visibility estimation for deep-learning-based MVS methods has received relatively less attention. Only a few visibility-estimation methods [26,27] for deep-learning-based MVS have been proposed. Most methods predict a pixel-wise visibility map for each source image by using two-view cost volumes [26] or warped image features [27]. They applied visibility-based weighted sum approaches to construct an aggregated cost volume and regressed a depth map from the cost volume. Zhang et al. [26] measured the entropy of a depth-probability distribution on a two-view cost volume to predict the visibility map.

Like Zhang et al. [26], our method predicts the pixel-wise visibility map based on two-view cost volumes. Moreover, our method considers the prior depth and previously estimated depth of a source image, which provides important information for visibility prediction. The application of this visibility information significantly improves the overall performance of stereo matching.

3. Prior depth-based MVS network

This study addresses the motion stereo problem where the depth map of a current frame captured from a moving monocular camera is sequentially estimated. Given a reference image $\mathbf{I}_0$ and N source images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ with corresponding camera parameters $\{Q_0, \ldots, Q_N\}$, the depth map $\mathbf{D}$ and confidence map $\mathbf{C}$ for $\mathbf{I}_0$ are estimated using the MVS method. The depths and confidences $\{\mathbf{D}_k, \mathbf{C}_k\}_{k=1}^N$ of source images were already computed and were known; therefore, we could use them as prior information for the MVS depth estimation problem, unlike in conventional studies.

Fig. 1 depicts the proposed network architecture that effectively integrates prior depth information to allow for MVS depth map estimation. The network consists of three stages: **Prior-Net** (Section 3.1), **MVS-Net** (Section 3.2), and **Refine-Net** (Section 3.3). Prior-Net first predicts a prior depth map $\mathbf{D}^{prior}$ with the corresponding confidence map $\mathbf{C}^{prior}$ of the current view using the previously estimated depths and confidences of the source views. To predict the prior depths, we fuse geometrically consistent depths of source views by confidence-equipped depth propagation of NCNN [17]. The proposed method significantly reduces the possibility of error propagation that frequently occurs in other sequential depth integration methods [9,16].

Next, MVS-Net performs MVS depth estimation by leveraging the prior depth information. We adopted the cascading cost volume formulation of CasMVSNet [11] as the MVS-Nets baseline architecture. It progressively narrows the range of the depth hypothesis using a cascade formulation of multiple stages. This approach efficiently processes high-resolution images with relatively low GPU memory usage and fast computation time. For each cascade stage, we formulate a cost volume through visibility-based stereo matching. Unlike existing methods [26,27], the proposed method accurately predicts the pixel-wise visibility from the prior information. MVS-Net then integrates $\mathbf{D}^{prior}$ and $\mathbf{C}^{prior}$ into the cost volume via Bayesian filtering. The depth map $\mathbf{D}^{post}$ and confidence map $\mathbf{C}^{post}$ are estimated from the posterior distribution of the cost volume.

Finally, Refine-Net refines the estimated $\mathbf{D}^{post}$ and $\mathbf{C}^{post}$ based on the photometric features of the reference image. It directly trains true depth errors to estimate confidences more accurately compared to the cost-distribution-based estimation [10]. The refined depth map $\mathbf{D}^{refine}$ and confidence map $\mathbf{C}^{refine}$ are stored in a database and used for future depth estimation.

3.1. Prior depth prediction

In the first stage, Prior-Net takes the depth maps $\{\mathbf{D}_1, \ldots, \mathbf{D}_N\}$ and their confidence maps $\{\mathbf{C}_1, \ldots, \mathbf{C}_N\}$ of the source views as inputs and predicts a single prior depth $\mathbf{D}^{prior}$ and the corresponding confidence $\mathbf{C}^{prior}$ for a reference view. Fig. 2 depicts the network architecture of Prior-Net.

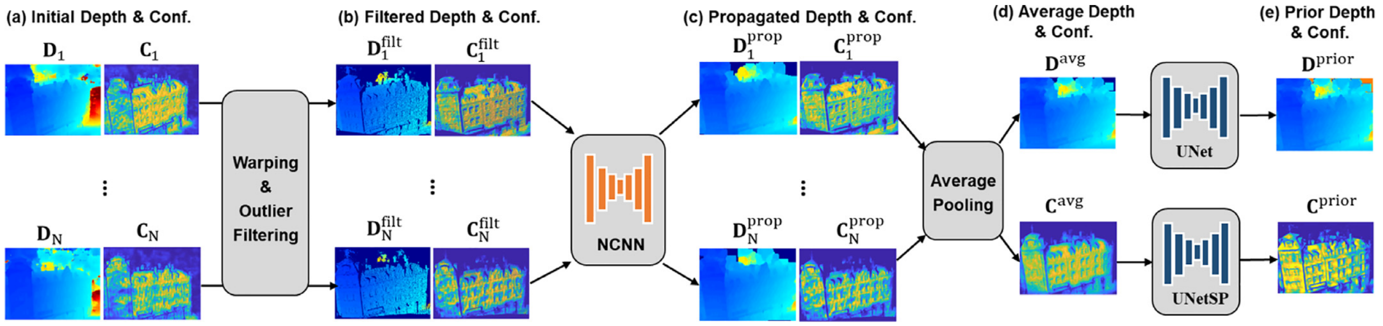


Fig. 2. The network architecture of Prior-Net. (a) Given the depth maps $\{\mathbf{D}_1, \ldots, \mathbf{D}_N\}$ and confidence maps $\{\mathbf{C}_1, \ldots, \mathbf{C}_N\}$ of source views, (b) Prior-Net first projects them onto the reference view and filters their outliers. (c) The filtered depths $\{\mathbf{D}_1^{filt}, \ldots, \mathbf{D}_N^{filt}\}$ and confidences $\{\mathbf{C}_1^{filt}, \ldots, \mathbf{C}_N^{filt}\}$ are fed into the NCNN [17] to generate the propagated depths $\{\mathbf{D}_1^{prop}, \ldots, \mathbf{D}_N^{prop}\}$ and confidences $\{\mathbf{C}_1^{prop}, \ldots, \mathbf{C}_N^{prop}\}$. (d) Prior-Net then integrates them into an averaged depth $\mathbf{D}^{avg}$ and confidence $\mathbf{C}^{avg}$. (e) Finally, the prior depth $\mathbf{D}^{prior}$ and confidence $\mathbf{C}^{prior}$ are estimated by refining $\mathbf{D}^{avg}$ and $\mathbf{C}^{avg}$, respectively.

3.1.1. Outlier filtering

To predict the prior depth map, our method only considers the reliable depth information of each depth map. It applies strict outlier filtering and produces filtered depth maps $\{\mathbf{D}_1^{filt}, \ldots, \mathbf{D}_N^{filt}\}$ and confidence maps $\{\mathbf{C}_1^{filt}, \ldots, \mathbf{C}_N^{filt}\}$. Our method first removes depths with confidence values lower than a defined threshold. Then, similar to the method in [2], it checks the geometric consistencies of the warped depths for outlier filtering. The confidences are also filtered according to the filtered depth pixels.

3.1.2. Depth propagation

The filtered depth maps $\{\mathbf{D}_1^{filt}, \ldots, \mathbf{D}_N^{filt}\}$ are composed of reliable depths; however, they are relatively sparse and may contain some empty regions. Therefore, our method generates interpolated depth maps by propagating the reliable depths into the neighboring regions using NCNN. NCNN propagates the high-confidence depths into the neighboring low-confidence or empty regions through consecutive CNN layers. The confidences are also inertially propagated between the CNN layers. Therefore, NCNN can produce not only propagated depth maps $\{\mathbf{D}_1^{prop}, \ldots, \mathbf{D}_N^{prop}\}$ but also propagated confidence maps $\{\mathbf{C}_1^{prop}, \ldots, \mathbf{C}_N^{prop}\}$ using confidence-equipped CNN layers. Similar to [17], the UNet architecture [28] is used in NCNN, which allows for the effective propagation of multi-scale information, to allow for depth interpolation.

3.1.3. Average pooling

Next, Prior-Net integrates the propagated depth maps $\{\mathbf{D}_1^{prop}, \ldots, \mathbf{D}_N^{prop}\}$ and confidence maps $\{\mathbf{C}_1^{prop}, \ldots, \mathbf{C}_N^{prop}\}$ into a single depth map $\mathbf{D}^{avg}$ and confidence map $\mathbf{C}^{avg}$. It applies weighted mean and standard mean operation to output the single depth and confidence, respectively:

$$\mathbf{D}^{avg} = \frac{\sum_{i=1}^{N} \mathbf{D}_i^{prop}\mathbf{C}_i^{prop}}{\sum_{i=1}^{N} \mathbf{C}_i^{prop}}, \quad \mathbf{C}^{avg} = \frac{\sum_{i=1}^{N} \mathbf{C}_i^{prop}}{N} \tag{1}$$

NCNN may produce wrong depth completion results when input depth maps contain large holes. However, the influence of wrong depth completion can be eliminated by this confidence-based average pooling. We only consider the geometrically consistent depths; their integration provides a reliable result. Since each $\mathbf{C}_i^{prop}$ represents the reliability with the density of the consistent depths in $\mathbf{D}_i^{prop}$, $\mathbf{C}^{avg}$ also reflects the integrated density of the consistent depths. Sparse depth or empty regions have low-confidence values in $\mathbf{C}^{avg}$ because most of the propagated depths with low confidences in $\mathbf{C}_k^{prop}$ are integrated. On the other hand, non-empty regions on each filtered depth guarantee integration of some consistent depths; therefore, the densely distributed depths produce high confidence integration results in $\mathbf{C}^{avg}$.

3.1.4. Prior depth and confidence prediction

Finally, the prior depth map $\mathbf{D}^{prior}$ and confidence map $\mathbf{C}^{prior}$ are produced by refining the averaged depth map $\mathbf{D}^{avg}$ and confidence map $\mathbf{C}^{avg}$. Our method feeds $\mathbf{D}^{avg}$ and $\mathbf{C}^{avg}$ into two UNet architectures and then refines them independently. For depth map refinement, our method employs a UNet block with a depth of 1 and several residual blocks [29]. For confidence refinement, our method first estimates the noise variance $\tilde{\mathbf{C}}^{prior}$ by using a UNet block with a depth of 3 and a Softplus activation function, where the noise variance represents the uncertainty of the predicted depth. Then, our method transforms the uncertainties of $\tilde{\mathbf{C}}^{prior}$ into normalized confidences of $\mathbf{C}^{prior}$ between 0 and 1 as follows:

$$\mathbf{C}^{prior} = \exp\left(-\frac{\tilde{\mathbf{C}}^{prior^2}}{2\sigma_{conf}^2}\right) \tag{2}$$

where σ_{conf} is a constant value.

To train Prior-Net, we employed the aleatoric uncertainty loss [18], which is the negative log-likelihood of a Gaussian distribution corresponding to the L_2 loss. However, because the L_2 loss often has a bias toward pixels with high depth values in the background, we applied a new L_1 loss that considers Laplace distribution instead of Gaussian distribution, as in the original aleatoric uncertainty loss [18,30].

$$\mathcal{L}_{prior} = \frac{1}{M}\sum_{x}\left(\frac{|\mathbf{D}^{prior}(x) - \mathbf{D}^{gt}(x)|}{\tilde{\mathbf{C}}^{prior}(x)} + \log\tilde{\mathbf{C}}^{prior}(x)\right) \tag{3}$$

where M is the total number of pixels, and $\mathbf{D}^{prior}(x)$ and $\mathbf{D}^{gt}(x)$ are the prior and ground truth depths at pixel x, respectively. $\tilde{\mathbf{C}}^{prior}(x)$ represents the noise variance of $\mathbf{D}^{prior}(x)$.

3.2. MVS Depth estimation

The architecture of MVS-Net follows the network structure of CasMVSNet [11]. We extend CasMVSNet to also consider prior depth information. We composed three cascade stages to estimate depth maps at different scales $\{\frac{1}{16}, \frac{1}{4}, 1\}$ in a coarse-to-fine manner. It progressively reduces the depth hypothesis range of a cost volume based on the depths estimated at the previous stage.

MVS-Net first extracts multi-scale deep features for each image and constructs the cascade cost volumes by fusing the extracted features. For each image $\mathbf{I}_k$, the feature pyramid network [31] is applied to extract three photometric features $\{\mathbf{F}_{k,1}^{photo}, \mathbf{F}_{k,2}^{photo}, \mathbf{F}_{k,3}^{photo}\}$ at multiple scales. Each feature map is used to generate a cost volume with the same spatial resolution at each stage.

Fig. 3 shows the architecture of the MVS-Net at a single stage; the stage number s is omitted for simplicity. MVS-Net first pre-

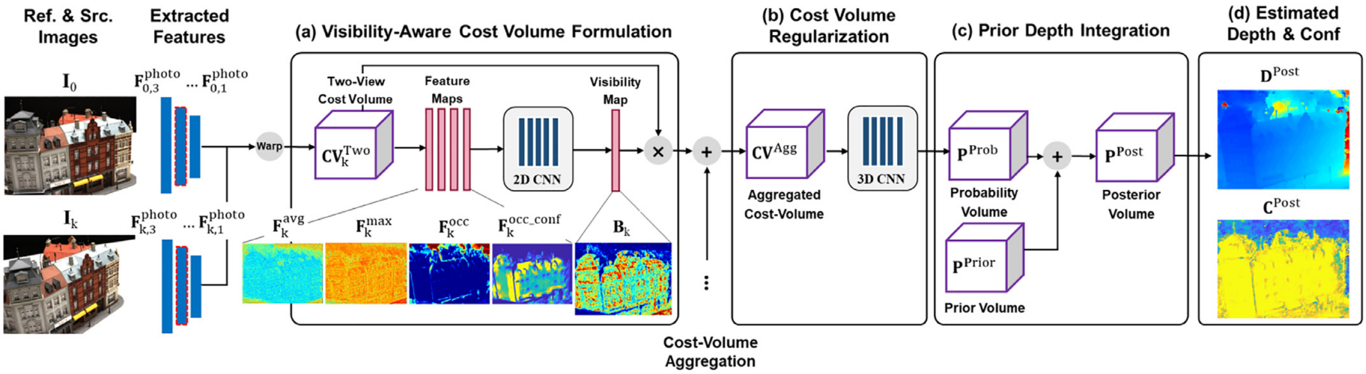


Fig. 3. The network architecture of MVS-Net. (a) MVS-Net produces the two-view cost volume $\mathbf{CV}_k^{two}$ with visibility map $\mathbf{B}_k$ of each source image $\mathbf{I}_k$. It then aggregates all two-view cost volumes into a single cost volume $\mathbf{CV}^{agg}$ using a weighted sum. (b) MVS-Net applies 3D CNNs to produce the probability volume $\mathbf{P}^{prob}$ from $\mathbf{CV}^{agg}$. (c) Next, it integrates $\mathbf{P}^{prob}$ with the prior probability volume $\mathbf{P}^{prior}$ into the posterior probability volume $\mathbf{P}^{post}$. (d) Finally, depth map $\mathbf{D}^{post}$ and confidence map $\mathbf{C}^{post}$ are estimated from $\mathbf{P}^{post}$.

dicts pixel-wise visibility $\mathbf{B}_k$ for each source image $\mathbf{I}_k$ using prior depth information. It then generates a cost volume $\mathbf{CV}^{agg}$ by aggregating multiple matching costs $\{\mathbf{CV}_1^{two}, \ldots, \mathbf{CV}_N^{two}\}$ while also considering their visibility maps $\{\mathbf{B}_1, \ldots, \mathbf{B}_N\}$. The visibility information is used to reduce the influence of mismatching errors in the occluded areas. Next, MVS-Net produces a probability volume $\mathbf{P}^{prob}$ by regularizing $\mathbf{CV}^{agg}$ using 3D CNNs. The probability volume represents the probability distribution of each depth direction. The prior depth $\mathbf{D}^{prior}$ is then directly integrated into $\mathbf{P}^{prob}$, resulting in the posterior depth probability distribution $\mathbf{P}^{post}$. Finally, the depth map $\mathbf{D}^{post}$ is estimated by computing the expectation value along each depth direction on $\mathbf{P}^{post}$. The estimated depth map is propagated into the next stage to reduce the range of the depth hypothesis for the higher-resolution image. The following descriptions detail each component of the MVS-Net in a single stage.

3.2.1. Visibility-aware cost volume formulation

For each source image $\mathbf{I}_k$, MVS-Net warps the feature map $\mathbf{F}_k^{photo}$ into different fronto-parallel planes of the reference image $\mathbf{I}_0$ using differentiable homography. A two-view cost volume $\mathbf{CV}_k^{two}$ is generated by computing the group-wise correlations between the reference and warped feature maps [26]. MVS-Net then predicts a visibility map $\mathbf{B}_k$ representing the pixels visibility in the source image $\mathbf{I}_k$. In contrast to existing methods [26,27] that focus only on matching the cost distribution for visibility prediction, MVS-Net applies an occlusion-aware strategy in that it considers the occluded areas estimated from the prior depth.

Fig. 3 a shows the architecture of the visibility prediction network. It is a lightweight 2D CNN composed of several 2D convolutional and ReLU blocks. The network takes the maximum and average cost matching information $\mathbf{F}_k^{\max}$ and $\mathbf{F}_k^{avg}$, respectively of a two-view cost volume $\mathbf{CV}_k^{two}$ as input features. The highest and average cost matching information implicitly reflects the saliency or matching quality [27]. Like the MVSNet-based methods [10,11,27], we applied a weighted Euclidean distance to estimate the matching where a $\{1 \times 1 \times 1\}$ convolution layer trains the weights. The matching cost is interpreted as a matching score that has a high value when two matched features are similar. Therefore, we use the maximum cost instead of the minimum cost to measure a matching quality. Similar to PVA-MVSNet [27], we do not normalize these feature maps $\mathbf{F}_k^{\max}$ and $\mathbf{F}_k^{avg}$. A normalized feature value at a pixel could vary significantly depending on the number of depth hypotheses, which leads to inconsistency of visibility prediction at different cascade stages. Therefore, normalizing cost volume before extracting $\mathbf{F}_k^{\max}$ and $\mathbf{F}_k^{avg}$ will not benefit the visibility prediction.

Furthermore, the network takes additional input features $\mathbf{F}_k^{occ}$ and $\mathbf{F}_k^{occ\_conf}$, which represents the occluded area and its confidence, respectively. The occluded area of a source image is roughly estimated from the prior depth information, which is strongly related to the visibility. Given a propagated depth map $\mathbf{D}_k^{prop}$ computed as described in Section 3.1 and a prior depth map $\mathbf{D}^{prior}$, the feature map $\mathbf{F}_k^{occ}$ representing the occluded area is computed using the relative difference

$$\mathbf{F}_k^{occ} = \frac{|\mathbf{D}_k^{prop} - \mathbf{D}^{prior}|}{\mathbf{D}^{prior}} \tag{4}$$

where $\mathbf{F}_k^{occ}$ is resized to the resolution of output depth map at a stage. Since several sub-regions of $\mathbf{D}_k^{prop}$ and $\mathbf{D}^{prior}$ may be inaccurate, their confidence $\mathbf{C}_k^{prop}$ and $\mathbf{C}^{prior}$ should also be considered. The feature map $\mathbf{F}_k^{occ\_conf}$ representing the confidence of $\mathbf{F}_k^{occ}$ is computed as

$$\mathbf{F}_k^{occ\_conf} = \min(\mathbf{C}_k^{prop}, \mathbf{C}^{prior}) \tag{5}$$

where $\mathbf{C}_k^{prop}$ is the propagated confidence map, and $\mathbf{C}^{prior}$ is the prior confidence map that has a same scale with $\mathbf{C}_k^{prop}$.

For each source image $\mathbf{I}_k$, the visibility prediction network predicts the visibility map $\mathbf{B}_k$ from the extracted features $\{\mathbf{F}_k^{\max}, \mathbf{F}_k^{avg}, \mathbf{F}_k^{occ}, \mathbf{F}_k^{occ\_conf}\}$. Pixels with low visibility are more likely to be occluded in the reference view; therefore, their matching cost should have a small impact on cost aggregation. Given the predicted visibility maps $\{\mathbf{B}_1, \ldots, \mathbf{B}_N\}$, the two-view cost volumes $\{\mathbf{CV}_1^{two}, \ldots, \mathbf{CV}_N^{two}\}$ are aggregated into a single cost volume $\mathbf{CV}^{agg}$ by the weighted sum:

$$\mathbf{CV}^{agg} = \frac{\sum_{k=1}^{N} \mathbf{B}_k \otimes \mathbf{CV}_k^{two}}{\sum_{k=1}^{N} \mathbf{B}_k} \tag{6}$$

where $\otimes$ represents the element-wise multiplication operation. This approach reduces the influence of mismatching errors induced by invisible pixels in advance. We further provide the illustration of the feature maps and the visibility map in the supplementary material (Section 4).

3.2.2. Cost volume regularization

The aggregated cost volume $\mathbf{CV}^{agg}$ generally contains a large amount of noise. Therefore, MVS-Net applies 3D CNNs for cost-volume regularization to reduce the noise and enforce the smoothness constraint [10]. It then normalizes the regularized cost volume along each depth direction using the softmax function. The normalized cost volume refers to the probability volume $\mathbf{P}^{prob}$, which is generally used in per-pixel depth probabilities; $\mathbf{P}^{prob}(x, d)$ represents the probability that pixel x has a depth value d.

3.2.3. Prior depth integration and depth regression

Next, our method integrates the prior depth information into $\mathbf{P}^{prob}$. The prior depth provides complementary information for the results of current stereo matching and helps produce temporally stable depths with improved accuracy. Our method formulates a prior probability distribution from $\mathbf{D}^{prior}$ and $\mathbf{C}^{prior}$ and then integrates it with the cost distribution on $\mathbf{P}^{prob}$ via Bayesian filtering. This integration approach considers the posterior probability distribution of a depth value instead of considering a single depth value; therefore, our method is able to mitigate the effect of mismatching or error propagation implicitly through statistical distributions.

Given a prior depth map $\mathbf{D}^{prior}$ and confidence map $\mathbf{C}^{prior}$, our method first constructs a prior probability volume $\mathbf{P}^{prior}$. For each depth direction of a pixel x on $\mathbf{P}^{prior}$, our method generates a unimodal distribution peaked at the prior depth $\mathbf{D}^{prior}(x)$ with the noise variance $\tilde{\mathbf{C}}^{prior}(x)$. The probability of a pixel x having a depth value d on the unimodal distribution is defined as [32]

$$\mathbf{P}^{prior}(x,d) = \text{softmax}\left(-\frac{|d-\mathbf{D}^{prior}(x)|}{\tilde{\mathbf{C}}^{prior}(x)}\right) \tag{7}$$

where $\tilde{\mathbf{C}}^{prior}(x)$ controls the sharpness of the peak around $\mathbf{D}^{prior}(x)$. Our method then integrates the prior probability $\mathbf{P}^{prior}$ volume with the current probability volume $\mathbf{P}^{prob}$ using Bayesian filtering. For each pixel x and depth d, the posterior probability is computed as

$$\mathbf{P}^{post}(x,d) = \frac{\mathbf{P}^{prior}(x,d)\times\mathbf{P}^{prob}(x,d)}{\sum_{d=d_{\min}}^{d_{\max}}\mathbf{P}^{prior}(x,d)\times\mathbf{P}^{prob}(x,d)} \tag{8}$$

We refer to volume $\mathbf{P}^{post}$ as the posterior probability volume. Finally, the depth values are regressed using the expectation value along each depth direction on $\mathbf{P}^{post}$ as

$$\mathbf{D}^{post}(x) = \sum_{d=d^{min}}^{d^{max}} d\times\mathbf{P}^{post}(x,d) \tag{9}$$

where $d^{\max}$ and $d^{\min}$ are the maximum and minimum ranges of the depth hypotheses on $\mathbf{D}^{post}$, respectively.

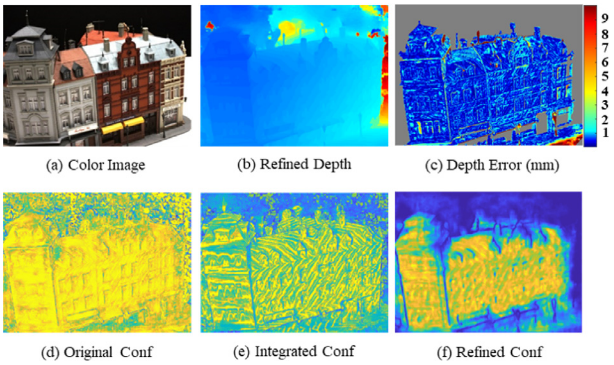
3.2.4. Integrated confidence estimation

The posterior probability volume $\mathbf{P}^{post}$ represents the probability distributions along each depth direction, which reflects the quality of the depth estimation. The confidence (or uncertainty) of an estimated depth is generally measured by the depth probability [10] or the variance [27] of the probability distribution. However, because the probability volume $\mathbf{P}_s^{post}$ at stage s in the cascade structure is constructed with restricted ranges of depth hypotheses, as shown in Fig. 4d, its probability distribution is not appropriate for confidence estimation. To address this issue, we provided a probabilistic method to calculate a representative confidence map from multiple probability volumes $\{\mathbf{P}_1^{post},\mathbf{P}_2^{post},\mathbf{P}_3^{post}\}$. For each stage s, we model the depth probability on $\mathbf{P}_s^{post}$ as a Laplacian distribution over the entire hypothesis range. The probability distributions at multiple stages are then integrated using a mixture model.

Given a probability volume $\mathbf{P}_s^{post}$ and estimated depth map $\mathbf{D}_s^{post}$ at the sth stage, the depth variance $\mathbf{V}_s(x)$ at pixel x and stage s is defined as [13]

$$\mathbf{V}_s(x) = \sum_{d=d_s^{min}}^{d_s^{max}} \mathbf{P}_s^{post}(x,d)\times|d-\mathbf{D}_s^{post}(x)| \tag{10}$$

where $d_s^{\max}$ and $d_s^{\min}$ are the maximum and minimum ranges of the depth hypotheses on $\mathbf{P}_s^{post}$, respectively. We then define the probability of depth d at pixel x and stage s using the Laplacian distribution [18]



(a) Color Image (b) Refined Depth (c) Depth Error (mm)
(d) Original Conf (e) Integrated Conf (f) Refined Conf

Fig. 4. An illustration of the confidence refinement for (a) the reference image. (b) The refined depth map $\mathbf{D}^{refine}$ and (c) the depth error between the refined depth and ground truth (gray color denotes none depth information). Refine-Net refines (e) the integrated confidence $\mathbf{C}^{post}$ and produces (f) the refined confidence map $\mathbf{C}^{refine}$. (d) The original confidence map is estimated from the cost distribution on probability volume at last stage as in [11]. The refined confidence represents the true depth error more accurately than the original and integrated confidences.

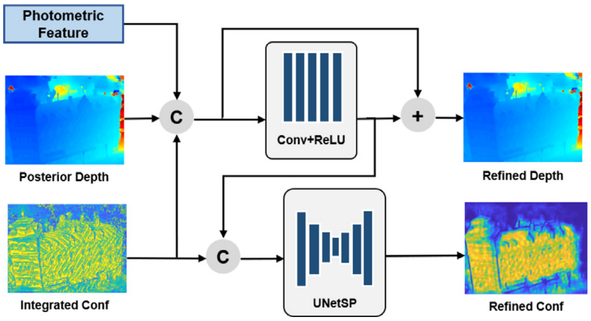


Fig. 5. The network architecture of Refine-Net. Refine-Net refines the posterior depth map $\mathbf{D}^{post}$ and confidence map $\mathbf{C}^{post}$ using the photometric features.

$$\text{p}\left(d\mid\mathbf{D}_s^{post}(x),\mathbf{V}_s(x)\right) = \frac{1}{2\mathbf{V}_s(x)}\exp\left(-\frac{|d-\mathbf{D}_s^{post}(x)|}{\mathbf{V}_s(x)}\right) \tag{11}$$

Finally, the integrated confidence $\mathbf{C}^{post}(x)$ of the estimated depth at pixel x is calculated using the mixture model

$$\mathbf{C}^{post}(x) = \sum_{s=1,2,3} w_s\text{p}\left(\mathbf{D}^*(x)\mid\mathbf{D}_s^{post}(x),\mathbf{V}_s(x)\right) \tag{12}$$

where $\mathbf{D}^*$ is the final depth map and is equal to the depth map $\mathbf{D}_3^{post}$ in the last stage. The mixture weights are set to equal values: $w_1=w_2=w_3=\frac{1}{3}$. Fig. 4e illustrates an example of the estimated confidence map. The integrated confidence map contains fewer outliers in the background and boundary regions compared to the initial confidence estimate (Fig. 4d).

3.3. Depth and confidence refinement

In the last stage, Refine-Net produces the final depth map $\mathbf{D}^{refine}$ and confidence map $\mathbf{C}^{refine}$ by refining the posterior depth map $\mathbf{D}^{post}$ and confidence map $\mathbf{C}^{post}$ (see Fig. 5). The integration of two probability volumes, $\mathbf{D}^{prior}$ and $\mathbf{D}^{prob}$ may produce discontinuous depths in a plane area. To alleviate these problems, our method uses the photometric features of the reference image as guidance information during depth map refinement. Furthermore, as shown in Fig. 4e, the confidence map estimated using the probability volume cannot accurately represent the true depth errors. Refine-Net directly learns depth estimation errors to produce a more accurate confidence map than the posterior confidence map.

3.3.1. Depth refinement

Refine-Net applies a 2D CNN to refine the posterior depth map $\mathbf{D}^{post}$. The 2D CNN includes several 2D convolutional layers followed by ReLU activations. This network takes the photometric feature $\mathbf{F}_{0,3}^{photo}$ of the reference image $\mathbf{I}_0$, depth map $\mathbf{D}^{post}$, and confidence map $\mathbf{C}^{post}$ as inputs to predict the residual depth map $\mathbf{D}^{res}$ [10]. The residual depth map is then added back into the input depth to generate the refined depth map $\mathbf{D}^{refine}$.

3.3.2. Refined confidence estimation

Refine-Net employs a 2D UNet architecture to predict the refined confidence map $\mathbf{C}^{refine}$. The network takes both the posterior confidence map $\mathbf{C}^{post}$ and the residual depth $\mathbf{D}^{res}$, computed from the depth refinement stage, as inputs. Similar to the network for prior confidence estimation (Section 3.1), the UNet architecture has Softplus activation at the final layer to output the noise variance $\tilde{\mathbf{C}}^{refine}$. Our method then transforms $\tilde{\mathbf{C}}^{refine}$ into the normalized confidences $\mathbf{C}^{refine}$ by applying the Gaussian weighting function as in Eq. (2). Fig. 4f shows the refined confidence map, which represents the true depth errors precisely.

3.3.3. Loss function

We obtain the depth loss $\mathcal{L}_{depth}^{s}$ by using L_1 loss for each stage s after the prior integration procedure.

$$\mathcal{L}_{depth}^{s} = \frac{1}{M}\sum_{x} |\mathbf{D}_{s}^{post}(x) - \mathbf{D}^{gt}(x)| \tag{13}$$

The prior loss $\mathcal{L}_{prior}$ for training Prior-Net is described in Section 3.1. Similar to the prior loss, we designed the loss for the refined depth and confidence maps by using L_1 aleatoric uncertainty loss

$$\mathcal{L}_{refine} = \frac{1}{M}\sum_{x}\left(\frac{|\mathbf{D}^{refine}(x) - \mathbf{D}^{gt}(x)|}{\tilde{\mathbf{C}}^{refine}(x)} + \log(\tilde{\mathbf{C}}^{refine}(x))\right) \tag{14}$$

In summary, the total loss is defined by the weighted sum over the losses above, as follows

$$\mathcal{L}_{total} = \sum_{s=1}^{3}\lambda_s \mathcal{L}_{depth}^{s} + \lambda_{prior}\mathcal{L}_{prior} + \lambda_{refine}\mathcal{L}_{refine} \tag{15}$$

where $\lambda_{s=1,2,3}$, λ_{prior}, and λ_{refine} are constant weights for each loss term, and in this study, we set the weights as $\lambda_{s=1,2,3} = \lambda_{prior} = \lambda_{refine} = 1.0$.

3.4. Efficiency improvement strategy

In this section, we introduce how to improve the efficiency of our network model. Since we focus on online 3D modeling, the efficiency of memory and computation time is also an important issue. However, the proposed framework (Fig. 1) sometimes requires a lot of memory and computation time because several complex networks are integrated.

Similar to the efficient methods [26,33,34], we can also apply a depth up-sample and refinement layer at Refine-Net to address this issue. MVS-Net quickly computes low-resolution depth and confidence maps (a quarter resolution) at three cascade stages instead of full-resolution maps. Refine-Net then up-samples the depth and confidence maps to the original high-resolution by performing a lightweight refinement using the RGB image. We used several 2D convolution layers to refine the depth and a 2D UNet with Softplus activation to output the refined confidence. This design can be used optionally according to a memory budget or runtime constraint.

4. Experiments on MVS benchmark

4.1. Implementation details

Training: The proposed network model was trained using the DTU dataset [19]. We directly used the ground-truth depths and view selection method provided by Yao et al. [10]. Similar to CasMVSNet [11], we set the image resolution to 640 × 512, the number of input views to 4, and the number of depth hypothesis planes to $\{64, 32, 8\}$ for training. The proposed MVS-Net was initialized from the pre-trained CasMVSNet model to increase the training speed. Prior-Net was also initialized by independently training the model (over 20 epochs with a batch size of 8) on the DTU training set without using the pre-trained model provided in the NCNN paper [17]. To train Prior-Net, the outputs of CasMVSNet were directly used as pre-computed depth and confidence maps of the input source images. After initializing both MVS-Net and Prior-Net, we trained the entire network end-to-end using the Adam optimizer with an initial learning rate of 0.0005.

Evaluation: The performance of the trained model was evaluated on the DTU evaluation set. The generalized performance of our network was also evaluated on the Tanks and Temples dataset [20] using the same trained model without fine-tuning. Similar to the model training, we used the depth and confidence maps computed by CasMVSNet as the inputs into Prior-Net. We set the number of input views to 7. We reconstructed the final point cloud by filtering the low-confidence depths based on confidence using a threshold of 0.5 and fusing all the depth maps [2]. We set the disparity threshold to 0.08, and the number of consistent views to 2.

4.2. Results for DTU dataset

We compared the performance of our method with conventional [2,3] and learning-based methods using the DTU dataset. We also evaluated the performance of the efficient version of our method (described in Section 3.4) denoted as **Ours-fast**. MVS-Net in Ours-fast estimates the depth maps at the resolution scales $\{\frac{1}{64}, \frac{1}{16}, \frac{1}{4}\}$ in the first three stages. It then up-sampled and refined the depth in Refine-Net to output a full-resolution depth in the last stage. We applied the identical setups with the original version described above for the training and evaluation process. Similar to our method, several learning-based methods [11–13,26,27] adopted the cascade structure for MVS depth estimation, which progressively navigates through depths in a coarse-to-fine manner by narrowing the depth hypothesis ranges. Some methods [26,27] also considered the visibility or attention information to perform depth estimation.

4.2.1. Evaluation on 3D modeling

Table 1 provides the quantitative results of the DTU evaluation dataset. We considered the three standard error metrics [19]: *accuracy, completeness*, and *overall* error. The learning-based methods that used the cascade structure [11–13] or visibility/attention information [26,27] generally outperformed the other learning-based methods [10,35,36]. Our method exhibited the best performance in terms of overall error. In particular, the overall error of our method at 0.319 had a significant margin over to the second-best performer at, 0.344. This indicates that an approach which considers prior depth information can remarkably improve the modeling performance.

Fig. 6 shows the qualitative comparison results of scans 13, 77, and 11. As can be seen in the figure, our method successfully produced clean and detailed reconstructions. Furthermore, our reconstruction provided larger areas of coverage of the target objects

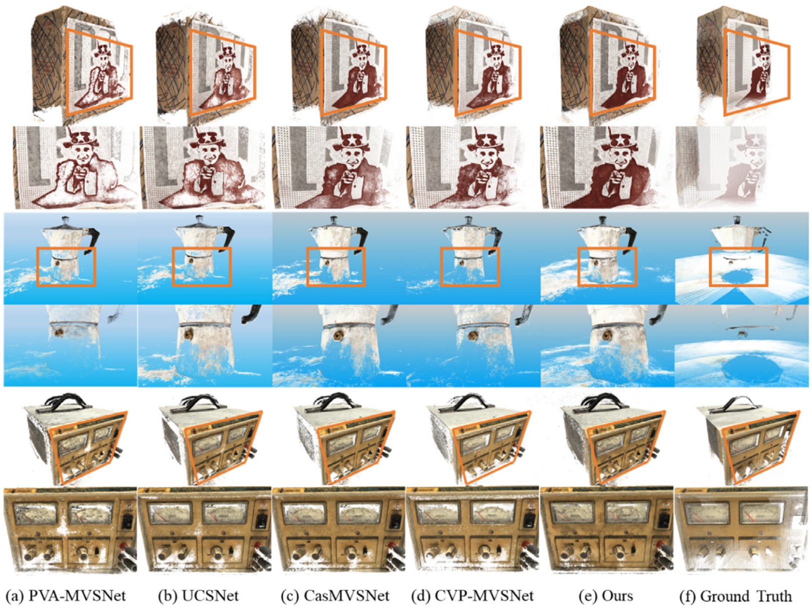

(a) PVA-MVSNet (b) UCSNet (c) CasMVSNet (d) CVP-MVSNet (e) Ours (f) Ground Truth

Fig. 6. Qualitative results of PVA-MVSNet [27], UCSNet [13], CasMVSNet [11], CVP-MVSNet [12], and our method on the DTU evaluation dataset.

Table 1
Quantitative results on the DTU evaluation dataset [19].

Methods	Mean error distance (mm)		
	Acc.	Comp.	Overall
Gipuma [2] ⋆	**0.283**	0.873	0.578
Colmap [3] ⋆†	0.400	0.664	0.532
MVSNet [10]	0.396	0.527	0.462
R-MVSNet [35]	0.385	0.459	0.422
P-MVSNet [36]	0.406	0.434	0.420
Fast-MVSNet [37]	0.336	0.403	0.370
Vis-MVSNet [26] ◦†	0.369	0.361	0.365
CasMVSNet [11] ◦	0.325	0.385	0.355
UCSNet [13] ◦	0.338	0.349	<u>0.344</u>
PVA-MVSNet [27] ◦†	0.379	<u>0.336</u>	0.357
CVP-MVSNet [12] ◦	<u>0.296</u>	0.406	0.351
PatchMatchNet [33] ◦†	0.429	**0.277**	0.352
IterMVS [34] ◦†	0.373	0.354	0.363
Ours ◦†	0.351	0.287	**0.319**
Ours-fast ◦†	0.350	0.305	0.327

⋆ conventional method, ◦ cascade structure, † visibility/attention-based method.

than the others, which significantly improved the reconstruction completeness.

4.2.2. Evaluation on depth and confidence maps

We also evaluated the quality of the estimated depths and corresponding confidences on the DTU evaluation dataset. The estimation quality of our method was compared with that of other learning-based methods [10,11,27,35]; all these methods produced confidence maps from the probability volume. To allow for a fair comparison, we used the pretrained models of the target methods with the same parameter settings, namely depth resolution (1152 × 864), input views (5), and the number of depth hypothesis planes (192).

To evaluate the performance of the depth maps, we measured two error metrics: *mean absolute error* (MAE) and *precision*. Precision is defined as the average percentage of depths for which the error is below a certain threshold. To measure the quality of the confidence maps, we used the *area under sparsification error plots* (AUSE) [38]. Two root-mean-square error (RMSE) curves for various depth densities were computed based on the filtered depths by changing the applicable thresholds for confidence and true depth error. The differences between these two RMSE curves were defined as the sparsification error plots, which represented the consistency between the predicted confidence and true errors. AUSE is a representative measure that quantifies the confidence quality.

Table 2 presents the quantitative results for the depth and confidence estimations. CasMVSNet performed the worst in terms of AUSE, but still showed comparable performances in precision and MAE. As mentioned in Section 3.2, CasMVSNet produces imprecise confidence maps because its cascade structure does not provide an integrated probability volume for the entire depth range. Conversely, because MVSNet and R-MVSNet formulate a single probability volume over an entire depth range, they showed better performances in terms of AUSE than the cascade-based methods, CasMVSNet and PVA-MVSNet. Our method achieved the best performance in terms of all the metrics. Even though our method also applied the cascade structure, it had the best AUSE of 0.283 with a significant margin over the second place, 0.335. Our method directly learns depth-estimation errors using the aleatoric uncertainty loss [18]. Therefore, the estimated confidences accurately represent the true depth errors.

Fig. 7 shows the qualitative comparisons of the tested methods. MVSNet and PVA-MVSNet produced discontinuous confidence maps with many outliers in the background and boundary regions. On the other hand, our method estimated smooth confidence maps and accurately identified the background regions. Our method also precisely captured the low-confidence areas at the boundary and occluded regions.

4.3. Results on tanks and temples dataset

In this section, we verify the generalization capability of our method by evaluating its performance on the Tanks and Temples benchmark, which contains diverse outdoor and indoor scenes. The details of the experimental results are described in Section 2 in the supplementary material. Tables 1 and 2 in the supplementary material show the quantitative results of our method and other state-of-the-art methods on the intermediate and advanced datasets. The tables show that our method achieved the best average F-score with comparable individual results in the intermediate and

Table 2
Quantitative results of depth and confidence maps on the DTU evaluation dataset.

	Depth Map			Confidence
Methods	Prec. 2mm (%)	Prec. 4mm (%)	MAE (mm)	AUSE
MVSNet [10]	70.30	79.68	16.69	0.335
R-MVSNet [35]	62.95	76.11	21.98	0.355
CasMVSNet [11]	75.16	80.26	12.53	0.483
PVA-MVSNet [27]	47.67	66.19	19.52	0.418
Ours	**77.80**	**82.82**	**9.68**	**0.283**

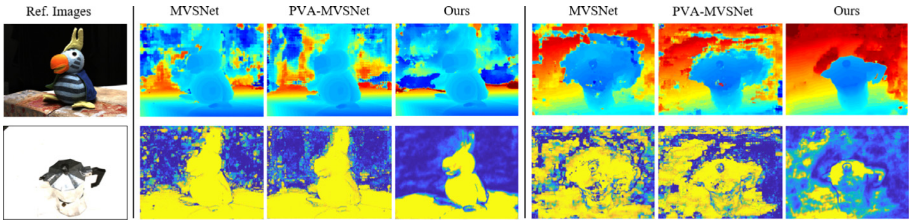


Fig. 7. Qualitative results of estimated depth map (top) and confidence map (bottom) on the DTU evaluation dataset (scan 4 and scan 77).

Table 3
Comparisons of our method with different model variants to evaluate the effectiveness of each module (visibility estimation **V**, prior depth integration **I**, depth refinement **R**).

Methods	Depth Map			Pointcloud			Time & Memory		
	Prec. 2mm	Prec. 4mm	MAE (mm)	Acc. (mm)	Comp. (mm)	Overall (mm)	# of Param	Time (s)	Mem (Mb)
Baseline	75.16	80.26	12.53	0.325	0.385	0.355	934K	0.368	5843
Model-A (V)	77.05	82.20	11.07	0.372	0.289	0.330	1428K	0.631	7030
Model-B (V+I)	77.79	82.81	9.68	0.371	0.275	0.323	1428K	0.634	7137
Model-C (V+I+R)	81.19	86.40	6.94	0.352	0.291	0.322	1554K	0.640	8397

advanced datasets. In particular, compared to CasMVSNet, which is the base approach of our method, we obtained better scores on all scenes. This demonstrates the effectiveness of our method in complex outdoor and indoor scenes.

4.4. Ablation study

We conducted an ablation study using the DTU evaluation dataset to verify the effectiveness of the key components of our method. We used the same settings as CasMVSNet [11], namely image resolution (1152 × 864), number of input views (5), and number of depth planes (192). We measured the time and memory consumption on a workstation with an Intel Xeon CPU (48 cores) with 252 GB of RAM and a Tesla V100 GPU with 32 GB of memory. We provided the performance gain of each key component in a progressive manner by comparing the four model variants as follows:

- **Baseline**: CasMVSNet [11] was directly adopted as the baseline method.
- **Model-A**: Only the visibility-aware cost-volume formulation, described in Section 3.2, was applied to the baseline model.
- **Model-B**: The prior depth integration step, described in Section 3.2, was applied to Model-A, which is the same as the full model of MVS-Net.
- **Model-C**: This is our full model, including Refine-Net in Section 3.3.

Table 3 presents the performances of the variant models with respect to depth estimation, point cloud reconstruction, and time complexity. Model-A showed better performance in terms of depth estimation and point cloud reconstruction than the baseline method. The visibility-based cost-volume formulation achieved a significant performance improvement over the original cost-volume formulation [11]. Model-B performed better than Model-A for all the criteria. Our method probabilistically integrated noise-suppressed prior depths into the current cost volume. This approach enhanced the accuracy of the estimated depths by reducing the possibility of error propagation as much as possible. Model-C applied strict outlier filtering based on predicted confidences; therefore, Model-C had lower model completeness than Model-B but showed a higher reconstruction accuracy. In particular, Model-C showed much better performance in terms of depth estimation than Model-B. This indicates that Refine-Net could effectively improve the quality of the estimated depths.

Each computation time on Table 3 represents the average time to process a single frame. Compared to CasMVSNet, Model-C takes more computation time of 0.272 s longer to process a single frame. This study focuses on accurate 3D modeling instead of fast depth computation; therefore, Model-C sacrifices some computation time overhead to improve the modeling performance. However, this time gap does not cause a bottleneck of the whole system in that the keyframe extraction period in the online modeling system is much longer than the computation time of Model-C. Both CasMVSNet and Model-C can compute a depth map of the current keyframe before extracting the next keyframe, so online modeling is possible for both methods without any bottleneck. In addition, the performance gain of Model-C is significant.

In the supplementary material, we also provide additional ablation studies, including (i) accuracy of prior depth map, (ii) efficiency for different image resolutions, and (iii) influence of depth plane number. Please refer to Section 3 in the supplementary material for detailed results.

5. Experiments on online 3D modeling

To verify the performance of our method during online 3D modeling, we conducted comparative experiments using motion stereo methods. We evaluated the modeling performance on large-

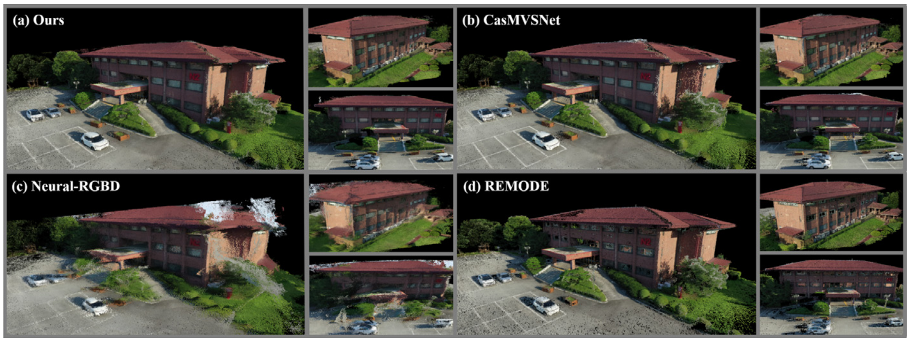


Fig. 8. Qualitative comparison of the reconstructed 3D models obtained from (a) our method, (b) CasMVSNet [11], (c) Neural-RGBD [16], and (d) REMODE [8] for Scenario 1.

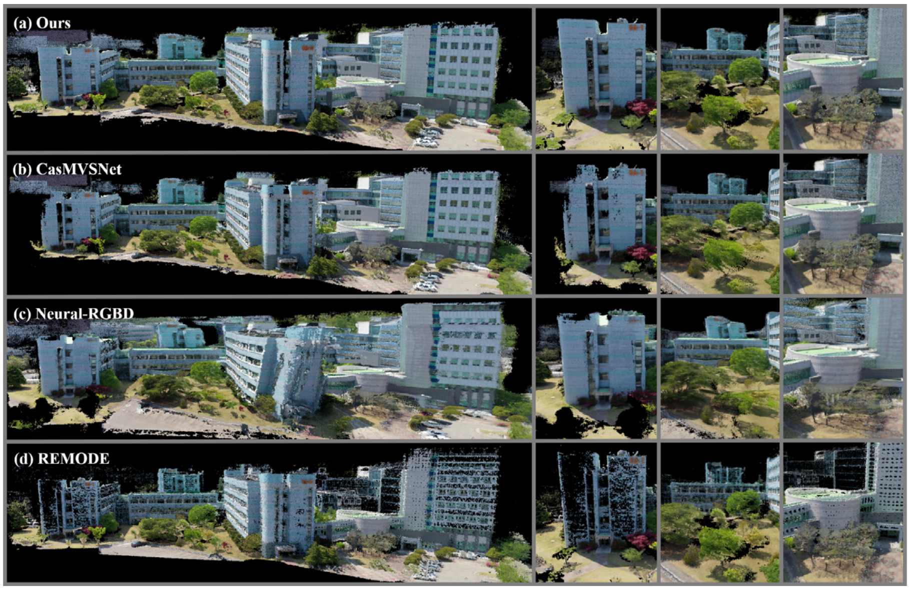


Fig. 9. Qualitative comparison of the reconstructed 3D models obtained from (a) our method, (b) CasMVSNet [11], (c) Neural-RGBD [16], and (d) REMODE [8] for Scenario 2.

scale structures in outdoor environments. We considered two scenarios: modeling a single structure (Scenario 1; Fig. 8) and modeling multiple structures (Scenario 2; Fig. 9). For each scenario, we acquired sequential images of aerial scenes using a monocular camera mounted on a micro aerial vehicle (MAV) and used them to perform online 3D modeling.

Our method was compared with two state-of-the-art motion stereo methods:

- **REMODE** [8]: This is a classic motion stereo method that uses handcrafted features for stereo matching. For each input source image, it updates the mean depth, variance, and inlier probability using a recursive Bayesian estimation approach. It removes outlier depths based on the estimated variance and inlier probability instead of the consistency check. Given a reference frame, it uses 10 neighboring frames as source images.
- **Neural-RGBD** [16]: This is a deep learning-based method for motion stereo. It sequentially propagates the depth probability distribution by integrating two consecutive cost volumes under a Bayesian filtering framework. It uses five source images. We used the provided pretrained model with the same parameter settings.

We also evaluated the performance of CasMVSNet [11], considering it as a baseline MVS method. We used the pretrained models with the same settings.

We applied each depth estimation method to the online 3D modeling system proposed in [14]. The implementation details for the online modeling system are presented in the supplementary material (Section 1). The system acquired image frames with a resolution of 1200 × 900 and estimated their poses using ORB-SLAM [15]. To estimate the camera poses accurately, we acquired stable images by using a gimbal camera stabilizer and by restricting the motion speed of the MAV to be small. Every method constructed a 3D model from the same pose estimation results of SLAM.

Table 4
Averaged computation time and GPU memory consumption of each MVS depth estimation method under the online 3D modeling scenarios.

Methods	Ours	CasMVSNet	Neural-RGBD	REMODE
Time (s)	0.778	0.468	2.353	0.792
Mem (Mb)	8,861	6,273	13,609	299

Table 4 tabulates the computation time and GPU memory consumption of each method. Note that all methods except Neural-RGBD allow online reconstruction of high-resolution 3D models. Since the SLAM extracts a keyframe every 1.3 s on average, there is no bottleneck caused by depth estimation for our method. Neural-RGBD requires the largest computation time and GPU memory since it constructs a single large cost volume, unlike the cascade cost volume construction. Therefore, Neural-RGBD is not appropriate for online depth estimation of high-resolution images. REMODE uses the least GPU memory as it is a classical method that does not apply a deep learning model.

The qualitative results of the reconstructed models for each scenario are shown in Figs. 8 and 9. REMODE generated relatively sparse models compared to other methods. It performed stereo matching based on handcrafted features, which generally produce sparse point clouds for untextured or specularly reflected surfaces. Neural-RGBD generated the most inaccurate reconstructed models with many outliers. When it propagated the depth probability distribution, incorrect predictions were continuously propagated to consecutive frames. This caused many outliers and inaccurate reconstructions.

Our method showed better qualitative performance than CasMVSNet. Our method completely reconstructed the entire surface of multiple structures with fewer outliers. It used sequentially estimated depth maps as prior information, which improved the completeness of the reconstructed models. To compute the prior information, our method checked the consistency of multiple depths on the source images, which significantly reduced the possibility of error propagation. Furthermore, our confidence estimates accurately represented the true depth errors; therefore, our method could filter outliers more precisely than CasMVSNet. These results demonstrate the feasibility of the proposed MVS method in an online 3D modeling system for outdoor scenes.

6. Limitations and discussion

Although our method could achieve outstanding results on benchmark datasets [19,20] and real-world aerial scenes, it has several major limitations. First, the depth maps of the source images should be computed first for prior depth estimation to take place. Our method determines the source images from insufficient candidates that are restricted to the previously processed frames. Therefore, it is difficult to obtain a set of source images that sufficiently cover the entire area of the reference image plane. At times, this can degrade the completeness of the estimated depth maps.

Second, our method concentrates on precise reconstruction instead of fast depth computation; therefore, it requires more computation time than CasMVSNet. When scenes change rapidly because of dynamic or fast camera motions, motion stereo methods that can compute depth maps quickly may be more effective. To speed up the computation time of our method, we also provide an efficiency improvement strategy in Section 3.4. This strategy computes a low-resolution depth map quickly and then upsamples the depth map to the original high resolution by using a lightweight refinement filter. This approach significantly improves the efficiency of runtime and memory consumption while it sacrifices a small amount of performance. Several studies [33,34] proposed an efficient MVS network model that did not use 3D CNNs for cost-volume regularization. Their models require only about 0.25 s to process a single frame at a one-megapixel resolution. The application of the backbone of these models [33,34] to our method would be a good direction for future work.

Third, the proposed method was developed based on the assumption that the camera poses estimated by the SLAM module are accurate. Therefore, the reconstruction quality of the online modeling system is strongly affected by localization errors. When a high localization error occurs, our method may produce an inaccurate 3D model which would contain multiple inconsistent surfaces. To address this issue, we intend to apply dense bundle adjustment [39] to our method in future work. It solves the dense SfM problem by optimizing the depth maps and camera poses simultaneously.

7. Conclusion

We present a novel network framework for online MVS reconstruction that effectively integrates sequentially estimated depth information. The framework first predicts a prior depth map by fusing the reliable depths of the source views. It then generates the cost volume based on the pixel-wise visibility information. The framework integrates the prior depth into the cost volume probabilistically. This approach improves the stereo-matching performance and completeness of estimated depth maps. Furthermore, the framework predicts the confidences of the estimated depths, which accurately represents the true depth errors. The predicted confidences are used to filter a large number of outliers. The experimental results on the MVS benchmarks show that the proposed method outperformed other state-of-the-art methods, especially in terms of the completeness of 3D model. The results for the scenarios using aerial scenes demonstrated that our method could reconstruct precise models even online.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education (NRF2016R1D1A1B01013573).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2022.109198.

References

[1] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, IEEE Trans. Pattern Anal. Mach. Intell. (2009).
[2] S. Galliani, K. Lasinger, K. Schindler, Massively parallel multiview stereopsis by surface normal diffusion, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 873–881.
[3] J.L. Schönberger, E. Zheng, J.-M. Frahm, M. Pollefeys, Pixelwise view selection for unstructured multi-view stereo, in: European Conference on Computer Vision, Springer, 2016, pp. 501–518.
[4] M. Grum, A.G. Bors, 3D modeling of multiple-object scenes from sets of images, Pattern Recognit. 47 (1) (2014) 326–343.
[5] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion: real-time dense slam and light source estimation, Int. J. Rob. Res. 35 (14) (2016) 1697–1716.
[6] Y. Fan, Q. Zhang, Y. Tang, S. Liu, H. Han, Blitz-SLAM: a semantic slam in dynamic environments, Pattern Recognit. 121 (2022) 108225.
[7] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, DTAM: dense tracking and mapping in real-time, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2320–2327.
[8] M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: probabilistic, monocular dense reconstruction in real time, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 2609–2616.
[9] Y. Hou, J. Kannala, A. Solin, Multi-view stereo by temporal nonparametric fusion, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2651–2660.
[10] Y. Yao, Z. Luo, S. Li, T. Fang, L. Quan, MVSNet: depth inference for unstructured multi-view stereo, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 767–783.
[11] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, P. Tan, Cascade cost volume for high-resolution multi-view stereo and stereo matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2495–2504.
[12] J. Yang, W. Mao, J.M. Alvarez, M. Liu, Cost volume pyramid based depth inference for multi-view stereo, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[13] S. Cheng, Z. Xu, S. Zhu, Z. Li, L.E. Li, R. Ramamoorthi, H. Su, Deep stereo using adaptive thin volume representation with uncertainty awareness, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2524–2534.
[14] S. Song, D. Kim, S. Choi, View path planning via online multiview stereo for 3-d modeling of large-scale structures, IEEE Trans. Rob. (2021).
[15] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Rob. 31 (5) (2015) 1147–1163.

[16] C. Liu, J. Gu, K. Kim, S.G. Narasimhan, J. Kautz, Neural RGB (r) D sensing: depth and uncertainty from a video camera, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10986–10995.

[17] A. Eldesokey, M. Felsberg, F.S. Khan, Confidence propagation through CNNs for guided sparse depth regression, IEEE Trans. Pattern Anal. Mach. Intell. 42 (10) (2019) 2423–2436.

[18] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, arXiv preprint arXiv:1703.04977 (2017).

[19] H. Aanæs, R.R. Jensen, G. Vogiatzis, E. Tola, A.B. Dahl, Large-scale data for multiple-view stereopsis, Int. J. Comput. Vis. 120 (2) (2016) 153–168.

[20] A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: benchmarking large-scale scene reconstruction, ACM Trans. Graph. (ToG) 36 (4) (2017) 1–13.

[21] S. Prakash, A. Robles-Kelly, A semi-supervised approach to space carving, Pattern Recognit. 43 (2) (2010) 506–518.

[22] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, K. Schindler, Learned multi-patch similarity, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1586–1594.

[23] A. Kar, C. Häne, J. Malik, Learning a multi-view stereo machine, arXiv preprint arXiv:1708.05375 (2017).

[24] M. Ji, J. Gall, H. Zheng, Y. Liu, L. Fang, SurfaceNet: an end-to-end 3D neural network for multiview stereopsis, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2307–2315.

[25] X. Yang, Y. Gao, H. Luo, C. Liao, K.-T. Cheng, Bayesian DeNet: monocular depth prediction and frame-wise fusion with synchronized uncertainty, IEEE Trans. Multimedia 21 (11) (2019) 2701–2713.

[26] J. Zhang, Y. Yao, S. Li, Z. Luo, T. Fang, Visibility-aware multi-view stereo network, arXiv preprint arXiv:2008.07928 (2020).

[27] H. Yi, Z. Wei, M. Ding, R. Zhang, Y. Chen, G. Wang, Y.-W. Tai, Pyramid multi--view stereo net with self-adaptive view aggregation, in: European Conference on Computer Vision, Springer, 2020, pp. 766–782.

[28] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[30] C. Wang, X. Wang, J. Zhang, L. Zhang, X. Bai, X. Ning, J. Zhou, E. Hancock, Uncertainty estimation for stereo matching based on evidential deep learning, Pattern Recognit. (2021) 108498.

[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[32] Y. Zhang, Y. Chen, X. Bai, S. Yu, K. Yu, Z. Li, K. Yang, Adaptive unimodal cost volume filtering for deep stereo matching, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 12926–12934.

[33] F. Wang, S. Galliani, C. Vogel, P. Speciale, M. Pollefeys, PatchmatchNet: learned multi-view patchmatch stereo, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14194–14203.

[34] F. Wang, S. Galliani, C. Vogel, M. Pollefeys, IterMVS: iterative probability estimation for efficient multi-view stereo, arXiv preprint arXiv:2112.05126 (2021b).

[35] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, L. Quan, Recurrent MVSNet for high-resolution multi-view stereo depth inference, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5525–5534.

[36] K. Luo, T. Guan, L. Ju, H. Huang, Y. Luo, P-MVSNet: learning patch-wise matching confidence aggregation for multi-view stereo, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10452–10461.

[37] Z. Yu, S. Gao, Fast-MVSNet: sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1949–1958.

[38] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, T. Brox, Uncertainty estimates and multi-hypotheses networks for optical flow, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 652–667.

[39] C. Tang, P. Tan, Ba-net: Dense bundle adjustment network, arXiv preprint arXiv:1806.04807 (2018).

Soohwan Song received the BS degree in information and communication engineering from Dongguk University, Seoul, South Korea, and the MS and PhD degrees in computer science from KAIST, Daejeon, South Korea, in 2013, 2015, and 2020, respectively. After graduation, he worked as a Postdoctoral Researcher with Neuro-Machine Augmented Intelligence Laboratory, KAIST. Since March in 2021, he has been with the Intelligent Robotics Research Division, ETRI, as a Senior Researcher. His research interests include robotics, motion planning, and computer vision.

Khang Truong Giang received the MS degree in computer science at KAIST, Daejeon, Republic of Korea, in 2021, where he is currently pursuing the PhD degree with the School of Computing. His research interests are machine learning, deep learning, and computer vision.

Daekyum Kim received the BS degree in mechanical engineering from the University of California, Los Angeles, Los Angeles, CA, USA, in 2015, and received the PhD degree in Computer Science at KAIST, Daejeon, Republic of Korea, in 2021. He is currently a Postdoctoral Research Fellow at the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. His research interests are in the areas of machine learning, computer vision, and robotics.

Sungho Jo received the BS degree in the school of mechanical & aerospace engineering from the Seoul National University, Seoul, Republic of Korea, the SM degree in mechanical engineering, and the PhD degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1999, 2001, and 2006 respectively. While pursuing the PhD, he was associated with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Laboratory for Information Decision and Systems (LIDS), and Harvard-MIT HST Neuro-Engineering Collaborative. Before joining the faculty with KAIST, he worked as a postdoctoral researcher with MIT media laboratory. Since December in 2007, he has been with the School of Computing, KAIST, where he is currently professor. His research interests include intelligent robots, neural interfacing computing, and wearable computing.