2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Deep Multi-view Depth Estimation with Predicted Uncertainty

Tong Ke, Tien Do, Khiem Vuong, Kourosh Sartipi, and Stergios I. Roumeliotis[†]

Abstract— **In this paper, we address the problem of estimating dense depth from a sequence of images using deep neural networks. Specifically, we employ a *dense-optical-flow* network to compute correspondences and then triangulate the point cloud to obtain an initial depth map. Parts of the point cloud, however, may be less accurate than others due to lack of common observations or small parallax. To *further* increase the triangulation accuracy, we introduce a depth-refinement network (DRN) that optimizes the initial depth map based on the image's contextual cues. In particular, the DRN contains an iterative refinement module (IRM) that improves the depth accuracy over iterations by refining the deep features. Lastly, the DRN also predicts the uncertainty in the refined depths, which is desirable in applications such as measurement selection for scene reconstruction. We show experimentally that our algorithm outperforms state-of-the-art approaches in terms of depth accuracy, and verify that our predicted uncertainty is highly correlated to the actual depth error.**

I. Introduction

Estimating dense depth from a sequence of images is necessary in applications such as 3D scene reconstruction and augmented reality. Classical methods address this problem by first computing point correspondences based on hand-crafted matching criteria, and then constructing a 3D point cloud, given the camera pose estimates from structure-from-motion (SFM) [1] or visual(-inertial) simultaneous localization and mapping (SLAM) [2], [3]. They typically fail, however, to obtain reliable correspondences at low-texture or reflective surfaces, which leads to an incomplete scene reconstruction.

Recently, deep learning-based methods have shown the potential to compensate for the aforementioned limitation of the classical methods. Specifically, approaches such as [4], [5] predict dense depth from a single image by taking advantage of images' contextual cues learned from large datasets; hence, they rely less on texture, as compared to classical methods. Moreover, to overcome the scale issue of single-view methods, depth-completion networks (e.g., [6], [7], [8], [9], [10]) leverage sparse point clouds from classical methods and complete the dense depth map using single-view cues. In order to further exploit multi-view information, depth-estimation networks taking multiple images as input have also been considered. In particular, [11], [12] employ cost volumes in their networks to embed geometric information, while [13], [14] explicitly leverage multi-view geometry by estimating dense optical flow.

In this work, we follow the latter paradigm. Specifically, we employ an optical flow network to compute dense correspondences between a keyframe image and its immediate neighbors, and then triangulate the dense matches to compute the 3D point cloud given the cameras' poses. Challenging conditions, however, such as lack of common observations or small parallax cause some points to have low-accuracy depth estimates. To represent these errors, we employ the Hessian and residual of the triangulation least squares and define the confidence scores for the initially triangulated depths, which we then use in the depth-refinement network (DRN).

Although some of the aforementioned issues can be partially alleviated by applying an adaptive frame-selection policy (see Sec. IV), we primarily focus on improving the accuracy by taking advantage of single-image depth estimation networks. Specifically, in order to leverage the image's contextual information as well as the confidence scores, we introduce a DRN that uses as input the initially triangulated depths, their confidence scores, and the keyframe image to produce a refined depth map and its uncertainty (the aleatoric one [15]). In particular, we propose an iterative refinement module (IRM) in its decoder that iteratively refines the deep features extracted by the encoder using a least-squares layer, which significantly improves the depths' accuracy as compared to the initial ones. The depth's uncertainty predicted by the DRN, however, and as shown by our experiments, are highly correlated with the actual depth errors, thus providing valuable criterion for selecting which measurements to use for 3D scene reconstruction. To summarize, our main contributions are:

- We introduce an algorithm for estimating depth from multiple images, which outperforms state-of-the-art methods (∼20% lower RMSE on ScanNet [16] dataset).
- We propose a depth-refinement network (DRN) with an iterative refinement module (IRM) that greatly improves the depths' accuracy and estimates their uncertainty.
- We further improve the depth estimation from multi-view by applying an adaptive frame-selection policy.

II. Related Work

Multi-view depth-estimation methods can be classified as:

Depth completion: One approach towards dense depth estimation from multiple views is to: (i) Create a sparse point cloud (by tracking distinct 2D points across images and triangulating their 3D positions) and then (ii) Employ a depth-completion neural network that takes the sparse depth image along with the RGB image as inputs and exploits the scene's context to create a dense-depth estimate (e.g., [6], [7], [8], [9], [10]). Although these approaches have relatively low processing requirements, they are typically sensitive to the inaccuracies and sparsity level of their depth input; thus, they often fail to produce accurate depth estimates in textureless regions that lack sparse depth information. To overcome this limitation, we obtain a *dense initial depth map* by triangulating a dense set of correspondences established by

[†]Tong Ke, Tien Do, Khiem Vuong, Kourosh Sartipi, and Stergios I. Roumeliotis are with University of Minnesota, Minneapolis, MN 55455 {kexxx069, doxxx104, vuong067, sarti009, stergios}@umn.edu.

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

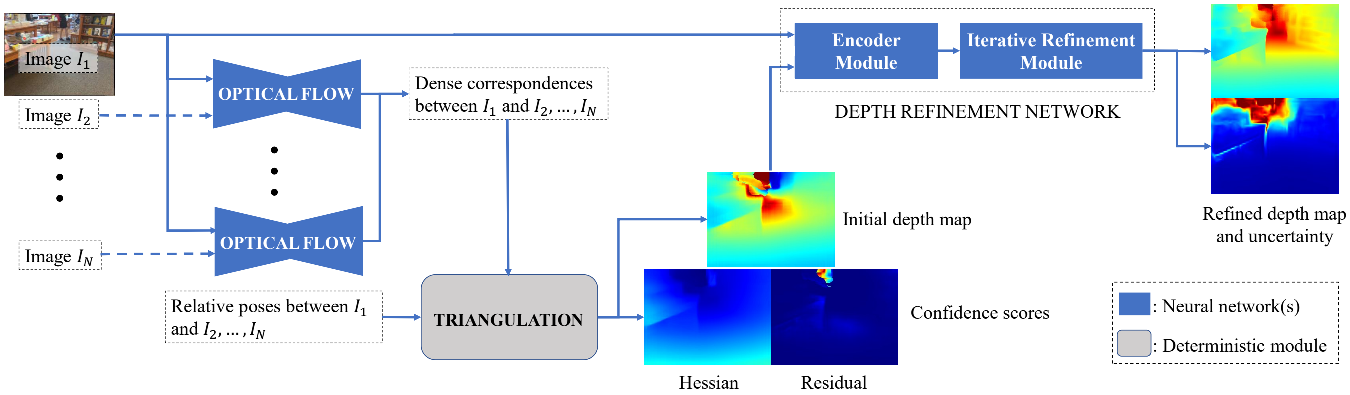

Fig. 1. Overview of the system. For a keyframe I_1, Step 1: We compute the dense optical flow between image I_1 and $I_k, (k = 2 \dots N)$. Step 2: We triangulate the initial depth map of I_1 and compute its confidence scores (see Sec. III-A). Step 3: The DRN (Sec. III-B) takes image I_1, the initial depth map, and confidence scores as input and iteratively [through its decoder (see Sect. III-B.2)] outputs the refined depth map and its uncertainties. Note that the DRN includes an IRM (Sec. III-B.2) in its decoder, whose objective is to preserve initial depth estimates of high confidences scores.

an optical flow method such as [17]. By doing so, and as we show experimentally, we significantly improve accuracy as compared to sparse-to-dense depth completion approaches.

Depth cost volume: Alternatively, dense information from multiple frames is obtained by estimating a depth probability volume from depth cost volumes [18], [12], [19], [20], [11], [21]. Specifically, by aggregating information across multiple frames, depth-cost-volume approaches yield higher accuracy compared to sparse-to-dense methods. Their precision, however, is bounded by the range of their depth sweeping planes predefined in the cost volumes. In contrast, our algorithm relies on optical flow and triangulation and thus, it is not restricted by the limited range and discretization effects of cost volumes. In our experiments, we show that our method outperforms [11], a state-of-the-art depth cost volume approach, by a significant margin (∼20% lower RMSE).

Flow-to-depth: Lastly and closely related to our work is the approach of estimating dense depth from dense optical flow [22], [13]. Specifically, an initial depth map along with its confidence scores are first obtained through least-squares triangulation of the dense optical-flow correspondences. Subsequently, the initial depth map is further improved by a DRN, often realized as a deep autoencoder, using its confidence scores and the RGB image. This DRN, however, may inaccurately modify the initial depth map, especially for pixels with high confidence scores (see Sect. III-B for more details). Previous works [23], [24] address this issue, in the context of depth completion when a *sparse ground truth depth map* is given, by (i) replacing the refined depths with the ground truth ones when these are available, and (ii) propagating this information to the neighboring pixels via a convolution spatial propagation operator. Our initial depth map, however, may contain significant noise and outliers, hence it cannot be employed as ground truth for depth propagation.

To address this issue, in this work, we improve the DRN by introducing an IRM that *seeks to minimize the difference between the initial and final depth estimates of pixels with high confidence scores*. To do so, we iteratively refine the joint deep feature representation of the RGB image, initial depth, and its confidence scores using a least-squares layer, analogous to the one in [25], [26]. As shown Sec. IV, the proposed IRM leads to significant accuracy improvement as compared to simply using the confidence score as input to a neural network. Furthermore, our approach estimates the refined depths' uncertainty (aleatoric uncertainty model [27], [15], [28]), which is employed to fuse dense-depth estimates across a scene [12], [7] (see 3D reconstruction experiment in Sect. IV-C).

III. Technical Approach

We hereafter present our method for estimating the dense depth map of an image I_1 given $N-1$ adjacent images I_2, I_3, $\dots$, I_N and their corresponding relative camera poses (these can be estimated online, e.g., by visual-inertial SLAM that combines inertial measurements with sparse visual feature tracks [29]). Fig. 1 depicts an overview of our pipeline. Specifically, we first employ a dense optical flow network between images I_1 and I_k, $(k = 2, \dots, N)$ to establish point correspondences and then use the cameras' relative poses for triangulation. As a result, we obtain an *initial depth map* for I_1, as well as the *triangulation's confidence scores*. Lastly, we employ a DRN that takes as input the initial depth map, the confidences scores, and image I_1 as input to refine the depth map and predict its uncertainty. As mentioned earlier, in the DRN, we include an IRM which significantly increases the accuracy over iterations using a least-squares layer. Next, we describe each module in detail.

A. Optical Flow and Triangulation

For estimating optical flow, we employ the network of RAFT [17] that takes as input a source image I_1 and a target image I_k and computes a displacement $\delta \mathbf{u}_i$ for every pixel position ${}^1\mathbf{u}_i = \begin{bmatrix} {}^1x_i & {}^1y_i & 1 \end{bmatrix}^T$ of I_1, so that ${}^k\mathbf{u}_i = {}^1\mathbf{u}_i + \delta \mathbf{u}_i$ is its corresponding pixel in I_k. Given a keyframe image I_1, for which we estimate the depth, and $N-1$ adjacent frames $I_2, \dots, I_N$, we run the optical-flow network pairwise between I_1 and I_k, $k = 2, \dots, N$, so as to find the corresponding pixel in I_k for every pixel of I_1.

From these correspondences and the relative poses of frames I_1 and I_k, $k = 2, \dots, N$, we compute the initial depth

$\bar{d}_i$ of each pixel in I_1 via triangulation. Specifically, we solve the following linear least-squares problem:

$$\bar{d}_i = \arg\min_{d_i} \sum_{k=2}^{N} \left\| ({}^k\mathbf{u}'_i/\|{}^k\mathbf{u}'_i\|) \times ({}^k_1\mathbf{R}\,{}^1\mathbf{u}'_i d_i + {}^k\mathbf{p}_1) \right\|^2 \qquad (1)$$

where ${}^k_1\mathbf{R}$, ${}^k\mathbf{p}_1$ are the orientation and position of frame I_1 with respect to frame I_k, and ${}^n\mathbf{u}'_i \triangleq \mathbf{K}^{-1\,n}\mathbf{u}_i, n = 1\ldots N$ with $\mathbf{K}$ being the camera intrinsic matrix. Note that ${}^k\mathbf{u}_i$ may be outside of the image, while the triangulation still yields reasonable depths at those points, though typically less accurate. As mentioned earlier, the initial depth map will be further improved by the DRN using as confidence scores the square root of the Hessian (which is a scalar here) and the norm of the residual from the least squares. The former reflects the quality of triangulation, i.e., the parallax, while the latter represents the reprojection error. Optionally, we select frames $I_2 \ldots I_N$ through an adaptive policy where I_{k+1} is selected only if the relative rotation angle or distance between it and I_k exceeds thresholds t_r, t_d, respectively. Using such policy, instead of using a time-based, fixed-size image sampling, improves the initial depth map's accuracy.

For training the optical flow network, a naive way is to apply an ℓ_1 or ℓ_2 loss on the depth computed from (1). This loss, however, only imposes constraints for the optical flow along the direction affecting the depth, i.e., the epipolar line, but not the direction perpendicular to it, where the magnitude of the triangulation residual is determined. To capture the errors in all directions, we propose the following loss by substituting the ground truth depth d_i^* of pixel ${}^1\mathbf{u}_i$ to the least squares' cost function of a two-view triangulation:

$$l_o = \sum_i \left\| ({}^2\mathbf{u}_i/\|{}^2\mathbf{u}_i\|) \times ({}^2_1\mathbf{R}\,{}^1\mathbf{u}_i d_i^* + {}^2\mathbf{p}_1) \right\|^2 \qquad (2)$$

B. Depth Refinement Network (DRN)

The initial depth map and confidence scores (see Sect. III-A) are used by the DRN for further accuracy improvement. In particular, the DRN seeks to preserve the initial triangulated depths for pixels with high confidence scores, while improving the rest of the depth map using the prior learned from the training data. Previous works (e.g., [22], [13]) propose simply passing the initial depth map, its confidence scores, and the RGB image to an autoencoder to refine depth. As shown in Fig. 2, however, although the refined depths from an autoencoder network are overall more accurate than the triangulated depths, they are often incorrectly modified in the low-error regions.To overcome this limitation, we employ the IRM described in Sect. III-B.2 (see Fig. 2). Lastly, note that the DRN approximates each output depth pixel as an independent Laplace random variable and the training of depth and uncertainty is performed with the aleatoric uncertainty model [15], [27]. The structure of DRN (Fig. 1) comprises: (i) An Encoder Module, and (ii) An IRM. In what follows, we describe each module in details.

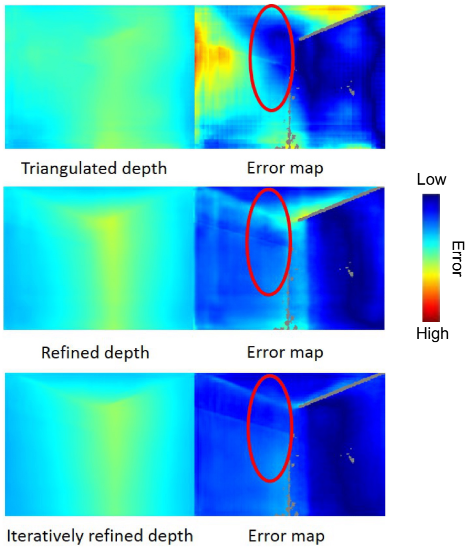

Fig. 2. Effect of the iterative depth refinement. The red ellipses indicate low-error regions in the triangulated depth. The initial step of the DRN causes the local error to increase but the iterative refinement module improves the depth estimates of these regions. The error color map uses red/blue for high/low error.

1) Encoder Module: This is employs an extension of the depth prediction architecture [6]. Specifically, we first use three ResNet-18 [30] architectures to extract the deep-feature representations of the RGB image, the surface normal (predicted by a neural network from the RGB image as in [6]), and the triangulated depth map with its confidence scores, and then concatenate them to generate a joint feature tensor h. Given h, the estimated depth map $\hat{d}$ and its uncertainty $\hat{\sigma}$ are computed as:

$$\hat{d} = \mathbb{D}(h;\theta), \;\; \hat{\sigma} = \Sigma(h;\phi) \qquad (3)$$

where θ and ϕ are the learned parameters of the depth decoder $\mathbb{D}$ and the uncertainty decoder Σ, respectively. Next, we describe the IRM that further refines the feature h to obtain a better depth estimate $\hat{d}$.

2) Iterative Refinement Module (IRM): The output of the depth decoder (a modified Panoptic Feature Pyramid Network [31]) $\hat{d}$ from the initial step of DRN may contain erroneous estimates. The IRM seeks to update the deep feature h such that the difference between the estimated depth $\hat{d}$ and the initial depth estimate $\bar{d}$ becomes smaller for pixels with high confident values $\bar{c}$. This can be formulated as a weighted least-squares problem, in which the weights $\hat{w}_i$ are computed from the deep feature h via a weight decoder $\mathbb{W}$ with the learned parameters γ:

$$\mathbb{C}(h) = \sum_i \hat{w}_i(\hat{d}_i - \bar{d}_i)^2 = \sum_i \mathbb{W}_i(h;\gamma)(\mathbb{D}_i(h;\theta) - \bar{d}_i)^2 \qquad (4)$$

where i indicates pixel position. Note that $\hat{w}_i$ contains the joint information of the confidence scores and other inputs to the DRN. We use a gated recurrent unit (GRU) [32] to iteratively update h as:

$$h^{(k+1)} = \mathrm{GRU}\left(h^{(k)}, \nabla_h \mathbb{C}\left(h^{(k)}\right)\right) \qquad (5)$$

where $\nabla_h \mathbb{C}(h)$ is computed using automatic differentiation [33]. The reset and update gates inside each GRU learn to select the information between memory (deep feature $h^{(k)}$) and geometry (minimizing $\mathbb{C}(h)$ via $\nabla_h \mathbb{C}$). Note that, for each iteration of the IRM, we only update the deep feature h, while we keep fixed the parameters of the GRU, the

TABLE I

PERFORMANCE OF DEPTH PREDICTION ON SCANNET TEST SET

Method	Abs. Rel↓	Sq. Rel↓	log-RMSE↓	i-RMSE↓	RMSE ↓	$E(\hat{d},\delta)$				
						1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
DDE-VISLAM [6]	0.156	0.079	0.174	0.134	0.300	30.92	52.82	80.22	94.20	98.25
NeuralRGBD [12]	0.097	0.050	0.132	0.093	0.249	-	-	90.60	97.50	99.30
Flow2Depth [13]	0.076	0.029	0.108	0.077	0.199	-	-	93.30	98.40	99.60
DeepV2D [11]	0.078	0.054	0.102	0.072	0.201	54.34	77.56	94.55	98.73	99.36
Ours (fixed)	**0.068**	**0.026**	**0.091**	**0.064**	**0.178**	**57.38**	**80.42**	**95.75**	**98.92**	**99.61**
Ours (adaptive)	**0.058**	**0.018**	**0.082**	**0.058**	**0.162**	**62.63**	**84.34**	**96.77**	**99.30**	**99.77**

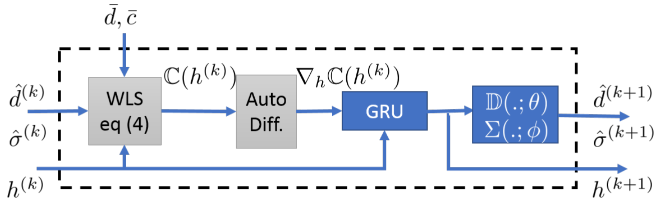

Fig. 3. An update block inside the IRM of Fig. 1. At iteration k, it takes as input the feature $h^{(k)}$, estimated depth $\hat{d}^{(k)}$ at iteration k, and the optical-flow-based depth $\bar{d}$ along with its confidence $\bar{c}$, and outputs the updated feature $h^{(k+1)}$ and the updated estimated depth $\hat{d}^{(k+1)}$ and its confidence $\hat{\sigma}_{k+1}$.

depth decoder $\mathbb{D}$, and the least-squares weights $\mathbb{W}$. The whole network's parameters will later be updated through backpropagation with the ground-truth depths. Lastly, the refined depth and its uncertainty are updated as:

$$\hat{d}^{(k+1)} = \mathbb{D}(h^{(k+1)};\theta),\ \hat{\sigma}^{(k+1)} = \Sigma(h^{(k+1)};\phi) \tag{6}$$

Fig. 3 depicts the iterative refinement process for iteration k. Note that the learned parameters in the update block are shared across iterations.

During training, we execute the above optimization for a fixed number of K iterations, where we employ the negative log-likelihood loss on the estimated depths with Laplace distribution [15], [27]:

$$l_r = \sum_{k=0}^{K} \lambda^{K-k}\left(\sum_i \frac{|\hat{d}_i^{(k)} - d_i^*|}{\hat{\sigma}_i^{(k)}} + \log\ \hat{\sigma}_i^{(k)}\right) \tag{7}$$

with d^* the ground truth depth and $\lambda < 1$ a constant damping factor. In our ScanNet experiment, we found through trial and error that $K = 5$ and $\lambda = 0.83$ offer balance between accuracy and efficiency.

IV. EXPERIMENTAL RESULTS

In this section, we experimentally compare the performance of our method against state-of-the-art approaches and analyze the effect of the presented modules in ablation studies. To this end, we employ ScanNet [16] and a dataset collected with Azure Kinect [34] for evaluation.

Evaluation metrics: The accuracy of depths is assessed using multiple standard metrics, including: Mean absolute relative error (Abs. Rel); Mean square relative error (Sq. Rel); Logarithmic root mean square error (log-rmse); Root mean square error of the inverse depth values (iRMSE); Root mean square error (RMSE); and $E(\hat{d},\delta)$ with $\delta = 1.05, 1.1, 1.25, 1.25^2, 1.25^3$, defined as the percentage of the estimated depths $\hat{d}$ for which $\max(\frac{\hat{d}}{d^*}, \frac{d^*}{\hat{d}}) < \delta$, where d^* is the ground-truth depth.

Experiment setup: The networks in this paper have been implemented in PyTorch [33] and our code is available at `https://github.com/MARSLab-UMN/DeepMultiviewDepth`. The optical-flow network RAFT is trained using the loss in (2) with the optimizer configuration of [17]. To train the DRN, we employ the Adam optimizer [35] with a learning rate of 10^{-4}. The training was done on an NVIDIA Tesla V100 GPU with 32GB of memory with a batch size of 16. The results of other works are obtained by running the original authors' code with their provided network weights (when available).

A. Comparison on ScanNet Dataset

ScanNet [16] is an RGB-D video dataset with more than 1500 sequences, annotated with 3D camera poses. We employ the ScanNet standard training set to train our network and an evaluation set provided by [36] to assess our network's performance against the state-of-the-art competing approaches that are also trained on ScanNet: *DDE-VISLAM* [6][1], *NeuralRGBD* [12], *Flow2Depth* [13], and *DeepV2D* [11]. We employ the ground-truth poses for evaluation, and unless otherwise specified, five images with a fixed skipping interval of 5 are provided for depth estimation per each input sequence, where the depth is estimated for the middle (i.e., third) image. For our algorithm, in addition to the fixed interval evaluation policy, denoted by *Ours (fixed)*, we employ an adaptive frame-selection policy (see Sect. III-A) with $t_r = 0.1\ rad, t_d = 0.08\ m$, denoted as *Ours (adaptive)*.

Table I summarizes the quantitative evaluation results of our proposed method and other state-of-the-art algorithms on the ScanNet.[2] As evident, *Ours (fixed)* outperforms all alternative approaches with a clear margin in all evaluation metrics. We further improve the performance by employing the adaptive frame-selection policy [*Ours (adaptive)*], featuring an overall ∼20% decrease in RMSE as compared to *DeepV2D* and *Flow2Depth*.

Based on Fig. 4 (top two rows), which depicts qualitative results of our method on ScanNet, we observe that: (i) The final refined depth map is closer to the ground truth as compared to the initial one, demonstrating the effectiveness of our depth refinement network; (ii) The error map and the predicted uncertainty map (color-coded with the same scale) are highly correlated, which enables us to employ the latter for measurement selection and fusion (see Sect. IV-C).

[1] Note that the sparse input points in the evaluation set [36] are significantly fewer than the set used by *DDE-VISLAM* for training; hence its performance is lower than the values reported in [6].

[2] Results of *Flow2Depth* and *NeuralRGBD* are obtained from [13].

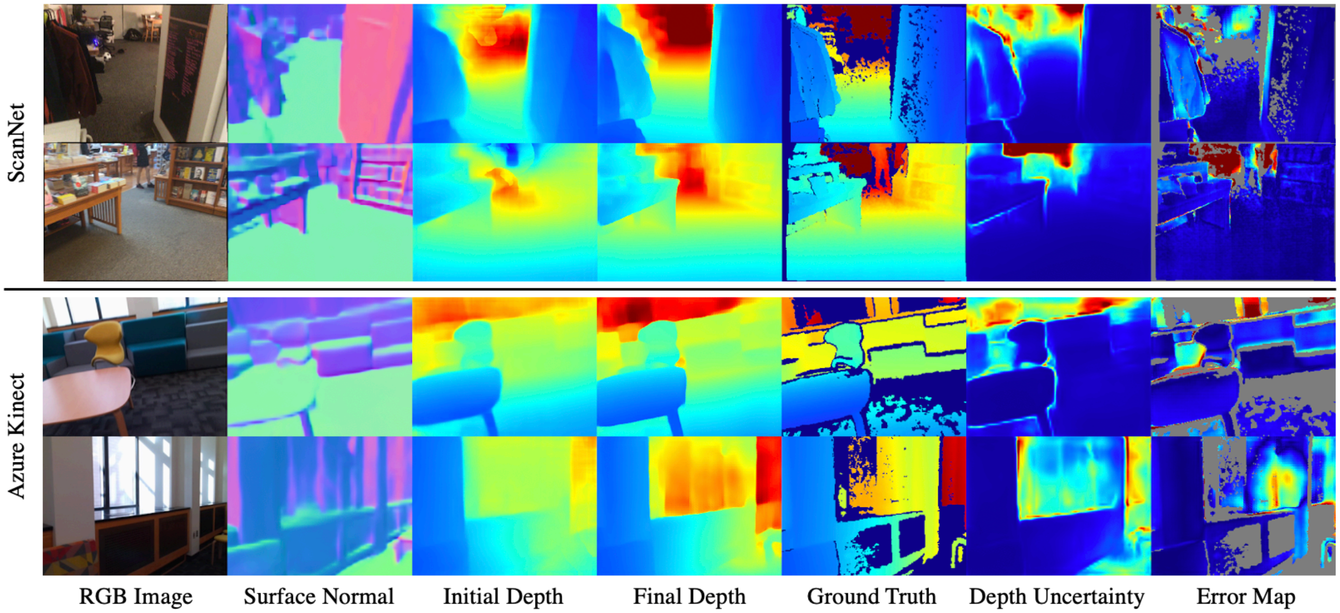

Fig. 4. Inputs, outputs, and intermediate results of our method. Note that the refined depths are significantly more accurate compared to the initial triangulated depths. Additionally, the uncertainty map is strongly correlated with the actual error map.

Lastly, we measured the inference time to compute one depth map of *DDE-VISLAM*, *DeepV2D*, and *Ours* on a single NVIDIA Tesla V100 GPU. *DDE-VISLAM* takes, on average, only 44 ms, at the expense of lower accuracy. *Ours* and *DeepV2D* take 168 ms and 362 ms, respectively.

B. Comparison on Azure Kinect dataset

To verify the generalization capability of our model, we perform cross-dataset evaluation by using the model trained on ScanNet to test on a dataset we collected with Azure Kinect [34] containing 1528 images. We employ the depth-sensor data as the ground truth, while the sliding-window filter of [29] is used to estimate the camera poses. For *Ours (fixed)* and DeepV2D, five images with a fixed skipping interval of 3 are provided for depth estimation per each input sequence where the depth is estimated for the middle (i.e., third) image, while *Ours (adaptive)* employs five images with adaptive policy. Table II shows that *Ours (fixed)* outperforms *DeepV2D* [11] in all metrics and *DDE-VISLAM* [6] in RMSE, 1.05, and 1.10. Moreover, the adaptive frame-selection policy employed by *Ours (adaptive)* further improves the results and achieves the best accuracy in all metrics. Fig. 4 (bottom two rows) depicts the qualitative results of our method on the Azure Kinect dataset.

TABLE II

PERFORMANCE OF DEPTH PREDICTION ON AZURE KINECT TEST SET.

Method	RMSE ↓	$E(\hat{d},\delta)$				
		1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
DDE-VISLAM [6]	0.298	28.75	50.46	**86.67**	**98.11**	**99.74**
DeepV2D [11]	0.321	33.62	54.65	83.46	96.03	98.63
Ours (fixed)	**0.287**	**33.72**	**56.68**	85.49	97.59	99.46
Ours (adaptive)	**0.265**	**35.97**	**58.86**	**87.35**	**98.33**	**99.86**

C. Uncertainty Estimation

To verify the correlation between the predicted uncertainty estimates and the actual errors, we compare the depth error statistics when excluding points whose uncertainty estimates are larger than certain thresholds as shown in Table III [using results of *Ours (adaptive)*]. Decreasing the value of the acceptable predicted uncertainty $\hat{\sigma}$ results in more accurate depth estimates at a small loss of image coverage. For example, we retain 90% of the depth values and reduce the RMSE by ∼20% when excluding points with uncertainty above 0.1. Additionally, the impact of the uncertainty-based masking of the predicted depth images on scene reconstruction (from the ScanNet dataset) is depicted in Fig. 5, where the depth RMSE is reduced by more than 1.4 (3.1) times for the scenes in the top (bottom) row, while only removing 20% of the total depth estimates. Hence, we demonstrated quantitatively and qualitatively that the uncertainty-based depth masking improves reconstruction accuracy.

TABLE III

DEPTH ACCURACY WITH DIFFERENT UNCERTAINTY THRESHOLDS.

$\hat{\sigma}$ thres.	Valid (%)	RMSE ↓	$E(\hat{d},\delta)$		
			1.05 ↑	1.10↑	1.25↑
0.5	99.66	0.159	63.25	84.49	96.63
0.16	96.11	0.132	64.40	85.63	97.25
0.10	91.71	0.117	65.43	86.54	97.66
0.08	88.37	0.110	66.09	87.04	97.81

D. Ablation Study

In this section, we analyze each component of our pipeline that contributes to the overall performance gain (19% in RMSE as compared to other state-of-the-art methods).

Sparse vs. Dense: In the proposed method, a *dense* initial depth map is provided to the DRN, instead of a sparse one as in depth-completion approaches. In order to study the effect of the initial depth map's density, we compare our DRN, without the IRM in its decoder [*Ours (w/o IRM)* ; see Sect. III-B], to the DDE-VISLAM [6], a depth completion network that takes sparse depth as input. Specifically, we randomly sample a fixed number of sparse points (e.g., 10, 100, 200 in Table IV) from the initial triangulated depth

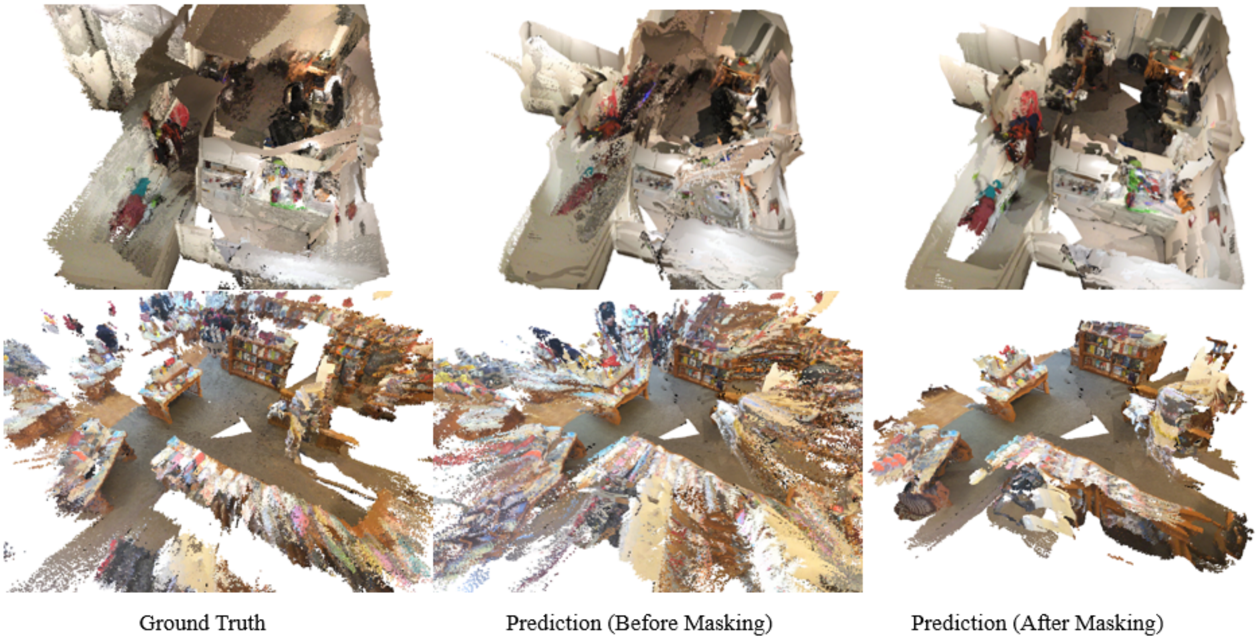

Fig. 5. 3D scene reconstructions using the ground-truth, predicted, and masked depths.

map that have high confidence scores (see Sect. III-A). These sampled depths are added to the sparse depth input of the *DDE-VISLAM*. In Table IV, we show that providing the dense depth estimates together with confidence scores contributes ∼14.5% in RMSE improvement as compared to the depth completion (*DDE-VISLAM+200*) and other state-of-the-art approaches.[3] These results confirm our hypothesis that employing a dense, instead of a sparse, initial depth map improves accuracy.

TABLE IV

DEPTH ACCURACY WITH SPARSE AND DENSE INPUT

Method	Dense	RMSE ↓	$E(\hat{d},\delta)$		
			1.05 ↑	1.10↑	1.25↑
DDE-VISLAM	✗	0.300	30.92	52.82	80.22
DDE-VISLAM + 10	✗	0.218	45.92	70.92	92.21
DDE-VISLAM + 100	✗	0.201	51.23	75.72	93.76
DDE-VISLAM + 200	✗	0.200	51.43	75.91	93.79
Ours (w/o IRM)	✓	**0.171**	**59.25**	**82.32**	**96.16**

Iterative vs. Non-iterative: To demonstrate the effectiveness of the proposed IRM (Sect. III-B.2), we compare our DRN with and without it, denoted as *Ours (w/ IRM)*, and *Ours (w/o IRM)*, respectively. During training, we use five iterations, while at inference time, we use seven iterations. Table V shows that the IRM contributes an additional 4.5% improvement in RMSE. Furthermore, we observe that there is little improvement after seven iterations, hence we limit the number of refinement steps during inference accordingly.

Importance of the triangulation confidence scores: To assess the significance of the initial depths' confidence scores (Sec. III-A), we train our DRN with the IRM as previously described, with four different input options: (i) The triangulated depth map ($\bar{d}$); (ii) The triangulated depth map and the residual as confidence score ($\bar{d},\bar{c}_r$); (iii) The triangulated depth map and the square root of the Hessian as confidence score ($\bar{d},\bar{c}_h$); (iv) The triangulated depth map and both confidence scores ($\bar{d},\bar{c}$), $\bar{c} = \{\bar{c}_r, \bar{c}_h\}$. As evident from Table VI, (iv) yields the best result with less than 0.02% increase in parameters [see Param.s column in (M) millions].

TABLE V

DEPTH ACCURACY WITH IRM

Method	Iterations	RMSE ↓	$E(\hat{d},\delta)$		
			1.05 ↑	1.10↑	1.25↑
Ours (w/o IRM)	0	0.171	59.25	82.32	96.16
Ours (w/ IRM)	1	0.166	61.60	83.66	96.55
	3	0.163	62.47	84.20	96.72
	5	**0.162**	62.59	84.30	96.76
	7	**0.162**	62.63	84.34	96.77
	9	**0.162**	**62.64**	**84.35**	**96.78**

TABLE VI

DEPTH ACCURACY WITH DIFFERENT INPUT CONFIDENCE SCORES

Configuration	Param.s (M)	RMSE ↓	$E(\hat{d},\delta)$		
			1.05 ↑	1.10↑	1.25↑
$\bar{d}$	39.241	0.168	61.79	83.60	96.28
$\bar{d},\bar{c}_r$	39.244	0.164	62.61	84.33	96.48
$\bar{d},\bar{c}_h$	39.244	0.168	61.98	83.39	96.35
$\bar{d},\bar{c}$	39.247	**0.162**	**62.63**	**84.34**	**96.77**

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced an algorithm that employs multiple image to compute the dense depth representation of scene with the corresponding uncertainty. Specifically, pixels tracked by dense optical flow are triangulated and provided to a depth refinement network (DRN) that further improves depth-estimation accuracy. To do so, the DRN first extracts deep features from the inputs and then performs a neural least-squares optimization within its iterative refinement module. In addition to the depth estimates, their corresponding uncertainty is predicted which is shown experimentally to be highly correlated with the actual depth errors. As part of our future work, we plan to employ the estimated uncertainty for selecting the depth measurement to be used in global 3D scene reconstruction.

[3] Note that *DDE-VISLAM+200* performs comparable, in terms of RMSE, to the state-of-the-art methods *DeepV2D* and *Flow2Depth*.

References

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. of the IEEE International Conference on Robotics and Automation*, Las Vegas, NV, June 26 – July 1 2016, pp. 4104–4113.

[2] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug 2018.

[4] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 18–22 2018, pp. 2002–2011.

[5] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 18–22 2018, pp. 2041–2050.

[6] K. Sartipi, T. Do, T. Ke, K. Vuong, and S. I. Roumeliotis, "Deep depth estimation from visual-inertial slam," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Online, Oct. 25–29 2020.

[7] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1055–1062, 2020.

[8] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1899–1906, 2020.

[9] C. Qu, T. Nguyen, and C. Taylor, "Depth completion via deep basis fitting," in *Proc. of the IEEE Winter Conference on Applications of Computer Vision*, Online, Mar. 2–4 2020, pp. 71–80.

[10] A. Sinha, Z. Murez, J. Bartolozzi, V. Badrinarayanan, and A. Rabinovich, "Deltas: Depth estimation by learning triangulation and densification of sparse points," in *Proc. of the European Conference on Computer Vision*, Online, Aug. 23–28 2020.

[11] Z. Teed and J. Deng, "DeepV2D: Video to depth with differentiable structure from motion," in *Proc. of the International Conference on Learning Representation*, Online, Apr. 27–30 2020.

[12] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural RGBD sensing: Depth and uncertainty from a video camera," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 10 986–10 995.

[13] J. Xie, C. Lei, Z. Li, L. E. Li, and Q. Chen, "Video depth estimation by fusing flow-to-depth proposals," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Online, Oct. 25–29 2020.

[14] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," in *SIGGRAPH*, Online, Aug. 17–28 2020.

[15] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proc. of the Advances in Neural Information Processing Systems*, Long Beach, CA, Dec. 4–9 2017, pp. 5574–5584.

[16] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26 2017, pp. 5828–5839.

[17] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Proc. of the European Conference on Computer Vision*, Online, Aug. 23–28 2020.

[18] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 767–783.

[19] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proc. of the European Conference on Computer Vision*, Munich, Germany, Sept. 8–14 2018, pp. 822–838.

[20] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, "Deepsfm: Structure from motion via deep bundle adjustment," in *Proc. of the European Conference on Computer Vision*, Online, Aug. 23–28 2020.

[21] X. Long, L. Liu, C. Theobalt, and W. Wang, "Occlusion-aware depth estimation with adaptive normal constraints," in *Proc. of the European Conference on Computer Vision*, Online, Aug. 23–28 2020.

[22] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman, "Learning the depths of moving people by watching frozen people," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 4521–4530.

[23] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. of the European Conference on Computer Vision*, Munich, Germany, Sept. 8–14 2018, pp. 103–119.

[24] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, "Depth completion from sparse lidar data with depth-normal constraints," in *Proc. of the IEEE International Conference on Computer Vision*, Seoul, Korea, Oct. 27 – Nov. 2 2019, pp. 2811–2820.

[25] S. A. H. Hosseini, B. Yaman, S. Moeller, M. Hong, and M. Akçakaya, "Dense recurrent neural networks for accelerated mri: History-cognizant unrolling of optimization algorithms," *Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1280–1291, 2020.

[26] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, "Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data," *Magnetic resonance in medicine*, vol. 84, no. 6, pp. 3172–3191, 2020.

[27] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Proc. of the European Conference on Computer Vision*, Munich, Germany, Sept. 8–14 2018, pp. 652–667.

[28] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Online, June 14–19 2020, pp. 3227–3237.

[29] K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Proc. of Robotics: Science and Systems*, Rome, Italy, July 12–16 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[31] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 6399–6408.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 6–12 2019, pp. 8024–8035.

[34] Microsoft, "Azure Kinect DK," https://azure.microsoft.com/en-us/services/kinect-dk/.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the International Conference on Learning Representations*, San Juan, Puerto Rico, May 2–4 2016.

[36] [Online]. Available: https://github.com/hufu6371/DORN