Video Depth Estimation by Fusing Flow-to-Depth Proposals

Jiaxin Xie, Chenyang Lei, Zhuwen Li, Li Erran Li, and Qifeng Chen

Abstract— **Depth from a monocular video can enable billions of devices and robots with a single camera to see the world in 3D. In this paper, we present a model for video depth estimation, which consists of a flow-to-depth layer, a camera pose refinement module, and a depth fusion network. Given optical flow and camera poses, our flow-to-depth layer generates depth proposals and their corresponding confidence maps by explicitly solving an epipolar geometry optimization problem. Our flow-to-depth layer is differentiable, and thus we can refine camera poses by maximizing the aggregated confidence in the camera pose refinement module. Our depth fusion network can utilize the target frame, depth proposals, and confidence maps inferred from different neighboring frames to produce the final depth map. Furthermore, the depth fusion network can additionally take the depth proposals generated by other methods to further improve the results. The experiments on three public datasets show that our approach outperforms state-of-the-art depth estimation methods, and has reasonable cross-dataset generalization ability: our model trained on KITTI still performs well on the unseen Waymo dataset.**

I. INTRODUCTION

Accurate dense depth estimation from a monocular video stream can be a backbone algorithm for autonomous robots and mobile devices. For autonomous ground or aerial vehicles, video depth estimation can provide additional information for navigation and obstacle avoidance. A mobile device with a low-cost monocular camera can enable tremendous augmented reality applications without the need for dedicated depth sensors.

A line of research work on monocular depth estimation has been dedicated to single image depth estimation [1]–[5]. However, single image depth estimation methods heavily rely on image priors learned from data, which may not generalize well to unseen scenes. Since it is difficult to obtain extremely accurate depth maps from single image, some researchers focus on depth from video by utilizing multiple video frames [6]–[11]. These approaches usually directly regress depth from deep features aggregated from multiple frames [9] or cost volumes constructed by a plane-sweep algorithm [11]. Some methods use optical flow as part of the input to their network [6] or as one auxiliary task [7]. Different from these methods, our approach capitalizes on state-of-the-art optical flow methods to refine camera poses and generate depth proposals to improve the final depth estimation with a novel flow-to-depth layer. This flow-to-depth layer is built upon solving the classical triangulation problem for 3D depth estimation, and has the potential to generalize well to unseen environments.

One critical design in our model is a differentiable flow-to-depth layer that solves an epipolar geometry optimization problem. The flow-to-depth layer takes optical flow and camera poses as input and produces depth proposals. We show that our flow-to-depth layer does not only produce geometrically reliable depth maps (proposals) and their confidence maps but also helps refine the camera pose between video frames. At the end of our model, we have a fusion network that takes a target frame, depth proposals, and their confidence maps inferred from neighboring frames to produce the final depth maps. Note that the fusion network can additionally take the depth proposals generated by other depth estimation methods. For optical flow, we utilize the state-of-the-art optical flow methods that have gained significant progress [12]. To obtain the initial camera pose, we can use sensors such as IMU and GPS or apply odometry algorithms [13].

We conduct extensive experiments on the KITTI [14], ScanNet [15], and Waymo datasets [16]. The experiments show that our approach significantly outperforms state-of-the-art methods in depth estimation. Our controlled experiments indicate that the differentiable flow-to-depth layers in our model significantly improve the overall accuracy of video depth estimation by refining camera poses and generating depth proposals. To our surprise, our model trained on KITTI can generalize well to the unseen Waymo dataset while other methods do not. We believe the reason for the strong generalization capability of our model is that we solve for the depth proposals based on solving traditional triangulation problems rather than memorizing visual content. In summary, the main contributions of our work are as follows.

- We present a novel framework with differentiable flow-to-depth layers for video depth estimation. The flow-to-depth layer refines camera poses and generates depth proposals by solving a triangulation problem between two video frames.
- A depth fusion network can merge the depth proposals from the flow-depth-layer to produce the final depth maps. The depth fusion network can optionally take the depth maps generated by other methods to improve the performance further.
- We conduct thorough experiments on monocular depth estimation and show that our approach produces more accurate depth maps than contemporaneous methods do. Our model also demonstrates stronger generalization capability across datasets.

Jiaxin Xie (jxieax@connect.ust.hk) and Chenyang Lei (cleiaa@ust.hk) are with the Department of Computer Science and Engineering, HKUST. Zhuwen Li (lzhuwen@gmail.com) is with Nuro Inc. Li Erran Li (erranlli@gmail.com) is with Alexa AI, this work was done prior to joining Amazon. Qifeng Chen (cqf@ust.hk) is with the Department of Computer Science and Engineering and the Department of Electronic and Computer Enginnering, HKUST.

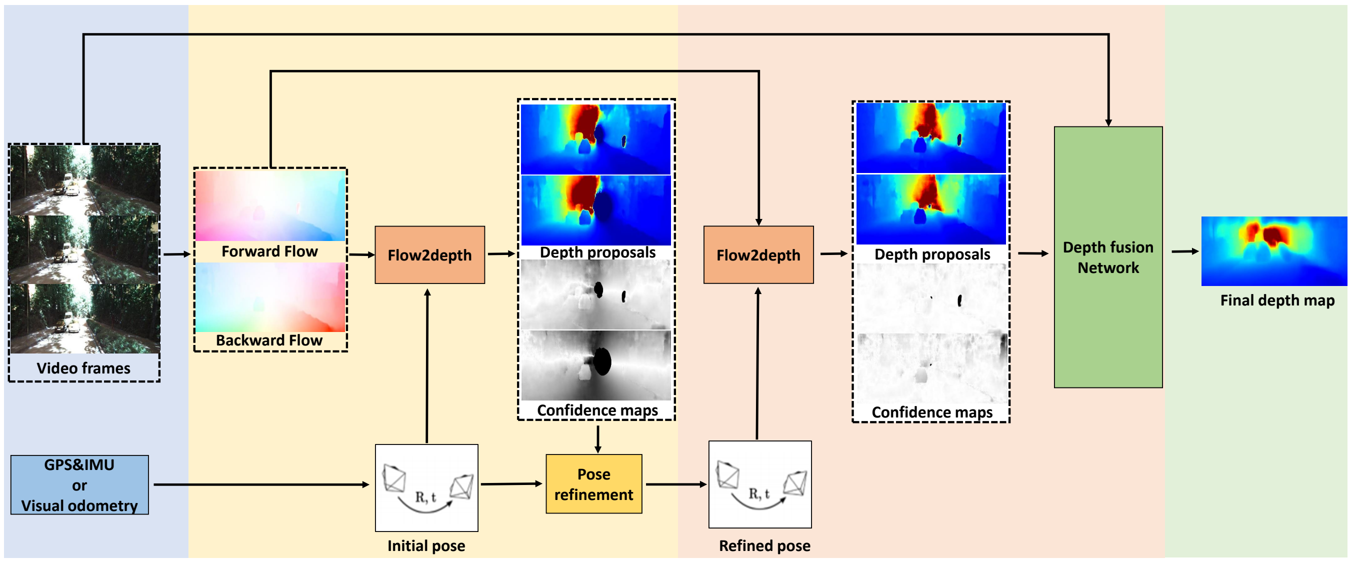

Fig. 1. The architecture of our overall framework. First, we estimate the optical flow from video frames and obtain initial camera poses from GPS and IMU or applying odometry algorithms. Second, the initial camera poses are refined by maximizing the sum of per-pixel confidence in the pose refinement module. Third, we can generate depth proposals and confidence maps with refined camera poses through the flow-to-depth layer. Finally, we obtain the final depth map by a depth fusion network that fuses the given depth proposals, confidence maps, and the target frame.

II. Related Work

In the literature, there is a large body of work on depth estimation from images. The settings can vary from a single image, binocular stereo, temporal sequences to discrete multiple views. We briefly review them below.

A. Single Image Depth

Early work in this line can be traced back to Saxena et al. [1] and Hoiem et al. [17]. The previous work learns to predict depth from single images using a discriminatively-trained Markov Random Field (MRF), while the later one classifies image pixels into different geometric regions, which can then be used to infer shapes. Most recently, with the success of deep learning, several works start to train deep convolutional neural networks to directly regress raw pixels to depth values [3]–[5], [18]. Our work is fundamentally different from these approaches in that we take geometrical constraints between video frames into consideration, which is suitable for various data and these single image approaches only learn distribution from the training images. Meanwhile, we also only need a single camera, the same device requirement as single image methods.

B. Binocular Stereo Depth

Depth estimation has been extensively exploited in the paired stereo setting, and the original problem is usually reformulated as a matching problem [19]. Thus, traditional stereo approaches [20], [21] often suffer from correspondence matching ambiguity in regions such as textureless areas, reflective surfaces, and repetitive patterns. Most recently, deep learning has also shown its success in stereo matching [22], [23]. The state-of-the-art approaches [24]–[27] usually construct a 3D cost volume and perform 3D convolutions on it. Along with this direction, improvements have been made by pyramid [25], semantic segmentation [26], learned affinity propagation [27], etc. The stereo pair setting usually generates an accurate depth and naturally adapts to dynamic scenes. However, the calibrated stereo camera is not ubiquitous in the real world. Compared to these approaches, our work focus on the monocular setting.

C. Depth from Video

Depth from video becomes a popular research problem recently, and the related work includes [6], [8], [10], [28]. Both [8] and [28] explicitly models the motion of moving objects, but their complex motion model takes lots of iterations to optimize. Instead of modeling moving objects accurately, we regard them as low-confident areas to train the network to do depth interpolation or inpainting in these areas. DeMoN [6] also proposes a neural architecture that alternates optical flow estimation and the estimation of camera motion and depth. Different from DeMON, our rigid depth and camera poses are more reliable since they are calculated via epipolar geometry and not directly predicted from neural networks.

D. Multi-view Reconstruction

Multi-view stereo (MVS) reconstructs 3D shapes from a number of discrete images, which is a core computer vision problem that has been studied for decades. Conventional MVS algorithms [29] perform 3D reconstruction by optimizing photometric consistency with handcrafted error functions to measure the similarity between patches. Similar to traditional stereo pair methods, these algorithms cannot handle poorly textured regions and reflective surfaces where photometric

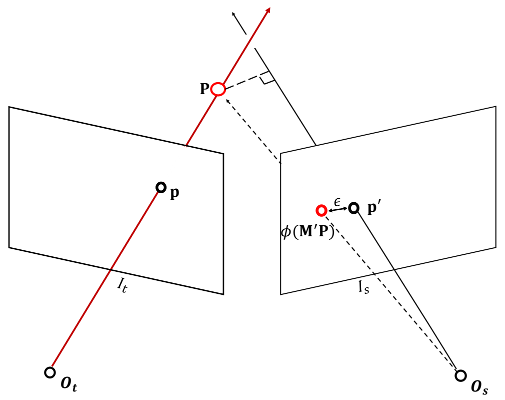

Fig. 2. The illustration of generating a depth proposal from optical flow. $\mathbf{p}$ and $\mathbf{p}'$ are corresponding points given by the optical flow. The objective of the flow-to-depth layer is to find an optimal $\mathbf{P}$ that minimizes the reprojection error ϵ.

consistency is unreliable. Recent deep learning methods [30]–[32] take the plane-sweep volume of deep features as input and produce depth maps for the reference images. The most significant difference between these cost volume-based methods and our approach is that we incorporate the multi-view geometry constraint via the flow-to-depth layer, which is new and effective.

III. Our Approach

Given a sequence of frames $\{I_1, ..., I_N\}$ from a monocular video, our objective is to predict the depth map of every video frame by utilizing the frames around it. The input to our model includes the target frame I_t, its neighboring frames $\{I_s\}$ and the initial camera pose transformations $\{T_{t,s}\}$ between I_t and $\{I_s\}$, which can be obtained from GPS and IMU, or by applying a visual odometry algorithm [33].

Fig. 1 illustrates the overall architecture of our proposed model, which consists of three critical components. The first part is the novel differentiable flow-to-depth layer. It takes optical flow and a camera pose as input and estimates rigid depth by triangulation in 3D. The layer produces both depth proposal map $D_{t,s}$ and confidence map $C_{t,s}$ for the target frame by solving the epipolar geometry problem using a least-squares method.

The second part is the camera pose refinement module. The initial relative camera pose $T_{t,s}$ may not be highly accurate due to noisy sensor outputs from GPS and IMU, or imperfect visual odometry algorithms. Since the flow-to-depth layer is differentiable, we can use it to backpropagate the gradients from the confidence map to the initial camera pose and refine the initial camera pose by maximizing the sum of per-pixel confidence. Our experimental results show that the pose refinement module significantly improves performance.

The last one is the depth fusion network. It takes the target frame, depth proposals, and confidence maps to generate the final depth map D. The intuition behind such a depth fusion network is that, for regions with high confidence, the network can directly use the provided depth values; otherwise, the network will perform depth interpolation or inpainting. Note that we also provide the target frame as an additional input to the depth fusion network, which provides strong image priors for inpainting the regions with low confidence. We find that utilizing depth proposals along with their confidence maps greatly improves the depth estimation quality.

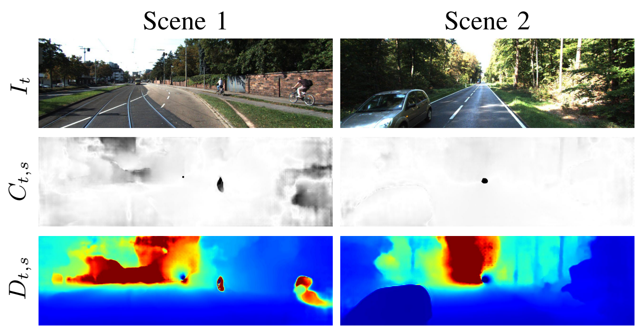

Fig. 3. The confidence maps (the second row) and depth proposals (the third row) generated by the flow-to-depth layer on the KITTI dataset. For the confidence maps, darker areas indicate lower confidence. For the depth proposals, blue areas indicate small depth values.

A. Flow-to-depth Layers

Parallax can appear between two video frames because of camera motion. We utilize this parallax to improve monocular depth estimation by introducing a differentiable flow-to-depth layer.

a) Depth proposals: Consider the depth estimation problem for a target frame I_t. Given a nearby source frame I_s, we leverage optical flow and relative camera pose between I_t and I_s to generate a depth proposal $D_{t,s}$. Fig. 2 illustrates configuration of our problem. With homogeneous coordinates, we consider a 3D point $\mathbf{P} = [x, y, d, 1]^T$ and its corresponding pixels in I_t and I_s as $\mathbf{p} = [u, v, 1]^T, \mathbf{p}' = [u', v', 1]^T$. Given $\mathbf{p}$ and $\mathbf{p}'$, we solve for an optimal $\mathbf{P}$ that minimizes the reprojection error. Let the world coordinate system be the camera coordinate system of I_t. Suppose $\mathbf{M}^{'}$ is the camera matrix for I_s, and $\mathbf{K}$ is the intrinsic matrix for I_t. In the following, we use numbers in subscript to slice vectors and matrices, and use comma to separate dimensions. Then we have

$$\mathbf{p} = \mathbf{K}\mathbf{P_{1:3}}, \mathbf{p}' = \mathbf{M}'\mathbf{P}. \tag{1}$$

Our reprojection error is formulated as:

$$\epsilon(d) = \|\phi(\mathbf{M}' \begin{bmatrix} d\mathbf{K}^{-1}\mathbf{p} \\ 1 \end{bmatrix}) - \mathbf{p}'\|, \tag{2}$$

where d is the depth of $\mathbf{P}$, and $\phi(\mathbf{x}) = \frac{\mathbf{x}}{\mathbf{x_3}}, \mathbf{x} \in \mathcal{R}^3$. For notation convenience, we denote $\mathbf{a} = \mathbf{M}'_{1:3,1:3}\mathbf{K}^{-1}\mathbf{p}, \mathbf{b} = \mathbf{M}'_{1:3,4}$. Then the optimal d^* minimizing $\epsilon(d)$ can be computed in a closed form:

$$d^* = \arg\min_d \|\phi(d\mathbf{a} + \mathbf{b}) - \mathbf{p}'\| = \frac{1}{\mathbf{m}^T\mathbf{m}}\mathbf{m}^T\mathbf{n}, \tag{3}$$

where $\mathbf{m} = \mathbf{a}_{1:2} - \mathbf{a}_3\mathbf{p}'_{1:2}, \mathbf{n} = \mathbf{b}_3\mathbf{p}'_{1:2} - \mathbf{b}_{1:2}$.

We can use optical flow algorithms, such as PWC-Net [12], to find dense pixel-wise correspondences between I_t, I_s, then solve for the optimal depth at each pixel location using Equation (3). Since this procedure is differentiable, we implement it as a flow-to-depth layer to enable end-to-end training.

b) Confidence maps: The reprojection error ϵ can serve as a confidence measure for the computed depth: the larger the reprojection error is, the more likely the depth is prone to error. We obtain a confidence map $C_{t,s}$ by converting ϵ into confidence using $\exp\left(-\frac{\epsilon}{\sigma}\right)$, where σ is a normalization constant. We set $\sigma = 20$ in experiments. Moreover, if the computed d is negative, we set its confidence to zero. Fig. 3 shows our depth proposals and the corresponding confidence maps.

B. Camera Pose Refinement

The quality of our depth proposals highly depends on the quality of camera poses. In practice, we can obtain an initial camera pose from sensors such as GPS and IMU, but the initial camera pose is not highly accurate due to sensor noise. To improve the accuracy of the camera pose, we utilize our flow-to-depth layer to refine the camera pose.

We can refine camera poses by building the relationship between camera poses and confidences map through the differentiable flow-to-depth layer. Typically, a good camera pose should lead to a large confidence map. We define a maximization objective function to improve the camera pose $T_{t,s}$:

$$L(T_{t,s}) = \sum_{p \in \mathcal{S}} C_{t,s}(p), \tag{4}$$

where S is the set of pixels with positive depth in the depth proposal. We exclude those pixels with negative depth as depth should be not negative. The objective is designed to maximize the sum of the confidence on each pixel with positive depth.

To minimize the objective function in (4), we use the L-BFGS-B optimizer [34], and set bounds $[-\pi, \pi]$ for rotation in M. Note that we are able to compute the gradients on the camera pose because the flow-to-depth layer is differentiable. We evaluate the performance with and without the pose refinement, and experiments show that the refinement can significantly improve the depth estimation.

C. Depth Fusion

For each pair of the target frame and the source frame, we can generate a depth proposal and a confidence map for the target frame. Then we can make use of depth proposals and confidence maps to produce a high-quality final depth map. Our depth fusion network is designed to merge them and perform refinement as needed. Compared to single image depth estimation methods, our approach has the benefits that the model can take advantage of the depth proposals and their confidence maps for better depth estimation.

As shown in Fig. 1, we concatenate the target frame I_t, depth proposals $D_{t,s}$, confidence maps $C_{t,s}$ as input to the depth fusion network. The output of the depth fusion network is the final depth map D. Besides the depth proposals and confidence maps computed by our flow-to-depth layer, our depth fusion network can also take the depth proposals generated by other methods to improve the estimation accuracy. We train our depth fusion network with provided ground-truth depth maps in a supervised fashion.

a) Loss function: Our depth loss is defined over each pixel p with ground-truth depth:

$$L_{depth} = \sum_{p} || \log D_p - \log D^*_p ||, \tag{5}$$

where D^* is the ground-truth depth map. We define the depth loss in the log domain rather than the linear domain to prevent distant pixels from dominating the loss.

We also use a smoothness loss by imposing smoothness regularization on the output disparity map (inverse of the depth map). The smoothness loss is defined as

$$L_{smooth} = \sum_{p} \nabla^2 \frac{1}{D_p}, \tag{6}$$

where ∇^2 is the Laplace operator.

The total loss for the depth fusion network is

$$L_{fusion} = \lambda_d L_{depth} + \lambda_s L_{smooth}, \tag{7}$$

where $\lambda_d = 1$ and $\lambda_s = 0.5$.

b) Network architecture: Our depth fusion network adopts the single view depth network in SfMLearner [9]. It is an encoder-decoder architecture with skip connections and multi-scale prediction.

IV. Experiments

A. Implementation

For the depth estimation of the target frame I_t, we use I_{t-k_1} and I_{t+k_2} as the source frames. Since depth proposals have poor results when the camera translation between I_t and I_s (defined as $\|O_s - O_t\|$) is too small, we search for the smallest k_1 that satisfies $\|O_{t-k_1} - O_t\| > T$ where T is a threshold. For the KITTI [14] and Waymo datasets [16], T is 80cm. For the ScanNet dataset [15], T is 12cm. We perform similar search for k_2.

To train the model, we use the Adam optimizer [36] with the learning rate of 0.0001, batch size of 4, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We use full-resolution video frames and ground-truth depth maps during training and evaluation. With one Nvidia 1080 Ti, our model trained on KITTI Eigen split converges after 25 epochs. Each epoch takes 4 hours.

B. Datasets

We conduct experiments on three datasets: the KITTI dataset [14], the ScanNet dataset [15], and the Waymo dataset [16].

The KITTI dataset contains outdoor images with depth maps projected from point clouds and also provides camera poses calculated from GPS and IMU. To compare with different previous works, we train our method in two different splits. One is the Eigen split proposed by Eigen et al. [3],

TABLE I
QUANTITATIVE EVALUATION ON THE KITTI DATASET.

Method	Type	split	**abs rel** $\downarrow$	**sq rel** $\downarrow$	**rms** $\downarrow$	**log rms** $\downarrow$	**irmse** $\downarrow$	**SIlog** $\downarrow$	$\boldsymbol{\delta_1}$ $\uparrow$	$\boldsymbol{\delta_2}$ $\uparrow$	$\boldsymbol{\delta_3}$ $\uparrow$
Eigen et al. [3] coarse	supervised	Eigen	0.194	1.531	7.216	0.273	-	-	0.679	0.897	0.967
Eigen et al. [3] fine	supervised	Eigen	0.190	1.515	7.156	0.270	-	-	0.692	0.899	0.967
GeoNet [7]	unsupervised+video	Eigen	0.155	1.297	5.857	0.233	0.018	0.229	0.793	0.931	0.973
Godard et al. [18]	unsupervised+stereo	Eigen	0.150	1.329	5.806	0.231	0.019	0.227	0.810	0.933	0.971
Kuznietsov et al. [35]	semi-supervised+stereo	Eigen	0.110	0.708	4.312	0.172	0.014	0.169	0.878	0.964	0.987
DORN [4]	supervised	Eigen	0.102	0.592	3.837	0.162	0.015	0.158	0.898	0.967	0.986
Ours	supervised+video	Eigen	**0.081**	**0.488**	**3.651**	**0.146**	**0.012**	**0.144**	**0.912**	**0.970**	**0.988**
NeuralRGBD [11]	supervised+video	Uhrig	0.105	0.532	3.299	0.150	0.013	0.144	0.887	0.972	0.990
Ours	supervised+video	Uhrig	**0.071**	**0.338**	**2.537**	**0.116**	**0.010**	**0.112**	**0.938**	**0.979**	**0.992**

in which the train set contains 33 video scenes, the test set consists of 697 images extracted from 28 video scenes, and ground-truth depth maps projected from single-frame point clouds. Another one is the Uhrig split [37] that came with the KITTI depth prediction and completion benchmark. It has 138 training scenes and 13 validation scenes. We randomly sample 50 images from every video scene in the validation set and get a test set that consists of 650 images. Meanwhile, this split provides denser ground-truth depth maps, which are accumulated by 11 consecutive frames point clouds. Since different video sequences in KITTI may have different resolutions, we resize all the training frames to 376×1241.

The ScanNet dataset is an RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, depth maps, and instance-level semantic segmentations. For the train set and test set, we follow the instructions of the Robust Vision Challenge 2018 Workshop at CVPR 2018.

The Waymo Open Dataset is a recently released autonomous driving dataset. It contains LiDAR and camera data from 1,000 video segments, split into training set and validation set. We randomly sample 5 images from every daytime validation video segment and obtain a total of 784 test images to do cross dataset experiment.

C. Baselines

In the KITTI Eigen split, we compare our method with several state-of-the-art depth estimation approaches: DORN [4], Kuznietsov et al. [35], Godard et al. [18], GeoNet [7], and Eigen et al. [3].

In the KITTI Uhrig split, we compare our method against state-of-the-art video depth estimation approaches: NeuralRGBD [11]. We re-train NeuralRGBD [11] in the Eigen split, but its results are poor. To have a fair comparison, we also train our method in the Uhrig split and compare it with the results by the pre-trained model of NeuralRGBD [11].

In the ScanNet and Waymo datasets, We carefully select two deep learning based methods for comparisons. For supervised single image depth estimation approaches, we choose DORN [4], which is state of the art. For supervised video depth estimation methods, we choose NeuralRGBD [11] that is highly related to our work.

D. Results

We conduct extensive experiments to evaluate the performance of our method and state-of-the-art methods. Our method is able to produce more accurate depth maps and outperforms the contemporaneous methods on most evaluation metrics. In addition, our method is more robust and shows great generalization ability.

a) Quantitative Evaluation: On the KITTI dataset, we train our model in the Eigen split and the Uhrig split separately. Table I summarizes the quantitative evaluation results of our method and other state-of-the-art baselines in both splits. For a fair comparison, we use exactly the same evaluation code provided by Zhou et al. [9] to evaluate all the methods except Eigen et al. [3]. We directly use the results reported on Eigen et al. because the provided source code only produces low-resolution 28×144 or 27×142 depth maps, but we evaluate on full-resolution depth maps. The results are much worse if we directly upsample their output low-resolution depth maps.

Regarding the metrics, we include widely the used ones from prior work [4], [7], and the metrics used by the KITTI depth estimation benchmark. They are **abs rel**: absolute relative error; **sq rel**: square relative error; **rms**: root mean square; **log rms**: log root mean square; **irmse**: inverse root mean square error; **SIlog**: scale-invariant logarithmic error; $\boldsymbol{\delta_i}$: the percentage of pixels with relative depth error $\delta < 1.25^i$. The $\downarrow$ indicates the lower the better, the$\uparrow$ does the opposite.

b) Qualitative evaluation: As shown in Table I, our method outperforms state-of-the-art methods in both splits. In the Eigen split, our method has outperformed several state-of-the-art depth estimation methods by a large margin. In the Uhrig split, our model additionally takes the depth maps generated by NeuralRGBD [11] for depth fusion, and has about 20-30% improvement in most metrics.

Table II compares our method with two representative approaches on the ScanNet dataset. As shown in Table II, our

TABLE II
QUANTITATIVE EVALUATION ON THE SCANNET DATASET.

Method	Type	**abs rel** ↓	**sq rel** ↓	**rms** ↓	**log rms** ↓	**irmse** ↓	**SIlog** ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
DORN [4]	supervised	0.096	0.033	0.217	0.127	0.099	0.120	0.907	0.981	**0.996**
NeuralRGBD [11]	supervised	0.097	0.050	0.249	0.132	0.093	0.126	0.906	0.975	0.993
Ours	supervised	**0.076**	**0.029**	**0.199**	**0.108**	**0.077**	**0.103**	**0.933**	**0.984**	**0.996**

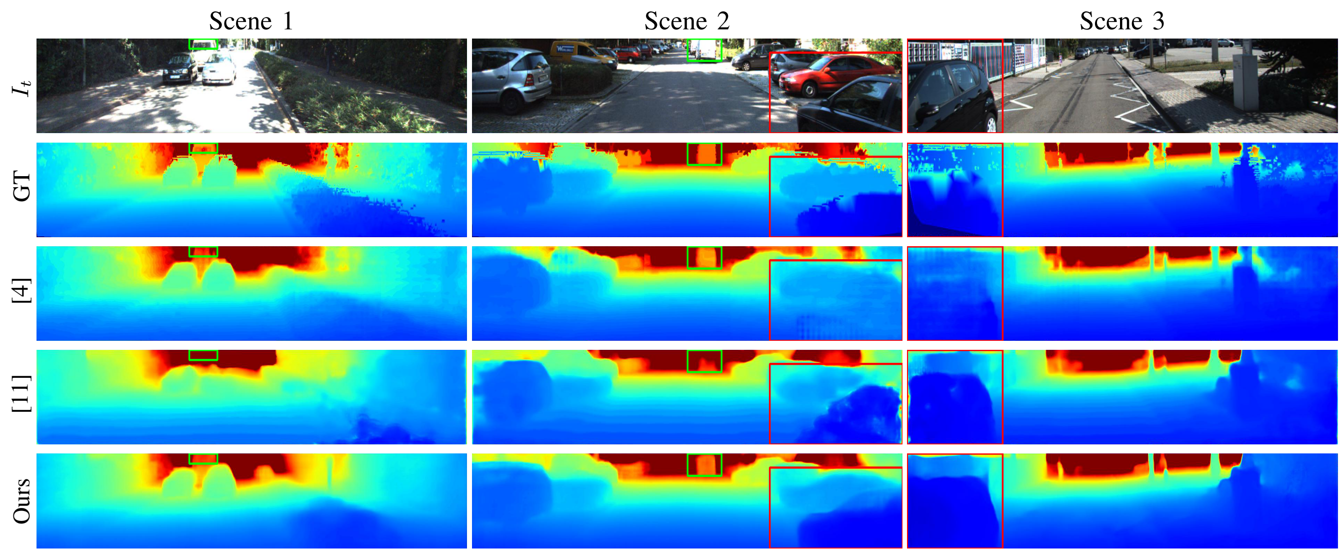

Fig. 4. The qualitative comparisons of DORN [4], NeuralRGBD [11], and ours on the KITTI dataset. The ground-truth depth map is interpolated from sparse measurements for visualization.

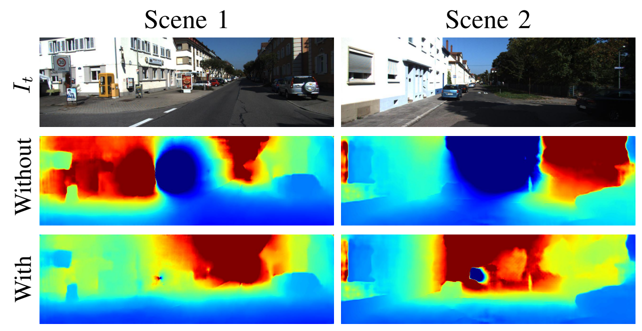

Fig. 5. The visualization of depth proposals with (the third row) and without (the second row) camera pose refinement. The quantitative comparison is shown on TABLE III.

TABLE III
QUANTITATIVE EVALUATION OF ABLATION STUDY.

Method	**abs rel** ↓	**sq rel** ↓	**rms** ↓	**SIlog** ↓	δ_1 ↑
RGB frames only	0.120	0.817	4.690	0.189	0.858
Ours (w/o refinement)	0.085	0.522	3.767	0.148	0.906
Ours (w/ refinement)	**0.081**	**0.488**	**3.651**	**0.144**	**0.912**

method performs better on the first nine metrics and achieves comparable performance with DORN [4] on metric δ_3. The depth proposals we used are the same as the model in the KITTI Uhrig split. Besides depth proposals generated by the flow-to-depth layer, the result of NeuralRGBD [11] serves as a depth proposal on this model, which speeds up the training process and improves performance.

Fig. 4 illustrates some qualitative results on the KITTI dataset. As the green boxes in Scene 1 and Scene 2 show, NeuralRGBD [11] misses the top of a van behind two cars in Scene 1 and only estimates the bottom part of a truck in Scene 2. Meanwhile, Both DORN [4] and our results include the whole van and truck. As we utilize RGB image priors, the depth values on the same object should be continuous or constant. Compared to NeuralRGBD, our depth fusion network can take advantage of the target frame when computed rigid depth is not reliable.

As the red boxes in Scene 2 and Scene 3 show, DORN [4] produces a blurry depth map that can not differentiate object boundaries. In contrast, both NeuralRGBD [11] and our method produce reasonably sharper results. Note that a common characteristic of NeuralRGBD [11] and ours is that we both use geometrical information. The sharper boundaries benefit from our flow-to-depth layer.

Fig. 6 shows the comparisons on the ScanNet dataset, where the first row shows depth maps and the second row shows error maps. As shown in the error maps, we produce depths with lower error compared to NeuralRGBD [11] and DORN [4]. Our output depth map is less noisy and more complete.

TABLE IV
QUANTITATIVE EVALUATION ON THE WAYMO DATASET.

Method	Type	**abs rel** $\downarrow$	**sq rel** $\downarrow$	**rms** $\downarrow$	**log rms** $\downarrow$	**irmse** $\downarrow$	**SIlog** $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
SfMLearner [9]	unsupervised	0.514	7.878	16.029	0.587	0.031	0.579	0.256	0.487	0.703
DORN [4]	cross dataset	0.389	5.056	12.432	0.451	0.024	0.442	0.353	0.660	0.867
NeuralRGBD [11]	cross dataset	0.177	2.646	9.891	0.402	0.072	0.396	0.790	0.921	0.958
Ours	cross dataset	**0.150**	**1.691**	**6.773**	**0.222**	**0.013**	**0.211**	**0.804**	**0.924**	**0.966**

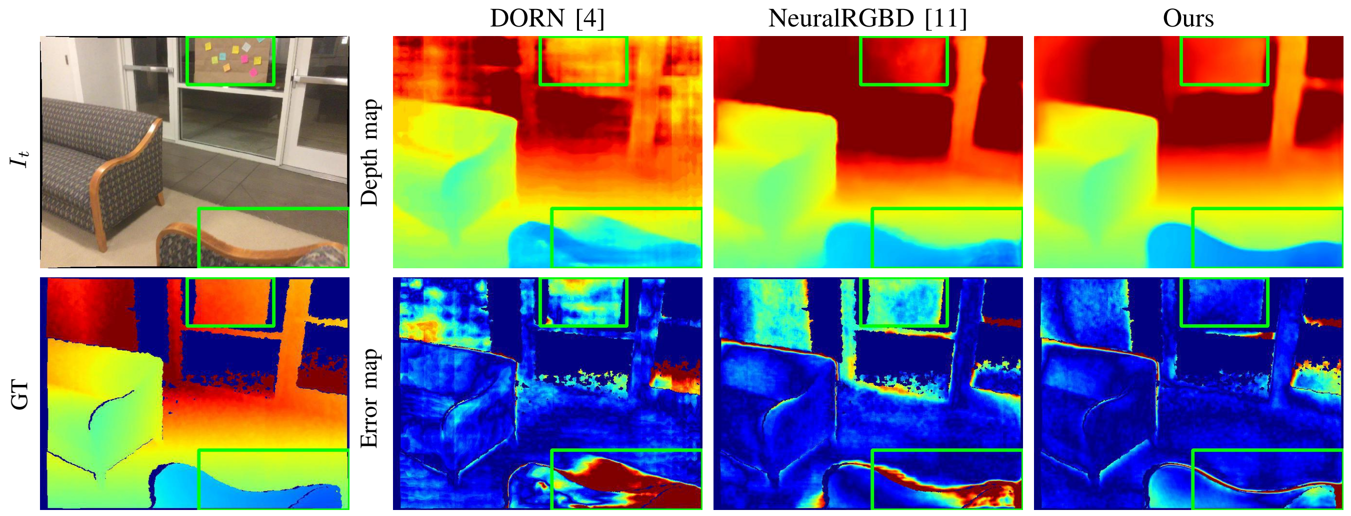

Fig. 6. The qualitative comparisons between DORN [4], NeuralRGBD [11], and ours on the ScanNet dataset. For the error maps, blue areas indicate low errors and red areas indicate high errors.

c) Ablation study: The accuracy of relative camera poses can significantly affect the video-based depth estimation performance. Fig. 5 shows depth proposals generated with and without pose refinement in two extreme examples. In the second row, without pose refinement, the initial camera pose produces poor depth proposals that have a vast region of negative depths. After pose refinement, in the third row, we can get depth proposals with higher confidence. We show a quantitative comparison of models with and without pose refinement in Table III. Our camera pose refinement improves these metrics by about 3 to 6 percent.

We also have an ablation experiment by training the depth fusion network to estimate depth directly from the target frame and source frames. The results of this experiment are shown in the first row of Table III, our complete model performs much better than this ablated model and this comparison validates the strength of flow-to-depth layer in our model.

d) Cross-dataset evaluation: Table IV reports the quantitative results of cross dataset evaluation on the Waymo dataset. Our model (trained on KITTI and test on Waymo) suffers less performance degeneration than NeuralRGBD [11] and DORN [4] in the cross-dataset evaluation and completely surpass SfMLearner [9] which is trained on the Waymo dataset in an unsupervised fashion. Fig. 7 shows the visual results of NeuralRGBD and our model on cross-dataset tasks. These results suggest that our depth proposals can often preserve object boundaries in the estimated depth maps, even on the cross-dataset results.

V. CONCLUSION

We have presented a video depth estimation method that builds upon a novel flow-to-depth layer. This layer can help refine camera poses and generate depth proposals. Beyond the depth proposals computed from the flow-to-depth layer, depth maps estimated by other methods can also serve as depth proposals in our model. In the end, a depth fusion network fuses all depth proposals to generate a final depth map. The experiments show that our presented model outperforms all other state-of-the-art depth estimation methods on the KITTI dataset, ScanNet dataset, and shows excellent generalization ability on the Waymo dataset. We hope our model can be a practical tool for other researchers and inspire more future work on monocular video depth estimation.

REFERENCES

[1] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in *NeurIPS*, 2005.
[2] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *ICCV*, 2015.
[3] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *NIPS*, 2014.
[4] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep Ordinal Regression Network for Monocular Depth Estimation,” in *CVPR*, 2018.
[5] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *3DV*, 2016.

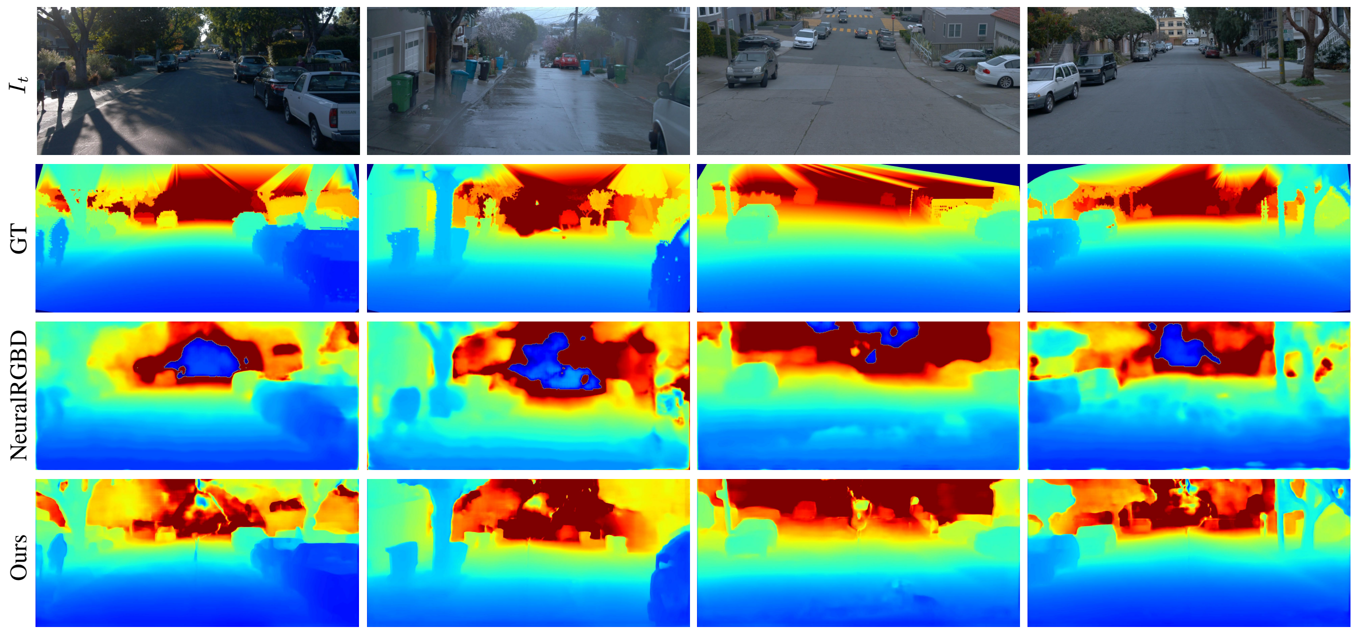

Fig. 7. The cross-dataset comparisons between NeuralRGBD [11] and ours on the Waymo dataset. All the depth maps are produced by models trained on the KITTI dataset.

[6] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *CVPR*, 2017.

[7] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018.

[8] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *CVPR*, 2016.

[9] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[10] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," *CoRR*, vol. abs/1812.04605, 2018. [Online]. Available: http://arxiv.org/abs/1812.04605

[11] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural rgb(r)d sensing: Depth and uncertainty from a video camera," in *CVPR*, June 2019.

[12] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *CVPR*, 2018.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *TPAMI*, 2018.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*, 2012.

[15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017.

[16] "Waymo open dataset: An autonomous driving dataset," 2019.

[17] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *ICCV*, 2005.

[18] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[19] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, 2002.

[20] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *CVPR*, 2005.

[21] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *TPAMI*, 2013.

[22] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *JMLR*, 2016.

[23] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *CVPR*, 2015.

[24] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*, 2017.

[25] J. Chang and Y. Chen, "Pyramid stereo matching network," in *CVPR*, 2018.

[26] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "Segstereo: Exploiting semantic information for disparity estimation," in *ECCV*, 2018.

[27] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *ECCV*, 2018.

[28] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *AAAI*, 2019.

[29] A. Harltey and A. Zisserman, *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006.

[30] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *ECCV*, 2018.

[31] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *ECCV*, 2018.

[32] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon, "Dpsnet: End-to-end deep plane sweep stereo," in *ICLR*, 2019.

[33] D. Nistér, O. Naroditsky, and J. R. Bergen, "Visual odometry," in *CVPR*, 2004.

[34] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *TOMS*, 1997.

[35] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *CVPR*, July 2017.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[37] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *3DV*, 2017.