Sequential Depth Completion With Confidence Estimation for 3D Model Reconstruction

Khang Truong Giang, Soohwan Song, Daekyum Kim, and Sunghee Choi

***Abstract*—This letter addresses a depth-completion problem for sequential data to reconstruct 3D models of outdoor scenes. While many deep-learning-based approaches have recently achieved promising results, their results are not directly applicable to 3D modeling because of several reasons. First, most results contain a lot of outliers because of irregularly distributed sparse measurements. Second, they ignore temporal coherence in sequential frames and produce temporally inconsistent depths. Therefore, we propose a new method that predicts temporally consistent depths with corresponding confidences from sequential frames. The suggested method can efficiently remove the outliers based on confidence estimates, which accurately represent the true prediction errors. The method also produces temporally consistent depths by integrating the depth information of consecutive frames. In addition, we present a 3D-modeling system that reconstructs a globally consistent 3D model in real-time using the results from the proposed depth completion method. Extensive experiments on synthetic and real-world datasets show that our method outperforms the other state-of-the-art methods in terms of both depth-completion and 3D-modeling accuracies.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Computer Vision for Transportation.**

I. Introduction

IT IS important to estimate accurate and dense depth maps for outdoor scenes in tasks related to computer vision and robotics including 3D environmental perception and autonomous navigation. As a primary means to achieve dense depth maps, depth-completion methods are widely applied. Depth completion aims to compute a dense depth map from sparse depth measurements that may come from LiDAR data or SLAM map points by leveraging high-resolution image information. To solve the depth-completion problem, conventional approaches have used filter-based methods [1], [2] or optimization-based methods [3], [4]. Recently, deep-learning-based methods [5]–[8] have emerged as a potential technique to solve depth-completion problems.

Manuscript received July 24, 2020; accepted November 17, 2020. Date of publication December 8, 2020; date of current version December 21, 2020. This letter was recommended for publication by Associate Editor L. Liu and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea funded by the Ministry of Education under Grant 2016R1D1A1B01013573 and in part by the Technology Innovation Program funded by the Ministry of Trade, Industry & Energy (MI, Korea) under Grant 10070171. *(Khang Truong Giang and Soohwan Song contributed equally to this work.) (Corresponding author: Sunghee Choi.)*

The authors are with the School of Computing, KAIST, Daejeon 34141, South Korea (e-mail: khangtg@kaist.ac.kr; dramanet30@kaist.ac.kr; daekyum@kaist.ac.kr; sunghee@kaist.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2020.3043172, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3043172

In this study, we investigate the possibility of reconstructing 3D models using a depth completion method. Depth-completion methods have their merits in that they enable to generate a more detailed 3D model than the 3D model based on the simple accumulation of LiDAR data or SLAM map points. Furthermore, given that the sparse depth measurements are accurate, it is possible to produce a more accurate 3D model than that produced by existing image-based reconstruction approaches, like multi-view stereo [9] or motion stereo [10] methods.

However, there are some factors that degrade the reconstruction performances of depth-completion methods. First, unreliable depth maps may be produced because the depth measurements are very sparse and irregularly distributed in the image space. In this reason, a more accurate depth-estimation approach is essential to obtain precise 3D models. Second, the estimated depth maps contain outliers in backgrounds and some occluded regions; these outliers should be filtered out. Third, depth maps are computed independently for each frame. This in result does not consider temporal coherence of consecutive frames. Therefore, depth-completion methods can generate depths that are temporally inconsistent, thus producing locally inconsistent 3D models.

Therefore, we propose a new deep-learning framework that predicts temporally consistent depths with their confidences from sequential frames to obtain accurate depth maps and 3D models. The proposed framework produces the confidence maps from comprehensive latent features that are trained directly from true depth errors. The predicted confidences provide precise information about reliability of the depth values. The confidences are used to filter out outlier depths like background or occluded areas as well. The framework also sequentially propagates the estimated depths from frame to frame to keep depths temporally consistent. Instead of directly propagating the depth maps [11], [12], we integrate the latent features of two consecutive frames, to effectively incorporate the comprehensive information of the features. This is done by proposing a confidence-based feature integration to apply different attentions to each consecutive feature. In addition, we present an online 3D-modeling system based on our depth-completion method. The modeling system generates a globally consistent 3D model in real-time, fusing the estimated depth maps with their confidences. Extensive experiments show that our method outperforms the other state-of-the-art methods in terms of depth-completion accuracy. We further demonstrate online modeling of a real-world scene as shown in Fig. 1.

The contributions of this work are summarized as follows:

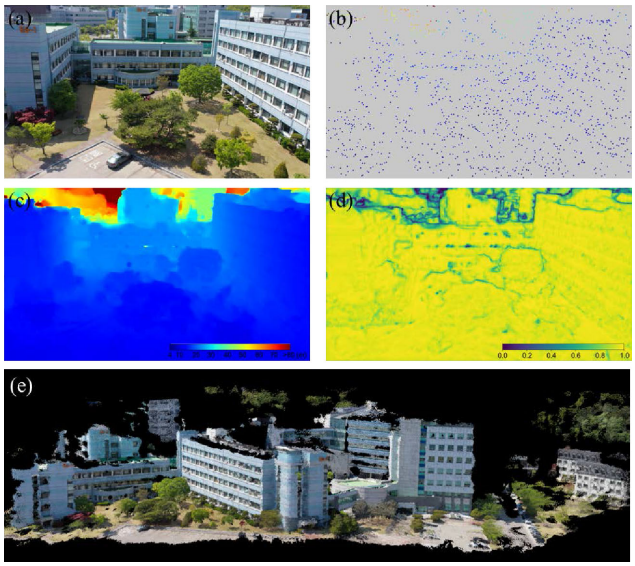


Fig. 1. 3D reconstruction result of a real-world scene based on the proposed depth-completion method. Our method takes (a) an image and (b) a sparse depth map as inputs and predicts (c) depth map and (d) corresponding confidence. The predicted depths and confidences are integrated into (e) a 3D model using the proposed online modeling system.

- We present a novel depth-completion framework that performs sequential depth fusion and confidence prediction. It can predict temporally consistent depths and filter out the outliers based on the confidence estimates.
- We propose a method to predict confidences of depth completion, which are trained end-to-end based on depth-estimation error. The predicted confidences precisely represent the true depth errors.
- We propose a novel online 3D-modeling system based on depth completion. This system applies strict noise filtering and constructs a globally consistent model using surfel-based mapping [13].
- We empirically evaluate the proposed method using the KITTI benchmark [5] and Aerial dataset [8]. We also provide results about 3D reconstruction. Our source code and demonstration video are publicly available.[1]

The remainder of this paper is structured as follows. Section II shows the related work on depth completion. Sections III and IV describe the proposed depth-completion method and online modeling system, respectively. Section V shows the experimental results, and Section VI summarizes the contributions made by this study.

II. Related Work

Depth Completion: Most conventional approaches address the depth-completion problem with bilateral filtering [1] or global optimization [3], [4] based on the underlying assumption that depth discontinuities coincide with color changes. To further enhance the depth-completion accuracy, several approaches consider additional information, such as semantic segmentation [15] or planarity measures [16].

[1]https://github.com/TruongKhang/depth-completion-seq-net

Recently, many studies [5], [6], [7], [17] have used convolutional neural networks for depth completion, which have achieved performance improvements over the conventional methods. Uhrig *et al.* [5] proposed sparsity invariant convolution, which evaluates the validity mask of sparse input to handle sparse irregular inputs in depth completion. Ma and Karaman [6] proposed an end-to-end deep-regression model, which is composed of an encoder-decoder deep network based on ResNet. Li *et al.* [17] suggested a cascade hourglass architecture to deal with multi-scale structures for depth completion. They progressively predicted depth maps in a coarse-to-fine manner with multi-scale guidance. Qiu *et al.* [7] additionally predicted surface normal maps and used them to regularize the depth completion and to filter noises.

Confidence Estimation: Some studies attempted to predict confidences of depth estimates to filter out noisy predictions. Diverse confidence estimation approaches have been proposed for different depth-estimation tasks, such as depth prediction [18], stereo vision [19], and multi-view stereo [20]. Conversely, confidence estimation for depth completion has received relatively less attention; there have been only a few confidence-estimation methods for depth completion.

Some depth-completion methods [7], [21], [14] estimated the confidences for internal processing without providing any confidence output for the final prediction. Qiu *et al.* [7] and Van Gansbeke *et al.* [21] used multiple inertial modules that predict depths in parallel, with corresponding confidences, and then applied attention-based integration using the predicted confidences. Eldesokey *et al.* [14] proposed the normalized convolutional network (NConv) that estimates confidences from convolution operations and propagates them to consecutive layers. Several methods [22], [23] produced confidence output along with the final depth estimate by directly training error-maps [22] or by reformulating the NConv framework probabilistically [23]. However, these methods [22], [23] solved unguided depth completion that ignores RGB guidance information. Teixeira *et al.* [8] proposed an image-guided confidence estimation method for guided depth completion. They extended NConv by appending an extra confidence-estimation network.

Sequential Depth Integration: Integrating sequential depth estimates could provide temporally consistent results with improved accuracy. Depth integration approaches have popularly been used in conventional depth-estimation approaches, such as monocular SLAM [24] and motion stereo [10], [25]. They estimate depth and uncertainty measurements in a video sequence and propagate them frame to frame based on multiple-depth-hypothesis filtering and fusion.

Recently, several deep-learning-based methods [11], [12] have been introduced to capture temporal coherence in sequential frames. Hou *et al.* [11] proposed motion stereo method that integrates sequential depth information using a Gaussian process model. Yang *et al.* [12] presented a video-based depth-prediction method that fuses sequential depth and uncertainty estimates in a Bayesian inference framework.

Although these methods were presented for motion stereo [11] or depth prediction [12], there are currently no appropriate depth-integration methods for depth completion. The previous

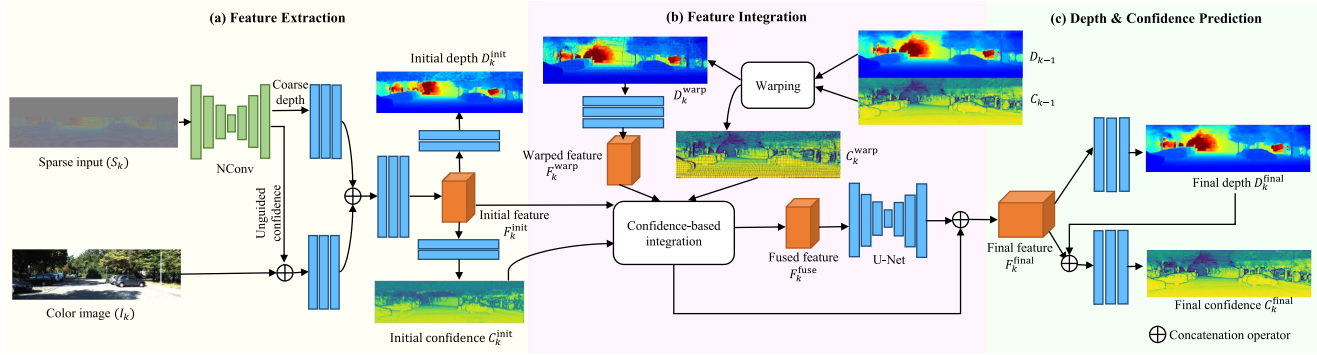


Fig. 2. Overview of the proposed network framework. Given input color image I_k and sparse depth measurement S_k of frame k; our method first extracts an initial latent feature F_k^{init} along with a confidence map C_k^{init} based on an NConv-based architecture [14] (Section III.A). Then, the initial feature is updated by fusing it with the depth feature F_k^{warp} of the previous frame, $k-1$. The fusion method uses an attention-based integration with respective confidence maps C_k^{init} and C_k^{warp} (Section III.B). The fused feature F_k^{fuse} is refined by U-Net architecture and then concatenated with F_k^{init} to produce a final feature F_k^{final}. Finally, the final feature is fed into a two-stream convolutional neural network to predict the final depth D_k and confidence C_k (Section III.C). These final outputs can be used for the depth completion of the subsequent frame.

methods continuously propagated depth hypotheses based on the assumption of a static scene. However, they are not suitable for dynamic scenes, like the KITTI dataset. Furthermore, incorrect predictions might also be continuously accumulated over multiple frames. To address these issues, instead of propagating depth hypotheses, we independently predict depth and confidence based on the integrated latent features of two consecutive frames. The latent features of consecutive frames are fused by attention-based integration and are regularized by a U-Net architecture.

III. Depth Completion Method

We assume that a data frame including an RGB image $I_k : \mathbb{R}^2 \supset \Omega \mapsto \mathbb{R}^3_+$, a sparse depth map $S_k : \Omega \supset \Omega_S \mapsto \mathbb{R}_+$, and a camera pose $T_{w,k} \in \mathbb{SE}(3)$ are acquired as input; where camera pose $T_{w,k}$ represents the pose of frame k with respect to the world coordinates w. For each input frame k, we estimate the dense depth map $D_k : \Omega \mapsto \mathbb{R}_+$ and the confidence map $C_k : \Omega \mapsto \mathbb{R}_+$ by integrating the information about the previously estimated depth map D_{k-1} and confidence map C_{k-1}. Fig. 2 depicts the proposed network architecture for depth completion and confidence estimation, which is composed of three stages: **feature extraction** (Section III.A), **feature integration** (Section III.B), and **depth and confidence prediction** (Section III.C). The feature-extraction stage computes a latent feature for each input frame. This stage also estimates the confidences of the latent features. The feature-integration stage updates the extracted latent feature by combining it with the corresponding feature of the previously estimated depth map. The integrated feature is then fed into a U-Net-like encoder-decoder architecture for feature refinement. The last stage concatenates the integrated feature with the initial feature and estimates the final depth map and the corresponding confidence map.

A. Feature Extraction

In the first stage, our model takes an image I_k and a sparse depth map S_k as inputs and produces a latent feature and the corresponding confidence for depth completion. The latent feature implicitly represents the dense depth information based on the input data. We use the feature-extraction layers in an NConv framework [14]. NConv can efficiently process highly sparse depth data with a small number of parameters using a multi-stream fusion strategy with a normalized convolution operation. Furthermore, NConv has one of the best performances to date, even though it requires less computation time and GPU memory than other state-of-the-art methods [6], [21]; therefore, NConv was adopted for our feature-extraction model.

The feature-extraction stage is based on a multi-stream framework with late fusion [14], which extracts multiple features from the image and from the sparse depth map separately and fuses them for performing depth completion. Fig. 2(a) depicts the feature-extraction pipeline, which consists of two pathways: the depth pathway and image pathway. The depth pathway considers only sparse depth S_k without color information. This pathway performs unguided depth completion from S_k and predicts a coarse dense depth with unguided confidence. The unguided confidence is computed by iteratively propagating the confidence starting from a sparse depth mask to consecutive normalized convolutional layers [26]. The depth pathway then extracts the latent features for the coarse dense depth. The image pathway concatenates the computed unguided confidence and the image I_k and then extracts their latent features. The extracted features from the two pathways are combined and fed into multiple convolutional layers with ReLU activations to produce the initial feature. It encodes the local affinity from the color image and the local depth information from the sparse input.

In contrast to NConv, we additionally compute a guided confidence for the extracted initial feature. This confidence is used for integrating sequential information. The network for guided confidence estimation is simply composed of two residual blocks [27] and one 1×1 convolution layer, followed by a sigmoid activation, to obtain output in the range $[0, 1]$. We denote the computed initial feature and the confidence map for a frame k, as $F_k^{init} \in \mathbb{R}^{h \times w \times c}$, respectively; where h and w are the dimensions of the input, and $c = 32$ is the number of feature channels used in our experiments.

B. Feature Integration

In this stage, our model refines the initial feature F_k^{init} by integrating the estimated depth information from the previous frame. Integrating sequential depth information could provide complementary information and produce temporally consistent results with improved accuracy. Given a previous depth map D_{k-1} and the corresponding confidence map C_{k-1}, we first warp them into the current frame k using a relative camera pose $T_{k,k-1}$ from frame $k-1$ to frame k. The relative camera pose can be calculated as $T_{k,k-1} = T_{w,k}(T_{w,k-1})^{-1}$. The warped depth map and warped confidence map are denoted as D_k^{warp} and C_k^{warp}, respectively. We then extract the feature F_k^{warp} of D_k^{warp} from a network, which is composed of a single convolution with ReLU and two residual blocks. The features are represented with 32 channels and the same resolution as the input image I_k.

The extracted two features, F_k^{init} and F_k^{warp}, are fused by the attention-based integration method [7]. Instead of directly integrating the depth maps [7], [12], we integrate the latent features. The initial feature F_k^{init} and the warped feature F_k^{warp} are fused by the following integration function:

$$F_k^{fuse} = \frac{C_k^{init} F_k^{init} + C_k^{warp} F_k^{warp}}{C_k^{init} + C_k^{warp}},$$

where initial confidence C_k^{init} and warped confidence C_k^{warp} are already normalized to the range $[0, 1]$; therefore, the weighted sum can be used for the integration function.

The integrated feature contains some abruptly discontinued feature values because the warping operation produces zero-depth-pixels in specific image locations. Therefore, we refine and smooth the integrated feature F_k^{fuse} using a U-Net-like encoder-decoder architecture [28]. The encoder consists of three down-sampling blocks, where each block includes two convolutional-ReLU layers with a max-pooling layer to down-size the feature to half resolution. The decoder consists of three upsampling blocks, in which the upsampling operators are executed by an interpolation method to produce the smoothing results. The final feature, F_k^{final}, is then determined by concatenating the refinement out of the integrated feature F_k^{fuse} with the initial feature F_k^{init}. The final feature encodes the depth hypotheses from multiple frames, while preserving the local information about the color image and sparse measurements.

C. Depth and Confidence Prediction

The last stage predicts the final depth map and the corresponding confidence map using the estimated final feature F_k^{final}. Our model, shown in Fig. 2(c), follows a two-stream network that predicts a depth map and a confidence map separately. One stream for depth prediction takes the final feature F_k^{final} as input, and the other stream for confidence estimation takes both F_k^{final} and the final depth output D_k^{final} as inputs. The two streams follow a similar format: one 3×3 convolution-ReLU block, followed by two residual blocks, and a final 1×1 convolution with a sigmoid function.

Loss function: The loss function of our network model is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_d(D_k^{final}) + \lambda_2 \mathcal{L}_c(C_k^{final}) + \lambda_3 \mathcal{L}_d(D_k^{init}) + \lambda_4 \mathcal{L}_c(C_k^{init}),$$

where $\mathcal{L}_d$ and $\mathcal{L}_c$ represent the losses on the estimated depth map and confidence map, respectively. The initial and final depths with confidences are used for loss computation. $\lambda_{1\ldots4}$ are constant weights for each loss term; in this study, we set the weights as $\lambda_1 = \lambda_2 = \lambda_4 = 1.0$ and $\lambda_3 = 0.3$.

The depth loss, $\mathcal{L}_d$, is defined as the pixel-wise mean squared error:

$$\mathcal{L}_d(D) = \frac{1}{N} \sum_p \|D(p) - D^{gt}(p)\|^2,$$

where N is the total number of pixels. $D(p)$ and $D^{gt}(p)$ are the predicted depth and ground-truth depth at pixel p, respectively.

We use the confidence loss following [12] for $\mathcal{L}_c$. For each training step, we first estimate a ground-truth confidence according to the depth-prediction error. The ground truth confidence at pixel p is defined as

$$C^{gt}(p) = e^{-|D(p) - D^{gt}(p)|}$$

The confidence loss $\mathcal{L}_c$, then, is defined as the mean squared error loss between the predicted confidence C and the ground truth C^{gt}:

$$\mathcal{L}_c(C) = \frac{1}{N} \sum_p \|C(p) - C^{gt}(p)\|^2,$$

where $C(p)$ is the predicted confidence at pixel p.

IV. Online 3D Modeling System

This section describes the online 3D-modeling system based on the proposed depth-completion method. Our depth-completion method generates a temporally consistent depth map using a sequential depth-integration framework; therefore, our method is more appropriate for 3D modeling than other methods [6], [8], [17]. Furthermore, the confidence map estimated from our method is used to filter out outlier depths to improve the precision of 3D reconstruction.

For each input frame, the system first estimates the camera pose and sparse 3D map points from a SLAM module. It then computes a depth map and updates a global 3D model at regular frame intervals. The depth map is computed by depth completion (as described in Section III), which uses the sparse map points as sparse depth input. The depth-completion method produces depth maps that include outliers in background regions and occluded areas. Furthermore, the sparse depths may be non-uniformly distributed in a specific area; this may cause unreliable depth estimates. Therefore, it is necessary to remove the outliers and unreliable depths.

To filter out the outliers, we considered two filtering criteria, depth confidence and geometric consistency. The depth confidence criterion uses the confidence map estimated in Section III and aims to remove the low-confidence depths. We regard the

depths with a confidence lower than a certain threshold as unreliable depths and filter them out. Geometric consistency checks the consistency of the estimated depth among previous depth estimates. The current frame is compared to the previous five frames for the consistency check. Similar to multi-view stereo depth filtering, as in [20], we first convert the estimated depth map into a point cloud and re-project it to each of the previous frames. We then compute the discrepancy between the projected depths and the original depths from the previous frames. The discrepancy is defined as $|d_{proj} - d_{orig}|/d_{orig}$, where d_{proj} and d_{orgin} are the projected depth and the original depth, respectively. If an estimated depth does not have at least three frames with a discrepancy smaller than a threshold, we regard the depth as geometrically inconsistent and filter it out. The discrepancy threshold balances accuracy against the completeness of a reconstructed model. In this paper, we heuristically determined the threshold as 0.01 to achieve balanced results for both accuracy and completeness.

Finally, the filtered depth maps are integrated into a global 3D model using a surfel-based fusion method [13]. The surfel-based method represents a 3D model as a collection of surfels, where each surfel has the following attributes: 3D position, normal, color, weight, and radius. Similar to [13], our method defines the surfels that have not been updated in a period as inactive surfels. Then, low-weighted inactive surfels are removed from the global 3D model. Therefore, the 3D model is again cleaned up over time.

V. Experimental Results

A. Dataset and Experimental Setup

This study focuses on the depth completion of outdoor scenes. We used two outdoor benchmark datasets, the *KITTI depth-completion benchmark* [5] and the *Aerial dataset* [8], to evaluate the performance of our depth-completion method.

- **KITTI dataset:** This dataset is composed of color images (1216×352 resolution) and sparse LiDAR measurements (about 4.0% coverage), where each frame is acquired from a mobile platform while driving around diverse outdoor scenes. We evaluated on the validation set instead of the test set because the test set does not provide sequential data.
- **Aerial dataset:** This is a photo-realistic synthetic dataset for various outdoor aerial scenes. It is appropriate for the evaluation of 3D-model reconstruction because it provides ground-truth 3D models and dense depths at full-resolution. It contains training and test datasets, which are generated from 26 independent aerial scanning trials. We considered only 19 scanning sets because the other scanning sets do not include ground-truth pose information. For each frame, we generated 10K random samples (about 2.8% coverage of the image resolution of 752×480) over the ground-truth depths to get the sparse depth input.

We implemented the depth-completion network using the PyTorch library and the 3D-modeling system in an ROS environment. Every network used in the experiments was trained and tested with the original image resolution on a machine with multiple GPUs. The ADAM optimizer was used for training the networks. We set the initial learning rate to 10^{-4}, the decay factor to 0.5, and ran for 30 epochs with a batch size of 8 for all datasets.

TABLE I
RESULTS OF DEPTH COMPLETION. THE ERROR METRICS OF RMSE AND MAE ARE IN METERS (m). DC-CONF-NET$_{\text{FILT}}$ AND OURS$_{\text{FILT}}$ ARE THE RESULTS OF FILTERED DEPTH MAPS USING CONFIDENCE THRESHOLDING

	KITTI dataset			Aerial dataset		
	RMSE	MAE	MRE	RMSE	MAE	MRE
NConv-CNN [14]	0.876	0.236	0.0127	0.713	0.150	0.0047
Sparse-to-Dense [6]	0.857	0.313	0.0187	2.587	1.989	0.0929
DC-Conf-Net [8]	1.018	0.264	0.0138	0.921	0.176	0.0045
MSG-CHN [17]	0.822	0.228	0.0122	0.681	0.139	0.0038
RGB-G&C [21]	0.800	0.215	0.0114	0.752	0.198	0.0129
Ours	**0.776**	**0.213**	**0.0113**	**0.612**	**0.135**	**0.0037**
DC-Conf-Net$_{\text{filt}}$ [8]	0.597	0.181	0.0101	0.394	0.100	0.0028
Ours$_{\text{filt}}$	**0.266**	**0.118**	**0.0087**	**0.146**	**0.077**	**0.0027**

TABLE II
NUMBER OF PARAMETERS, RUNTIME, AND GPU MEMORY CONSUMPTION OF EACH DEPTH-COMPLETION NETWORK. ALL RUNTIMES WERE MEASURED ON A WORKSTATION WITH INTEL XEON CPU (32 CORES), 64 GB OF RAM, AND TITAN-V GPU WITH 12 GB OF MEMORY. OUR METHOD DOES NOT REQUIRE A SIGNIFICANTLY LARGE COMPUTATIONAL TIME, CONSIDERING ITS COMPLEX NETWORK STRUCTURE. IT IS CAPABLE OF REAL-TIME PROCESSING WITH ABOUT 1.9 GB OF MEMORY CONSUMPTION

	#Params	KITTI dataset		Aerial dataset	
		Time [s]	Mem [MB]	Time [s]	Mem [MB]
NConv-CNN [14]	355K	0.01	1658	0.01	1622
Sparse-to-Dense [6]	26.1M	0.01	11434	0.01	10134
DC-Conf-Net [8]	980K	0.04	3098	0.03	2784
MSG-CHN [17]	364K	0.01	1634	0.01	1546
RGB-G&C [21]	2.6M	0.02	1470	0.02	1530
Ours	1.6M	0.04	1950	0.04	1754

B. Comparison With the State-of-the-Arts

To evaluate the performance of our depth-completion method, our method was compared with the state-of-the-art methods **NConv-CNN** [14], **Sparse-to-Dense** [6], **MSG-CHN** [17], **RGB-G&C** [21], and **DC-Conf-Net** [8], using their public source codes. Similar to our method, DC-Conf-Net predicts confidences along with depths, while other methods do not produce confidences. To evaluate the depth-completion performance, we calculated three standard error metrics [5]: *root mean square error* (RMSE), *mean absolute error* (MAE), and *mean absolute relative error* (MRE).

Table I compares the results of each method. Our method had the best performance in all metrics. Sparse-to-Dense and RGB-G&C had relatively good performances on the KITTI dataset, but they do not perform well on the Aerial dataset. Moreover, as tabulated in Table II, their networks require a large number of parameters, over 2.6M. By contrast, our network requires only a moderate number of parameters (1.6M) while significantly outperforming all these methods. In particular, our method

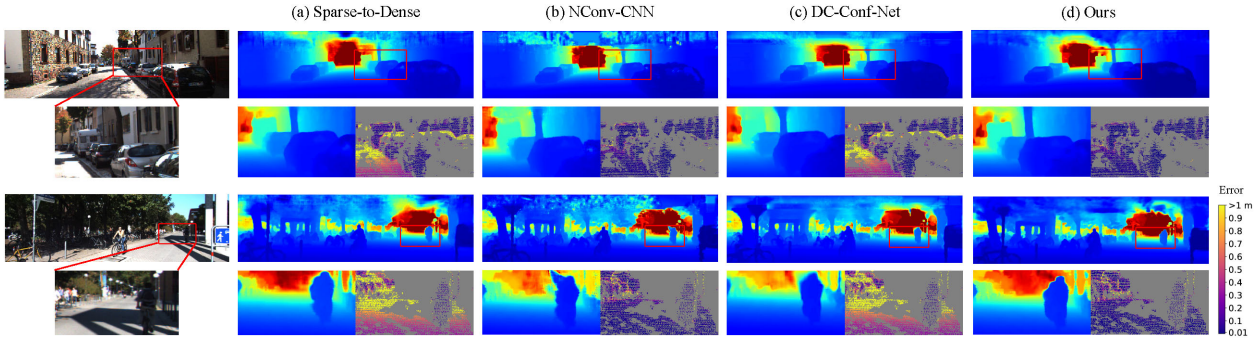


Fig. 3. Qualitative comparison between methods on the KITTI dataset. For each test image, we provide a zoomed-in detail view for the region highlighted with red boxes. We also provide the errors of estimated depths (each bottom right figure) for better comparison. The gray color points in the error maps represent the pixels with no depth information in the ground truth. Sparse-to-Dense [6] and DC-Conf-Net [8] had many high-error points. NConv-CNN [14] produced low-error points but it cannot recover the object shape. Our method obtained both low-error points and detailed shape recovery.

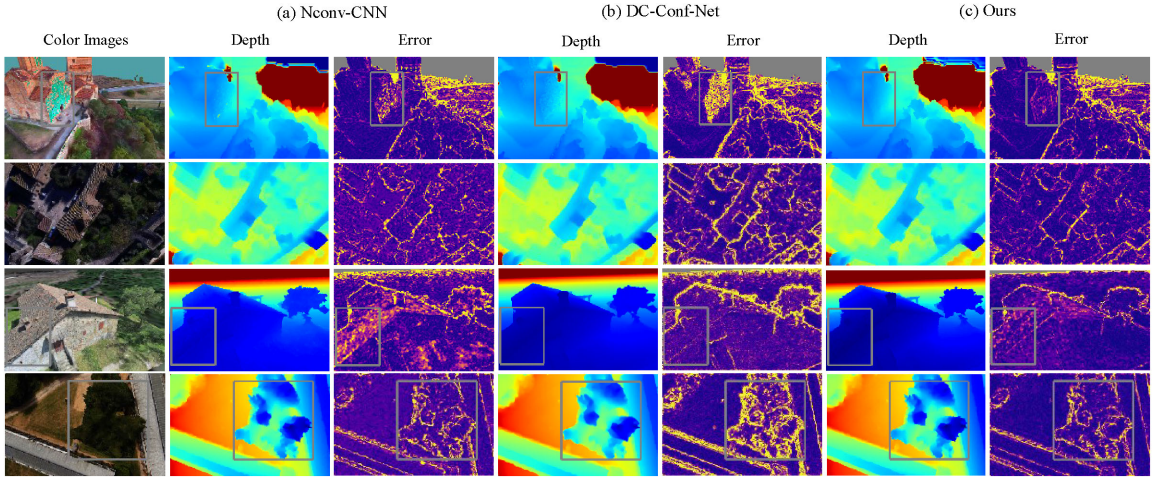


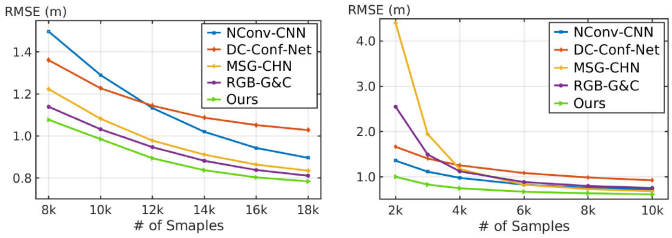
Fig. 4. Qualitative comparison with other methods on Aerial dataset. For each method, we provide the full depth maps as well as the error maps. Our method produced relatively accurate depth maps comparing to the depth maps of (a) NConv-CNN [14] and (b) DC-Conf-Net [8] as highlighted with boxes.

showed a significant performance improvement compared to NConv-CNN. The network structure of NConv-CNN is similar to our initial stage's network structure (Section III.A) except for the sequential depth integration. This suggests that integrating sequential information from the subsequent frames based on the confidence estimates could improve the performance of depth completion.

To analyze the performance of the confidence prediction, we filtered out the lowest-confidence depths in our method and DC-Conf-Net, using their respective thresholds, and evaluated the performance of the remaining depths. The filtered depth maps covered approximately 90% and 96% of the image plane for the KITTI and Aerial datasets, respectively. $\text{Ours}_{\text{filt}}$ and $\text{DC-Conf-Net}_{\text{filt}}$ in Table I represent the results of filtered depths based on their confidence estimates. Both performances were significantly improved compared to their original performances. Especially in our method, the confidence-based filtering reduced the average RMSE by 29.0% and the average MAE by 56.2%. A more detailed discussion about the quality of confidence prediction will be addressed in Section V.C.

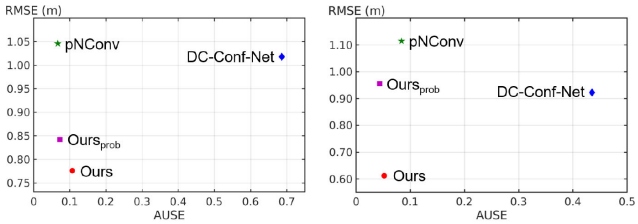
Qualitative comparisons of several methods on the KITTI and Aerial datasets are shown in Fig. 3 and Fig. 4, respectively. As can be seen in these figures, our method produced depth maps with cleaner and sharper object boundaries in both close and distant areas. Moreover, our method largely reduced the depth errors and recovered better details compared to the other methods.

Generalization capability: We also verify the generalization capability of our method by evaluating the depth completion performance under different input depth sparsities. Fig. 5 shows the performances of our method and other methods on KITTI and Aerial datasets with different input depth sparsity. The sparse inputs were randomly sampled from the original inputs according to a given sparsity. With the density decreasing, the performances of most methods were degraded gradually. In particular, MSG-CHN and RGB-G&C showed significant performance drops when the number of samples is lower than 4k on Aerial dataset. On the other hand, our method always outperformed the other methods and provided good results at very sparse inputs. These results demonstrate the robustness of



(a) KITTI dataset (b) Aerial dataset

Fig. 5. RMSEs of depth completion results under different input depth sparsities. Our method always outperforms the other methods and has good performances, even with very sparse inputs.



(a) KITTI dataset (b) Aerial dataset

Fig. 6. The comparison results with respect to AUSE and RMSE on (a) KITTI and (b) Aerial datasets where left-bottom is better. Our method performs best in terms of RMSE while having comparable AUSEs.

our method to various input sparsities. The sequential integration of our method complements a preliminary result from very sparse inputs. Therefore, our method can obtain accurate depth completion results even with very sparse inputs.

C. Evaluation of Confidence Prediction

This section evaluates the quality of the predicted confidence in our method compared to that of DC-Conf-Net [8] and probabilistic NConv (**pNConv**) [23]. pNConv is an extension method of NConv to estimate depth confidences probabilistically. It directly trains depth uncertainties similar to our method but utilizes a different loss function, *aleatoric uncertainty loss*: $L_{prob}(D,V) = \frac{1}{N}\sum_p \frac{\|D(p)-D^{gt}(p)\|^2}{V(p)} + \log(V(p))$, where $V(p)$ is the predicted noise variance at pixel p. This loss explicitly formulates a maximum likelihood problem of the Gaussian error model. Furthermore, we evaluated a variation of our method, $\mathbf{Ours_{prob}}$, which was trained by the aleatoric uncertainty loss instead of the original loss.

To measure the quality of confidences, we used the *area under sparsification error plots* (AUSE) [29]. Fig. 6 shows the comparison results representing the relationship between AUSE and RMSE. As shown in Fig. 6, DC-Conf-Net performs the worst in terms of AUSE. Performing confidence-guide depth refinement through additional networks, DC-Conf-Net used the refined depths to train the confidences. This model did not explicitly associate confidence with depth-prediction error. pNConv had comparable AUSEs, while it performs worst in terms of RMSE. It is designed for unguided depth completion that ignores RGB information; therefore, it generally underperforms the other guided methods. As compared to our original method (**Ours**), $\text{Ours}_{\text{prob}}$ showed better AUSE performances while significantly underperforming on RMSE. The models trained by the aleatoric uncertainty loss could not obtain accurate depth predictions since they can be biased toward the low-error depths [22]. On the other hand, our original method had the best performance in terms of RMSE and comparable quality of confidence prediction simultaneously.

D. 3D Modeling Results

This section demonstrates the performance of the 3D-modeling system proposed in Section IV on the Aerial dataset (synthetic scenes) and on real-world data. The Aerial dataset was used to measure the modeling performance of our depth-completion method quantitatively using the ground-truth information. The real-world data, on the other hand, was used to demonstrate the feasibility of the proposed 3D-modeling system in real-world applications.

Aerial dataset: We used the depth and confidence results computed in the previous experiments (Section V.B) directly to reconstruct 3D models for the Aerial dataset. The models constructed by depths and confidences estimated from our depth-completion method were compared with those of DC-Conf-Net. We computed three metrics [30] for the evaluation, *precision*, *recall*, and *f-score* (P, R, and F), by comparing a constructed model with its ground-truth model. The ground-truth models were created by accumulating all ground-truth depths directly to focus only on observed surfaces. We used the ground-truth poses instead of SLAM outputs to measure the reconstruction quality, regardless of pose uncertainty.

Fig. 7 shows the reconstructed 3D models with their quality measures in two test scenarios. Our method outperformed DC-Conf-Net in terms of overall performance. DC-Conf-Net may produce temporally inconsistent depths; therefore, many depth points were filtered out in the geometric consistency check. Conversely, our method produced temporally consistent depths, which allows us to construct a complete 3D model with high recall. Furthermore, our method can precisely remove outliers based on accurate confidence estimates. This improves the precision of reconstructed models.

Real-world scene: The camera mounted on the MAV captured sequential images at a resolution of 848×480. We used ORB-SLAM [31] for pose estimation and sparse depth estimation. We used the sparse map points as sparse depth input (about 0.25% coverage). Our pretrained model on the Aerial dataset was used for depth completion.

Fig. 1 shows examples of depth-completion results and a reconstructed model of the real-world scene. As can be seen, our modeling system produced a satisfactory reconstruction with globally consistent surfaces. Furthermore, outliers, including background and occluded regions, were clearly removed in the reconstructed model. This indicates that the proposed depth-filtering method is effective and feasible in real-world environments. We also provide the comparison results of our method and the motion-stereo method [10] in the demo video.

(a) DC-Conf-Net (P: 0.907, R: 0.915, F: 0.911) (b) Ours (P: 0.932, R: 0.987, F: 0.959) (c) DC-Conf-Net (P: 0.883, R: 0.917, F: 0.899) (d) Ours (P: 0.895, R: 0.963, F: 0.928)

Fig. 7. 3D-modeling results of two synthetic scenes. Each column shows the reconstructed pointcloud and ground-truth model with per-point error coded by color as described in [30]. For each result, we computed modeling performances in terms of *precision*, *recall*, and *f-score* (P, R, and F).

VI. Conclusion

We proposed a network architecture for depth completion that produces depths with corresponding confidences and presented an online 3D-modeling system based on that depth-completion method. Our method integrates sequential depth information to produce temporally consistent depths with improved accuracy. The predicted confidences of our method accurately represent the true prediction errors of the depth estimates. The proposed modeling system handles large numbers of outliers in the predicted depths using several filtering steps, including confidence-based thresholding and a geometric consistency check. The filtered depths are integrated into a globally consistent 3D model using surfel-based fusion. To the best of our knowledge, this is the first work that implements an online modeling system to reconstruct 3D outdoor scenes using a depth-completion method. To reduce the operating time, a future study should be conducted to incorporate a model compression method [32] into our network.

References

[1] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96–101, 2007.

[2] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[3] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 291–298.

[4] H. Lee, S. Song, and S. Jo, "3d reconstruction using a sparse laser scanner and a single camera for outdoor autonomous vehicle," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 629–634.

[5] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 11–20.

[6] F. Mal and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 4796–4803.

[7] J. Qiu *et al.*, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3313–3322.

[8] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1055–1062, Apr. 2020.

[9] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 501–518.

[10] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 2609–2616.

[11] Y. Hou, J. Kannala, and A. Solin, "Multi-view stereo by temporal non-parametric fusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2651–2660.

[12] X. Yang, Y. Gao, H. Luo, C. Liao, and K.-T. Cheng, "Bayesian denet: monocular depth prediction and frame-wise fusion with synchronized uncertainty," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2701–2713, Nov. 2019.

[13] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

[14] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2423–2436, Oct. 2020.

[15] W. Huang, X. Gong, and M. Y. Yang, "Joint object segmentation and depth upsampling," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 192–196, Feb. 2015.

[16] K. Matsuo and Y. Aoki, "Depth image enhancement using local tangent plane approximations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3574–3583.

[17] A. Li *et al.*, "A multi-scale guided cascade hourglass network for depth completion," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 32–40.

[18] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[19] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia, "Beyond local reasoning for stereo confidence estimation with deep learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 319–334.

[20] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 767–783.

[21] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *Proc. 16th Int. Conf. Mach. Vis. Appl.*, 2019, pp. 1–6.

[22] H. Hekmatian, J. Jin, and S. Al-Stouhi, "Conf-net: Toward high-confidence dense 3d point-cloud with error-map prediction," 2019.

[23] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 014–12 023.

[24] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 834–849.

[25] S. Song, D. Kim, and S. Jo, "Active 3d modeling via online multi-view stereo," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1–8.

[26] H. Knutsson and C.-F. Westin, "Normalized and differential convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1993, pp. 515–523.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.* Springer, 2015, pp. 234–241.

[29] E. Ilg *et al.*, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 652–667.

[30] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[32] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2015, *arXiv:1510.00149*.