3D LiDAR and Stereo Fusion using Stereo Matching Network with Conditional Cost Volume Normalization

Tsun-Hsuan Wang, Hou-Ning Hu, Chieh Hubert Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, Min Sun

Abstract— The complementary characteristics of active and passive depth sensing techniques motivate the fusion of the Li-DAR sensor and stereo camera for improved depth perception. Instead of directly fusing estimated depths across LiDAR and stereo modalities, we take advantages of the stereo matching network with two enhanced techniques: Input Fusion and Conditional Cost Volume Normalization (CCVNorm) on the LiDAR information. The proposed framework is generic and closely integrated with the cost volume component that is commonly utilized in stereo matching neural networks. We experimentally verify the efficacy and robustness of our method on the KITTI Stereo and Depth Completion datasets, obtaining favorable performance against various fusion strategies. Moreover, we demonstrate that, with a hierarchical extension of CCVNorm, the proposed method brings only slight overhead to the stereo matching network in terms of computation time and model size.

I. INTRODUCTION

The accurate 3D perception has been desired since its vital role in numerous tasks of robotics and computer vision, such as autonomous driving, localization and mapping, path planning, and 3D reconstruction. Various techniques have been proposed to obtain depth estimation, ranging from active sensing sensors (e.g., RGB-D cameras and 3D LiDAR scanners) to passive sensing ones (e.g., stereo cameras). We observe that these sensors all have their own pros and cons, in which none of them perform well on all practical scenarios. For instance, RGB-D sensor is confined to its short-range depth acquisition and thereby 3D LiDAR is a common alternative in the challenging outdoor environment. However, 3D LiDARs are much more expensive and only provide sparse 3D depth estimates. In contrast, a stereo camera is able to obtain denser depth map based on stereo matching algorithms but is typically incapable of producing reliable matches in regions with repetitive patterns, homogeneous appearance, or large illumination change.

Thanks to the complementary characteristic across different sensors, several works [1][2] have studied how to fuse multiple modalities in order to provide more accurate and denser depth estimation. In this paper, we consider the fusion of passive stereo camera and active 3D LiDAR sensor, which is a practical and popular choice. Existing works along this research direction mainly investigate the output-level combination of the dense depth from stereo matching with the sparse measurement from 3D LiDAR. However, rich information provided in stereo images is thus not well utilized in the procedure of fusion. In order to address this issue, we propose to study the design choices for more closely integrating the 3D LiDAR information into the process of stereo matching methods (illustrated in Fig. 1).
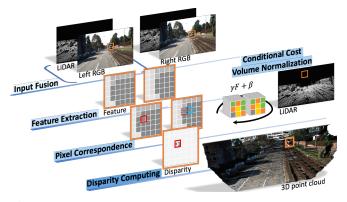

Fig. 1: Illustration of our method for 3D LiDAR and stereo fusion. The high-level concept of stereo matching pipeline involves 2D feature extraction from the stereo pair, obtaining pixel correspondence, and finally disparity computation. In this paper, we present (1) Input Fusion and (2) Conditional Cost Volume Normalization that are closely integrated with stereo matching networks. By leveraging the complementary nature of LiDAR and stereo modalities, our model produces high-precision disparity estimation.

The motivation that drives us toward this direction is an observation that typical stereo matching algorithms usually suffer from having ambiguous pixel correspondences across stereo pairs, and thereby 3D LiDAR depth points are able to help reduce the search space of matching and resolve ambiguities.

As depth points from 3D LiDAR sensors are sparse, it is not straightforward to simply treat them as additional features connected to each pixel location of a stereo pair during performing stereo matching. Instead, we focus on facilitating sparse points to regularize higher-level feature representations in deep learning-based stereo matching. Recent state-of-the-arts on deep models of stereo matching are composed of two main components: matching cost computation [3][4] and cost volume regularization [5][6][7][8], where the former basically extracts the deep representation of image patches and the latter builds up the search space to aggregate all potential matches across stereo images with further regularization (e.g., 3D CNN) for predicting the final depth estimate.

Being aligned with these two components, we extend the stereo matching network by proposing two techniques: (1) **Input Fusion** to incorporate the geometric information from sparse LiDAR depth with the RGB images for learning joint feature representations, and (2) **CCVNorm** (**C**onditional **C**ost **V**olume **Norm**alization) to adaptively regularize cost volume optimization in dependence on LiDAR measurements. It is worth noting that our proposed techniques have

little dependency on particular network architectures but only relies on a commonly-utilized cost volume component, thus having more flexibility to be adapted into different models. Extensive experiments are conducted on the KITTI Stereo 2015 Dataset [9] and the KITTI Depth Completion Dataset [10] to evaluate the effectiveness of our proposed method. In addition, we perform ablation study on different variants of our approach in terms of performance, model size and computation time. Finally, we analyze how our method exploits the additional sparse sensory inputs and provide qualitative comparisons with other fusion schemes to further highlight the strengths and merits of our method.

II. RELATED WORKS

Stereo Matching. Stereo matching has been a fundamental problem in computer vision. In general, a typical stereo matching algorithm can be summarized into a four-stage pipeline [11], consisting of *matching cost computation*, *cost support aggregation*, *cost volume regularization*, and *disparity refinement*. Even when deep learning is introduced to stereo matching in recent years and brings a significant leap in performance of depth estimation, such design paradigm is still widely utilized. For instance, [3] and [4] propose to learn a feature representation for matching cost computation by using a deep Siamese network, and then adopt the classical semi-global matching (SGM) [12] to refine the disparity map. [13] and [5] further formulate the entire stereo matching pipeline as an end-to-end network, where the cost volume aggregation and regularization are modelled jointly by 3D convolutions. Moreover, [6] and [7] propose several network designs to better exploit multi-scale and context information. Built upon the powerful learning capacity of deep models, this paper aims to integrate LiDAR information into the procedure of stereo matching networks for a more efficient scheme of fusion.

RGB Imagery and LiDAR Fusion. Sensor fusion of RGB imagery and LiDAR data obtain more attention in virtue of its practicability and performance for depth perception. Two different settings are explored by several prior works: LiDAR fused with a monocular image or stereo ones. As the depth estimation from a single image is typically based on a regression from pixels, which is inherently unreliable and ambiguous, most of the recent monocular-based works aim to achieve the completion on the sparse depth map obtained by LiDAR sensor with the help of rich information from RGB images [14][15][16][17][18][19], or refine the depth regression by having LiDAR data as a guidance [20][21].

On the other hand, since the stereo camera relies on the geometric correspondence across images of different viewing angles, its depth estimates are less ambiguous in terms of the absolute distance between objects in the scene and can be well aligned with the scale of 3D LiDAR measurements. This property of stereo camera makes it a practical choice to be fused with 3D LiDAR data in robotic applications, where the complementary characteristics of passive (stereo) and active (LiDAR) depth sensors are better utilized [22][23][24].

For instance, Maddern *et al.* [25] propose a probabilistic framework for fusing LiDAR data with stereo images to generate both the depth and uncertainty estimate. With the power of deep learning, Park *et al.* [1] utilize convolutional neural network (CNN) to incorporate sparse LiDAR depth into the estimation from SGM [12] of stereo matching. However, we argue that the sensor fusion directly applied to the depth outputs is not able to resolve the ambiguous correspondences existing in the procedure of stereo matching. Therefore, in this paper we advance to encode sparse LiDAR depth at earlier stages in stereo matching, i.e., matching cost computation and cost regularization, based on our proposed **CCVNorm** and **Input Fusion** techniques.

Conditional Batch Normalization. While the Batch Normalization layer improves network training via normalizing neural activations according to the statistics of each mini-batch, the Conditional Batch Normalization (CBN) operation instead learns to predict the normalization parameters (i.e., feature-wise affine transformation) in dependence on some conditional input. CBN has shown its generability in various application for coordinating different sources of information into joint learning. For instance, [26] and [27] utilize CBN to modulate imaging features by a linguistic embedding and successfully prove its efficacy for visual question answering. Perez *et al.* [28] further generalize the CBN idea and point out its connections to other conditioning mechanisms, such as concatenation [29], gating features [30], and hypernetworks [31]. Lin *et al.* [32] introduces CBN to a task of generating patches with spatial coordinates as conditions, which shares similar concept of modulating features by spatial-related information. In our proposed method for the fusion of stereo camera and LiDAR sensor, we adopt the mechanism of CBN to integrate LiDAR data into the cost volume regularization step of the stereo matching framework, not only because of its effectiveness but also the clear motivation on reducing the search space of matching for more reliable disparity estimation.

III. METHOD

As motivated above, we propose to fuse 3D LiDAR data into a stereo matching network by using two techniques: Input Fusion and CCVNorm. In the following, we will first describe the baseline stereo matching network, and then sequentially provide the details of our proposed techniques. Finally, we introduce a hierarchical extension of CCVNorm which is more efficient in terms of runtime and memory consumption. The overview of our proposed method is illustrated in Fig. 2.

A. Preliminaries of Stereo Matching Network

The end-to-end differentiable stereo matching network used in our proposed method, as shown in the bottom part of Fig. 2, is based on the work of GC-Net [5] and is composed of four primary components which are in line with the typical pipeline of stereo matching algorithms [11]. First, the deep feature extracted from a rectified left-right stereo pair is learned to compute the cost of stereo matching.
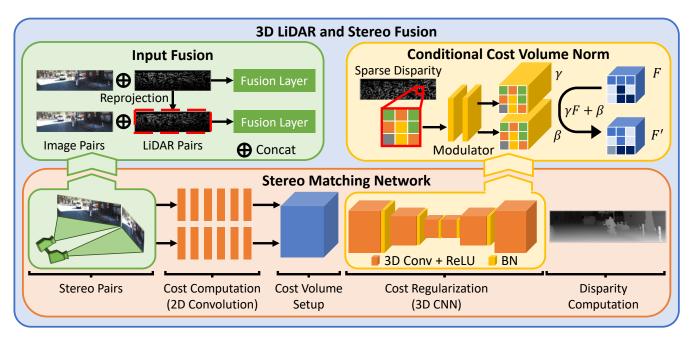
Fig. 2: Overview of our 3D LiDAR and stereo fusion framework. We introduce (1) **Input Fusion** that incorporates the geometric information from sparse LiDAR depth with the RGB images as the input for the Cost Computation phase to learn joint feature representations, and (2) **CCVNorm** that replaces batch normalization (BN) layer and modulates the cost volume features F with being conditioned on LiDAR data, in the Cost Regularization phase of stereo matching network. With the proposed two techniques, Disparity Computation phase yields disparity estimate of high-precision.

The representation with encoded context information acts as a similarity measurement that is more robust than simple photometric appearance, and thus it benefits the estimation of pixel matches across stereo images. A cost volume is then constructed by aggregating the deep features extracted from the left-image with their corresponding ones from the right-image across each disparity level, where the size of cost volume is 4-dimensional $C \times H \times W \times D$ (i.e., *feature size $\times$ height $\times$ width $\times$ disparity*). To be detailed, the cost volume actually includes all the potential matches across stereo images and hence serves as a search space of matching. Afterwards, a sequence of 3D convolutional operations (3D-CNN) is applied for cost volume regularization and the final disparity estimation is carried out by regression with respect to the output volume of 3D-CNN along the D dimension.

B. Input Fusion

In the cost computation stage of stereo matching network, both left and right images of a stereo pair are passed through layers of convolutions for extracting features. In order to enrich the representation by jointly reasoning on appearance and geometry information from RGB images and LiDAR data respectively, we propose Input Fusion that simply concatenates stereo images with their corresponding sparse LiDAR depth maps. Different from [14] that has explored a similar idea, for the setting of stereo and LiDAR fusion, we form the two sparse LiDAR depth maps corresponding to stereo images by reprojecting the LiDAR sweep to both left and right image coordinates with triangulation for converting depth values into disparity ones.

C. Conditional Cost Volume Normalization (CCVNorm)

In addition to Input Fusion, we propose to incorporate information of sparse LiDAR depth points into the cost regularization step (i.e., 3D-CNN) of stereo matching network, learning to reduce the search space of matching and resolve ambiguities. As inspired by Conditional Batch Normalization (CBN) [27][26], we propose CCVNorm (**Co**nditional **Co**st **Vo**lume **No**rmalization) to encode the sparse LiDAR information L^s into the features of 4D cost volume F of size $C \times H \times W \times D$. Given a mini-batch $\mathcal{B} = \{F_{i,\cdot,\cdot,\cdot,\cdot}\}_{i=1}^N$ composed of N examples, 3D Batch Normalization (BN) is defined at training time as follows:

$$F_{i,c,h,w,d}^{BN} = \gamma_c \frac{F_{i,c,h,w,d} - \mathbb{E}_{\mathcal{B}}[F_{\cdot,c,\cdot,\cdot,\cdot}]}{\sqrt{Var_{\mathcal{B}}[F_{\cdot,c,\cdot,\cdot,\cdot}] + \epsilon}} + \beta_c \quad (1)$$

where ϵ is a small constant for numerical stability and $\{\gamma_c, \beta_c\}$ are learnable BN parameters. When it comes to Conditional Batch Normalization, the new BN parameters $\{\gamma_{i,c}, \beta_{i,c}\}$ are defined as functions of conditional information L_i^s, for modulating the feature maps of cost volume in dependence on the given LiDAR data:

$$\gamma_{i,c} = g_c(L_i^s), \qquad \beta_{i,c} = h_c(L_i^s) \quad (2)$$

However, directly applying typical CBN to 3D-CNN in stereo matching networks could be problematic due to few considerations: (1) Different from previous works [27][26], the conditional input in our setting is a sparse map L^s with varying values across pixels, which implies that normalization parameters should be carried out pixel-wisely; (2) An alternative strategy is required to tackle the void information

contained in the sparse map L^s; (3) A valid value in $L^s_{h,w}$ should contribute differently to each disparity level of the cost volume.

Therefore, we introduce CCVNorm (as shown in bottom-left of Fig. 3) which better coordinates the 3D LiDAR information with the nature of cost volume to tackle the aforementioned issues:

$$F^{CCVNorm}_{i,c,h,w,d} = \gamma_{i,c,h,w,d} \frac{F_{i,c,h,w,d} - \mathbb{E}_{\mathcal{B}}[F_{\cdot,c,\cdot,\cdot,\cdot}]}{\sqrt{Var_{\mathcal{B}}[F_{\cdot,c,\cdot,\cdot,\cdot}] + \epsilon}} + \beta_{i,c,h,w,d}$$

$$\gamma_{i,c,h,w,d} = \begin{cases} g_{c,d}(L^s_{i,h,w}), & \text{if } L^s_{i,h,w} \text{ is valid} \\ \overline{g}_{c,d}, & \text{otherwise} \end{cases}$$

$$\beta_{i,c,h,w,d} = \begin{cases} h_{c,d}(L^s_{i,h,w}), & \text{if } L^s_{i,h,w} \text{ is valid} \\ \overline{h}_{c,d}, & \text{otherwise} \end{cases}$$

(3)

Intuitively, given a LiDAR point $L^s_{h,w}$ with a valid value, the representation (i.e., $F_{c,h,w,d}$) of its corresponding pixel in the cost volume under a certain disparity level d would be enhanced/suppressed via the conditional modulation when the depth value of $L^s_{h,w}$ is consistent/inconsistent with d. In contrast, for those LiDAR points with invalid values, the regularization upon the cost volume degenerates back to a unconditional batch normalization version and the same modulation parameters $\{\overline{g}_{c,d}, \overline{h}_{c,d}\}$ are applied to them. We experiment the following two different choices for modelling the functions $g_{c,d}$ and $h_{c,d}$:

Categorical CCVNorm: a $\hat{D}$-entry lookup table with each element as a $D \times C$ vector is constructed to map LiDAR values into normalization parameters $\{\gamma, \beta\}$ of different feature channels and disparity levels, where the LiDAR depth values are discretized here into $\hat{D}$ levels as entry indexes.

Continuous CCVNorm: a CNN is utilized to model the continuous mapping between the sparse LiDAR data L^s and the normalization parameters of $D \times C$-channels. In our implementation, we use the first block of ResNet34 [33] to encode LiDAR data, followed by one 1×1 convolution for CCVNorm in different layers respectively.

D. Hierarchical Extension

We observe that both Categorical and Continuous CCVNorm require a huge number of parameters. For each normalization layer, the Categorical version demands $\mathcal{O}(\hat{D}DC)$ parameters to build up the lookup table while the CNN for Continuous one even needs more for desirable performance. In order to reduce the model size for practical usage, we advance to propose a hierarchical extension (denoted as **HierCCVNorm**, which is shown in the top-right of Fig. 3), serving as an approximation of the Categorical CCVNorm with much fewer model parameters. The normalization parameters of HierCCVNorm for valid LiDAR points are computed by:

$$\gamma_{i,c,h,w,d} = \phi^g(d)g_c(L^s_{i,h,w}) + \psi^g(d)$$
$$\beta_{i,c,h,w,d} = \phi^h(d)h_c(L^s_{i,h,w}) + \psi^h(d)$$

(4)

Basically, the procedure of mapping from LiDAR disparity to a $D \times C$ vector in Categorical CCVNorm is now decomposed
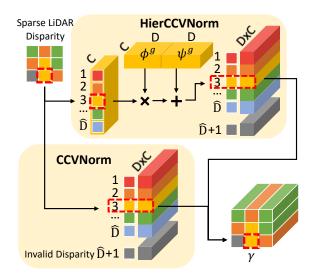

Fig. 3: Conditional Cost Volume Normalization. At each pixel (red dashed bounding box), based on the discretized disparity value of corresponding LiDAR data, categorical CCVNorm selects the modulation parameters γ from a $\hat{D}$-entry lookup table, while the LiDAR points with invalid value are separately handled with an additional set of parameters (in gray color). On the other hand, HierCCVNorm produces γ by a hierarchical modulation of 2 steps, with modulation parameters $g_c(\cdot)$ and $\{\phi^g, \psi^g\}$ respectively (cf. Eq. 4).

into two sequential steps. Take γ for an example, g_c is first used to compute the intermediate representation (i.e., a vector in size C) conditioned on $L^s_{i,h,w}$, and is then modulated by another pair of modulation parameters $\{\phi^g(d), \psi^g(d)\}$ to obtain the final normalization parameter γ. Note that $\phi^g, \psi^g, \phi^h, \psi^h$ are basically the lookup table with the size of $D \times C$. With this hierarchical approximation, each normalization layer only requires $\mathcal{O}(DC)$ parameters.

IV. EXPERIMENTAL RESULTS

We evaluate the proposed method on two KITTI datasets [9][10] and show that our framework is able to achieve favorable performance in comparison with several baselines. In addition, we extensively conduct a series of ablation study to sequentially demonstrate the effectiveness of our design choices in the proposed method. Moreover, we investigate the robustness of our approach with respect to the density of LiDAR data, as well as benchmark the runtime and memory consumption. The code and model will be made available for the public.

A. Experimental Settings

KITTI Stereo 2015 Dataset. KITTI Stereo dataset [9] is commonly used for evaluating stereo matching algorithms. It contains 200 stereo pairs for each of training and testing set, where the images are in size of 1242×375. As the ground truth is only provided for the training set, we follow the identical setting as previous works [25][1] to evaluate our model on the training set with LiDAR data. For model training, since only 142 pairs among the training set are associated with LiDAR scans and they cover 29 scenes in the

TABLE I: Evaluation on the KITTI Stereo 2015 Dataset.

Method	Sparsity	> 3 px ↓	> 2 px ↓	> 1 px ↓
SGM [12]		20.7	-	-
MC-CNN [3]	None	6.34	-	-
GC-Net [5]		4.24	5.82	9.97
Prob. Fusion [25]	LiDAR	5.91	-	-
Park *et al.* [1]	Data	4.84	-	-
Ours Full		**3.35**	**4.38**	**6.79**

TABLE II: Evaluation on the KITTI Depth Completion Dataset.

Data	Method	iRMSE ↓	iMAE ↓	RMSE ↓	MAE ↓
	NConv-CNN [18]	2.60	1.03	0.8299	0.2333
Mono	Ma *et al.* [15]	2.80	1.21	0.8147	0.2499
	FusionNet [19]	2.19	0.93	0.7728	**0.2150**
Stereo	Park *et al.* [1]	3.39	1.38	2.0212	0.5005
	Ours Full	**1.40**	**0.81**	**0.7493**	0.2525

KITTI Completion dataset [10], we hence train our network on the subset of the Completion dataset with images of non-overlapping scenes (i.e., 33k image pairs remained for training).

KITTI Depth Completion Dataset. KITTI Depth Completion dataset [10] collects semi-dense ground truth of LiDAR depth map by aggregating 11 consecutive LiDAR sweeps together, with roughly 30% pixels annotated. The dataset consists of 43k image pairs for training, 3k for validation, and 1k for testing. Since no ground truth is available in the testing set, we split the validation set into 1k pairs for validation and another 1k pairs for testing that contain non-overlapped scenes with respect to the training set. We also note that the full-resolution images (in size of 1216×352) of this dataset are bottom-cropped to 1216×256 because there is no ground truth on the top.

Evaluation Metric. We adopt standard metrics in stereo matching and depth estimation respectively for the two datasets: (1) On KITTI Stereo [9], we follow its development kit to compute the percentage of disparity error that is greater than 1, 2 and 3 pixel(s) away from the ground truth; (2) On KITTI Depth Completion [10], Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and their inverse ones (i.e., iRMSE and iMAE) are used.

Implementation Details. Our implementation is based on PyTorch and follows the training setting of GC-Net [5] to have $\mathcal{L}1$ loss for disparity estimation. The optimizer is RMSProp [34] with a constant learning rate 1×10^{-3}. The model is trained with batch size of 1 using a randomly-cropped 512×256 image for 170k iterations. The maximum disparity is set to 192. We apply CCVNorm to the 21, 24, 27, 30, 33, 34, 35th layers in GC-Net. We note that our full model refers to the setting of having both Input Fusion and HierCCVNorm, unless otherwise specified.

B. Evaluation on the KITTI Datasets

For the KITTI Stereo 2015 dataset, we compare our proposed method to several baselines of stereo matching and LiDAR fusion in Table I. We draw few observations here: 1) Without using any LiDAR data, deep learning-based stereo matching algorithms (i.e., MC-CNN [3] and GC-Net [5]) perform better than the conventional one (i.e., SGM [12]) by a large margin; 2) GC-Net outperforms MC-CNN since its entire stereo matching process is formulated in an end-to-end learning framework, and it even performs competitively compared to two other baselines having LiDAR data fused either in input or output spaces (i.e., Probabilistic Fusion [25]

and Park *et al.* [1] respectively). This observation shows the importance of using an end-to-end trainable stereo matching network as well as designing a proper fusion scheme; 3) Our full model learns to well leverage the LiDAR information into both the matching cost computation and cost regularization stages of the stereo matching network and obtains the best accuracy for disparity estimation against all the baselines.

In addition to disparity estimation, we compare our model with both monocular depth completion approaches and fusion methods of stereo and LiDAR data on the KITTI Completion dataset in Table II. From the results of Park *et al.*, we observe that even with more information from stereo pairs, the performance is not guaranteed to be better than state-of-the-art method for monocular depth completion (i.e., NConv-CNN [18], Ma *et al.* [15], and FusionNet [19]) if the stereo images and LiDAR data are not properly integrated. On the contrary, our method with careful designs of the proposed Input Fusion and HierCCVNorm is able to outperform baselines of both monocular or stereo fusion. It is also worth noting that, our model shows significant boost on the metrics related to inverse depth (i.e., iRMSE and iMAE) since our method is trained to predict disparity. Particularly, we emphasize here the importance of the inverse depth metrics, since they demand higher accuracy in the closer region, which are especially suitable for robotic tasks.

C. Ablation Study

In Table III, we show the effectiveness of the proposed components step-by-step. Two additional baselines for fusion are introduced to have more throughout comparison: **Feature Concat** and **Naive CBN**. Feature Concat uses a ResNet34 [33] to encode LiDAR data, as utilized in other depth completion methods [14][15], and concatenate the LiDAR feature to the cost volume feature. Naive CBN follows a straightforward design of CBN that modulates the cost volume feature conditioned on valid LiDAR depth values.

Overall Results. First, we find that Input Fusion significantly improves the performance comparing to GC-Net. This highlights the significance of incorporating geometry information in the early matching cost computation (MCC) stage, mentioned in Sec. III-B. Next, in the cost regularization (CR) stage, we compare Feature Concat, Naive CBN, and different variants of our methods. All our CCVNorm variants outperform other mechanisms in fusing the LiDAR information to the cost volume in stereo matching networks. This demonstrates the benefit of applying the proposed CCVNorm scheme which serves as a regularization step on

TABLE III: Ablation study on the KITTI Depth Completion Dataset. "IF", "Cat", and "Cont" stand for **I**nput **F**usion, categorical and continuous variants of CCVNorm, respectively. For different stages, "MCC" stands for **M**atching **C**ost **C**omputation and "CR" is **C**ost **R**egularization. The bold font indicates top-2 performance.

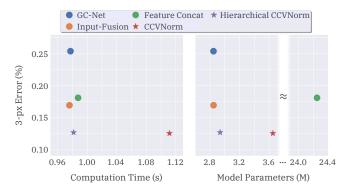| Method	Stage	Disparity					Depth			
		> 3 px ↓	> 2 px ↓	> 1 px ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	iRMSE ↓	iMAE ↓
GC-Net [5]	MCC	0.2540	0.4303	1.5024	0.6526	0.4020	1.0314	0.4054	1.6814	1.0356
+ IF		0.1694	0.3086	1.0405	0.5559	0.3245	0.7659	0.2613	1.4324	0.8362
+ FeatureConcat		0.1810	0.3227	1.1335	0.5946	0.3812	0.8791	0.3443	1.5318	0.9821
+ NaiveCBN		0.2446	0.4342	1.5616	0.6405	0.3915	1.0067	0.3808	1.6505	1.0087
+ CCVNorm (Cont)	CR	0.1363	0.2532	1.0265	0.5856	0.3688	0.8661	0.3385	1.5087	0.9500
+ CCVNorm (Cat)		0.1254	0.2596	1.1348	0.5625	0.3574	0.8942	0.3425	1.4493	0.9209
+ HierCCVNorm (Cat)		0.1268	0.2592	1.1124	0.5615	0.3583	0.8898	0.3403	1.4466	0.9230
+ IF + FeatureConcat	MCC	0.1578	0.2958	1.0012	0.5550	0.3256	0.7622	0.2643	1.4303	0.8389
+ IF + CCVNorm (Cont)	+	0.1460	0.2657	0.9586	0.6137	0.3235	0.7727	0.2573	1.5795	0.8335
+ IF + CCVNorm (Cat)	CR	**0.1194**	**0.2406**	**0.9227**	**0.5409**	**0.3124**	**0.7325**	**0.2501**	**1.3940**	**0.8049**
+ IF + HierCCVNorm (Cat)		**0.1196**	**0.2457**	**0.9554**	**0.5420**	**0.3131**	**0.7493**	**0.2525**	**1.3968**	**0.8069**

Fig. 4: Error v.s. computation time and model parameters. It demonstrates that our hierarchical CCVNorm achieves comparable performance to the original CCVNorm but with much less overhead in computational time and model parameters.
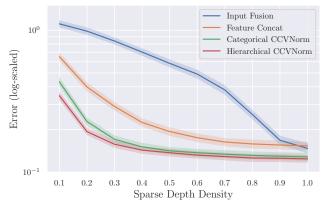

Fig. 5: Robustness to LiDAR density. The 1.0 value in the horizontal axis indicates a complete LiDAR sweep and the shadow indicates the standard deviation. The figure shows that our method is more robust to LiDAR sub-sampling comparing to other baselines.

feature fusion for facilitating stereo matching (Sec. III-C). Finally, our full models with Input Fusion and categorical CCVNorm (with and without the hierarchical extension) produce the best results in the ablation.

Categorical v.s. Continuous. In addition, we empirically find that the categorical CCVNorm may serve as a better conditioning strategy than the continuous variant. Another interesting discovery is that the categorical variant performs competitively compared to the continuous one in most metrics (for disparity) except for the 1-px error. This is not surprising since the conditioning label for categorical CCVNorm is actually discretized LiDAR data, which may possibly lead to the propagation of quantization error. While the continuous variant performs better in 1-px error, they may not necessarily yield better results in sub-pixel errors (i.e., disparity RMSE and MAE), since cost volume is naturally a discretization in the disparity space, thus making the continuous variant harder to handle sub-pixel predictions [5].

Benefits of Hierarchical CCVNorm. In Table III, our hierarchical extension approximates the original CCVNorm and achieves comparable performance. We further show the computational time and model size for various conditioning mechanisms in Fig. 4 that highlights the advantages of our hierarchical CCVNorm. In both sub-figures, the scattered point closer to the left-bottom corner indicates a more

accurate and efficient model. The figure shows that our hierarchical CCVNorm achieves good performance boost with only a small overhead in both computational time and model parameters compared to GC-Net. Note that, *Feature Concat* adopts a standard strategy to encode LiDAR data in depth completion methods [14][15], resulting in much more parameters introduced. Overall, our hierarchical extension can be viewed as an approximation of the original CCVNorm with a huge reduction of computational time and parameters.

D. Robustness to LiDAR Density

In Fig. 5, we study the robustness of different fusion mechanisms to the change of density in LiDAR data. We use 1.0 on the horizontal axis of Fig. 5 to indicate a full LiDAR sweep, and gradually sub-sample from it and observe how the performance of each fusion approach varies. The results highlight that both variants of CCVNorm (i.e., Categorical and Hierarchical CCVNorm) are consistently more robust to different density levels in comparison with other baselines (i.e., Feature Concat and Input Fusion only).

First, Input Fusion is highly sensitive to the density of sparse depth due to its property of treating both valid/invalid pixels equally and setting invalid values as a fixed constant,
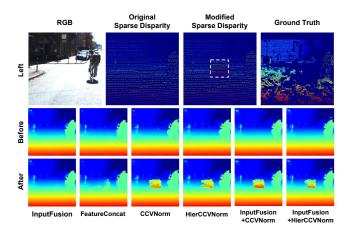
Fig. 6: Sensitivity to LiDAR data. We manually modify the sparse disparity input (indicated by the white dashed box in "Modified Sparse Disparity") and observe the effect in disparity estimates. The results show that all our variants better reflect the modification of LiDAR data during the matching process.

and hence introducing numerical instability during network training/inference. Second, by comparing our two variants with Feature Concat, we observe that both methods do not suffer from severe performance drop in high LiDAR density $(0.7 \sim 1.0)$. However, in low-level density $(0.1 \sim 0.6)$, Feature Concat drastically degrades the performance while ours remains robust to the sub-sampling. This robustness results from our CCVNorm that modulates the pixel-wise feature during the cost regularization stage and introduces additional modulation parameters for invalid pixels (shown in Eq. (3)). Overall, this experiment validates that our CCVNorm can function well under varying or non-stable LiDAR density.

E. Discussions

Sensitivity to Sensory Inputs. In Fig. 6, we present an example to investigate the sensitivity of different fusion mechanisms with respect to the conditional 3D LiDAR data: we manually modify a certain portion of sparse LiDAR disparity map (indicated by the white dashed box on the third image in the top row of Fig. 6), and visualize the changes in stereo matching outputs produced by this modification (referring to the bottom two rows of Fig. 6).

Interestingly, using "Input Fusion only" is unaware of the modification in the LiDAR data and produces almost identical output (before v.s. after in Fig. 6). The reason is that fusion solely on the input level is likely to lose the LiDAR information through the procedure of network inference. For "Feature Concat", where the fusion is performed in the later cost regularization stage, the change starts to be visible but not significant. On the contrary, all our variants based on CCVNorm (or having combination with Input Fusion) successfully reflect the modification of the sparse LiDAR data onto the disparity estimation output. Hence, this verifies again our contribution in proposing proper mechanisms for incorporating sparse LiDAR information with dense stereo matching.

Qualitative Results. Fig. 7 provides an example to illustrate

TABLE IV: Computational time (unit: second). Our method only brings small overhead (0.049 seconds) compared to the baseline GC-Net.

Method	SGM [12]	Prob. [25]	Park. [1]	GC-Net [5]	Ours
Time	0.040	0.024	0.043	0.962	1.011

qualitative comparisons between several baselines and the variants of our proposed method. Our full model (i.e., Input Fusion + hierarchical CCVNorm) is able to handle scenarios with complex structure by taking advantage of the complementary nature of stereo and LiDAR sensors. For instance, as indicated by the white dashed bounding box in Fig. 7, GC-Net fails to estimate disparity accurately on the objects containing the deformed shape (e.g., bicycles) in low illumination. In contrast, our method is capable of capturing the details of bicycles in disparity estimation due to the help from the sparse LiDAR data.

F. Computational Time

We provide an analysis of computational time in Table IV. Except for Probabilistic Fusion [25] which is tested on a Core i7 processor and an AMD Radeon R9 295x2 GPU as reported in the original paper, all the other methods run on a machine with a Core i7 processor and an NVIDIA 1080Ti GPU. In general, the models based on stereo matching networks (i.e., GC-Net and ours) take longer for computation but provide significant improvement in performance (see Table I) in comparison with conventional algorithms. While improving the overall runtime performance via introducing more efficient stereo matching networks is out of the scope of this paper, we show that the overhead introduced by our Input Fusion and CCVNorm mechanisms upon the GC-Net method is only 0.049 seconds, validating the efficiency of our fusion scheme.

V. CONCLUSIONS

In this paper, built upon deep learning-based stereo matching, we present two techniques for incorporating LiDAR information with stereo matching networks: (1) Input Fusion that jointly reasons about geometry information extracted from LiDAR data in the matching cost computation stage and (2) CCVNorm that conditionally modulates cost volume feature in the cost regularization stage. Furthermore, with the hierarchical extension of CCVNorm, the proposed method only brings marginal overhead to stereo matching networks in runtime and memory consumption. We demonstrate the efficacy of our method on both the KITTI Stereo and Depth Completion datasets. In addition, a series of ablation studies validate our method over different fusion strategies in terms of performance and robustness. We believe that the detailed analysis and discussions provided in this paper could become an important reference for future exploration on the fusion of stereo and LiDAR data.

REFERENCES

[1] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation with the 3d lidar and stereo fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 1, 2, 4, 5, 7
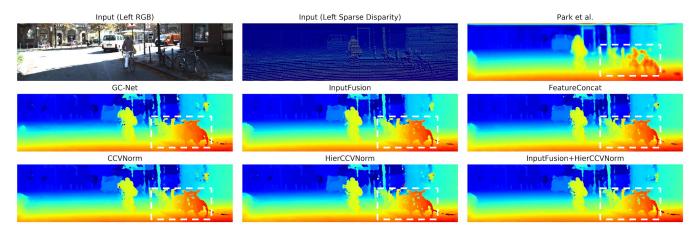
Fig. 7: Qualitative Results. Comparing to other baselines and variants, our method captures details in complex structure area (the white dashed bounding box) by leveraging complementary characteristics of LiDAR and stereo modalities.

[2] S. S. Shivakumar, K. Mohta, B. Pfrommer, V. Kumar, and C. J. Taylor, "Real time dense depth estimation by fusing stereo with sparse depth measurements," *ArXiv:1809.07677*, 2018. 1

[3] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 5

[4] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[5] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 5, 6, 7

[6] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2017. 1, 2

[7] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[8] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[9] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4, 5

[10] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017. 2, 4, 5

[11] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision (IJCV)*, 2002. 2

[12] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008. 2, 5, 7

[13] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[14] F. Mal and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2, 3, 5, 6

[15] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," *ArXiv:1807.00275*, 2018. 2, 5, 6

[16] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017. 2

[17] Z. Huang, J. Fan, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *ArXiv:1808.08685*, 2018. 2

[18] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *ArXiv:1811.01791*, 2018. 2, 5

[19] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," *ArXiv:1902.05356*, 2019. 2, 5

[20] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *European Conference on Computer Vision (ECCV)*, 2018. 2

[21] T.-H. Wang, F.-E. Wang, J.-T. Lin, Y.-H. Tsai, W.-C. Chiu, and M. Sun, "Plug-and-play: Improve depth estimation via sparse data propagation," *AarXiv:1812.08350*, 2018. 2

[22] K. Nickels, A. Castano, and C. Cianci, "Fusion of lidar and stereo range for mobile robots," in *International Conference on Advanced Robotics (ICAR)*, 2003. 2

[23] D. Huber, T. Kanade, *et al.*, "Integrating lidar into stereo for fast and improved disparity computation," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2011. 2

[24] V. Gandhi, J. Čech, and R. Horaud, "High-resolution depth maps based on tof-stereo fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 2

[25] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. 2, 4, 5, 7

[26] E. Perez, H. De Vries, F. Strub, V. Dumoulin, and A. Courville, "Learning visual reasoning without strong priors," *ArXiv:1707.03017*, 2017. 2, 3

[27] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 3

[28] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[29] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ArXiv:1511.06434*, 2015. 2

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997. 2

[31] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *ArXiv:1609.09106*, 2016. 2

[32] C. H. Lin, C.-C. Chang, Y.-S. Chen, D.-C. Juan, W. Wei, and H.-T. Chen, "COCO-GAN: Conditional coordinate generative adversarial network," 2019. 2

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5

[34] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent." 5