

# AN UNSUPERVISED OPTICAL FLOW ESTIMATION FOR LIDAR IMAGE SEQUENCES

Xuezhou Guo\*, Xuhu Lin\*, Lili Zhao, Zezhi Zhu, Jianwen Chen

School of Information and Communication Engineering  
University of Electronic Science and Technology of China  
Chengdu 611731, China

## ABSTRACT

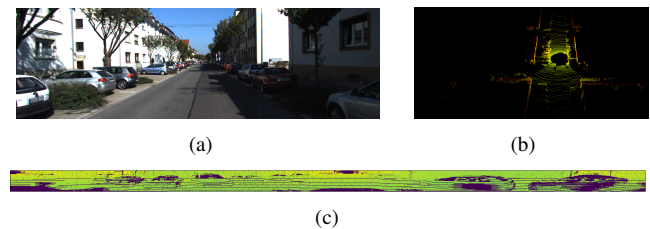
In recent years, the LiDAR images, as a 2D compact representation of 3D LiDAR point clouds, are widely applied in various tasks, e.g., 3D semantic segmentation, LiDAR point cloud compression (PCC). Among these works, the optical flow estimation for LiDAR image sequences has become a key issue, especially for the motion estimation of the inter prediction in PCC. However, the existing optical flow estimation models are likely to be unreliable for LiDAR images. In this work, we first propose a light-weight flow estimation model for LiDAR image sequences. The key novelty of our method lies in two aspects. One is that for the different characteristics (with the spatial-variation feature distribution) of the LiDAR images w.r.t. the normal color images, we introduce the **attention mechanism** into our model to improve the quality of the estimated flow. The other one is that to tackle the lack of large-scale LiDAR-image annotations, we present an unsupervised method, which directly minimizes the inconsistency between the reference image and the reconstructed image based on the estimated optical flow. Extensive experimental results have shown that our proposed model outperforms other mainstream models on the KITTI dataset, with much fewer parameters.

**Index Terms**— Optical Flow, LiDAR Image Sequences, Unsupervised Learning.

## 1. INTRODUCTION

With the rapid development of Light detection and ranging (LiDAR) sensors, the captured sparse point clouds, which contain a set of 3D points representing the surrounding environment, have been widely used in various applications, such as, autonomous driving, augmented reality (AR), and drones. Recently, 2D LiDAR image sequences, which are generated by a projection from 3D LiDAR point clouds, are often encountered in many fields, e.g., the 3D semantic segmentation [1–4] and LiDAR point cloud compression (PCC) [5–8]. Among these works, some efforts on PCC aim to reduce the temporal redundancy of LiDAR point clouds, where the motion estimation becomes an essential issue. It is known that

\* Xuezhou Guo and Xuhu Lin contribute equally.

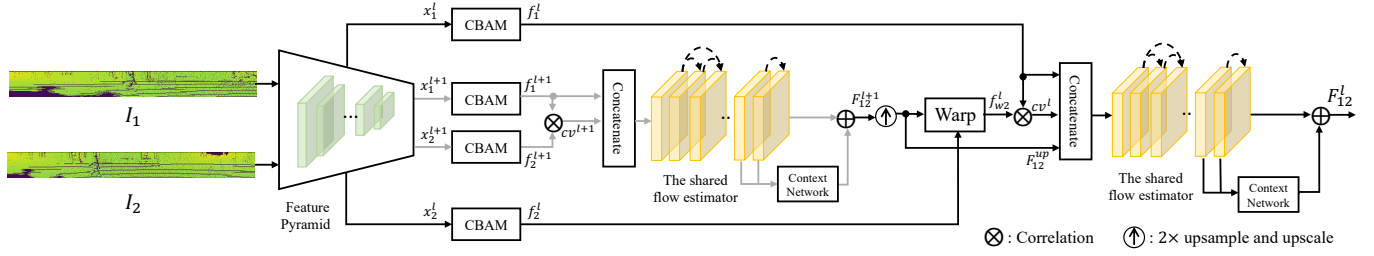


**Fig. 1: Demonstration of one frame in the KITTI dataset.** (a) The original RGB image just for reference; (b) The LiDAR point cloud; (c) The corresponding LiDAR image.

the scene flow [9] is a widely-used format of 3D motion field for LiDAR point clouds (e.g., [10–13]). However, the scene flow estimation is not applicable for 2D LiDAR image sequences. Thus, an effective optical flow estimation method for LiDAR image sequences is needed.

Recent years have witnessed remarkable success in applying CNNs for the optical flow estimation (e.g., [14–16]). Among these methods, PWC-Net [16] is one of the state-of-the-art optical flow estimation algorithms. It exploits a pyramid network to extract features from the normal color images, and then constructs a cost volume as the measurement for the difference between warped and original features. It seems straightforward to naively feed LiDAR image sequences into the existing optical flow estimation models. However, directly applying these models leads to unsatisfactory results. This can be explained by the fact that the intrinsic spatial characteristics of LiDAR images are different from that of the normal color images, as shown in Fig. 1. Unlike the normal color images, the feature distribution of LiDAR images varies greatly at different pixel locations [3]. Therefore, it is not effective to only use the standard convolutions for extracting the features of LiDAR images.

Fortunately, the attention mechanism has shown its potential to tackle this problem. Among the existing state-of-the-art algorithms about the attention mechanism, a Convolutional Block Attention Module (CBAM) [17] is proposed to adaptively refine features along the channel and spatial dimensions. The channel attention determines which features are more meaningful, and the spatial attention focuses on which



**Fig. 2: The architecture of our proposed optical flow estimation model for LiDAR images.** It contains three main parts: the feature pyramid with CBAM, a shared flow estimator for all levels and the context network. For simplicity, we only show the pipeline of the top two levels in different line colors. The remaining modules have the same structure.

portions of the image are more informative.

In this paper, we first propose a light-weight unsupervised flow estimation model for LiDAR image sequences. In general, the main pipeline of our work is based on PWC-Net [16]. Specifically, the LiDAR point clouds are firstly projected into a set of 2D LiDAR images as the inputs of the proposed model. Then, the ‘coarse’ features of the input LiDAR images are extracted by a pyramid network, which is followed by a Convolutional Block Attention Module (CBAM) [17]. After that, the ‘refined’ features will be fed into the subsequent flow estimator. Although the CBAM introduces additional parameters, by sharing the same flow estimator for all levels as [18], our proposed model is with a smaller size than PWC-Net. Finally, a sub-network takes the last two outputs of the flow estimator as inputs and refines the estimated optical flow by enlarging the receptive field.

The main contributions of this work are: (i) To the best of our knowledge, we first propose an effective flow estimation model for LiDAR image sequences, which is based on the unsupervised learning; (ii) To better handle the inherited spatial characteristics of the LiDAR images, we integrate the attention mechanism to improve the capacity of feature extraction for the standard convolutions; (iii) The proposed method achieves a comparable performance w.r.t. the mainstream optical flow estimation methods, but with fewer parameters.

## 2. THE PROPOSED METHOD

Given two frames  $I_1, I_2$  of the LiDAR image sequences, our goal is to estimate the forward optical flow  $F_{12}$ . In this section, we will first illustrate the projection method (i.e., the transformation from 3D point clouds to 2D LiDAR image sequences) for the data pre-processing. Then, various stages of the proposed framework will be described in details.

### 2.1. The Projection from the 3D LiDAR Point Clouds into the 2D LiDAR Image Sequence

As computed in [4], the sparse LiDAR point clouds are firstly projected into a set of 2D LiDAR images (i.e., a LiDAR image

sequence). For each point  $(x, y, z)$  of LiDAR point clouds, a corresponding pixel  $(u, v)$  of the LiDAR image sequence can be available as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0.5 [1 - \arctan(y, x)\pi^{-1}] w \\ [1 - (\arcsin(z, \rho^{-1}) + f_{\text{down}}) f^{-1}] h \end{pmatrix}, \quad (1)$$

where  $w$  and  $h$  represent the width and height of the projected LiDAR image,  $\rho$  denotes the range from each point to the sensor’s center as  $\rho = \sqrt{x^2 + y^2 + z^2}$ , and  $f = |f_{\text{down}}| + |f_{\text{up}}|$  is the vertical field-of-view (FOV) of the LiDAR sensor.

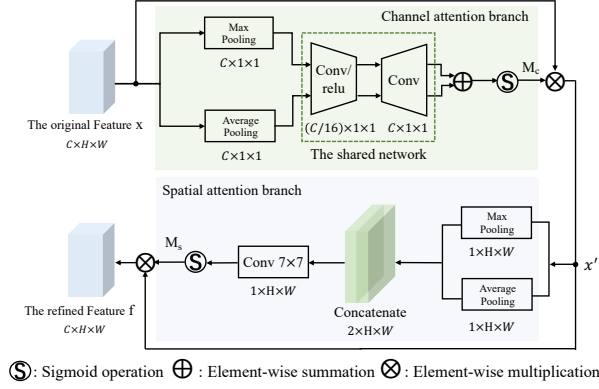
### 2.2. The Optical Flow Estimation Model

The main pipeline of our proposed model is illustrated in Fig. 2. This model contains the following components: the feature pyramid network, the Convolutional Block Attention Module (CBAM), the cost volume, the optical flow estimator and the context network. We will discuss these sub-modules respectively.

**Feature Pyramid Network.** At the first stage, for two consecutive frames of LiDAR images  $I_1$  and  $I_2$ , a seven-level pyramid network is adopted to generate their feature representations. From the 1st to the 7th level, the numbers of feature channels are 1, 16, 32, 64, 96, 128, 192, respectively. **Each level contains two convolutional layers followed by a Leaky ReLU function.** At the  $l^{\text{th}}$  level, the outputs of the pyramid network are the ‘coarse’ feature representations  $x_1^l, x_2^l$  for  $I_1$  and  $I_2$ , respectively. Next,  $x_1^l$  and  $x_2^l$  will be refined by the CBAM.

**Convolutional Block Attention Module.** The Convolutional Block Attention Module (CBAM) can adaptively select important features along the channel and spatial dimension. The structure of the CBAM is shown in Fig. 3. There are two main branches. Suppose that its input feature map is  $x \in \mathbb{R}^{C \times H \times W}$ , where  $C, H, W$  denote the number of channels, the height, and the width of 3D features, respectively.

Firstly, at the channel attention branch as shown in Fig. 3 (top), the CBAM learns a channel attention map  $M_c \in \mathbb{R}^{C \times 1 \times 1}$  to refine  $x$ . The intermediate feature map generated



**Fig. 3:** The structure of CBAM, which is proposed by [17]. It is embedded into our model to refine the extracted features.

in this branch is represented as  $x'$ . Then, at the spatial attention branch as shown in Fig. 3 (bottom),  $x'$  is fed as the input, and the spatial attention map  $M_s \in \mathbb{R}^{1 \times H \times W}$  is learned to refine features  $f$  along the spatial dimension, which is the final desired feature map. This process can be formulated as

$$\begin{aligned} x' &= x \otimes M_c(x), \\ f &= x' \otimes M_s(x'), \end{aligned} \quad (2)$$

where  $\otimes$  denotes the element-wise multiplication. For the ‘coarse’ feature representations  $x_1^l, x_2^l$  from the pyramid network at the  $l^{th}$  level, their refined features, denoted as  $f_1^l, f_2^l$ , are used for the subsequent flow estimation process.

**Cost Volume.** The cost volume is widely used in the optical flow estimation tasks [15, 19]. It measures how closely a pixel’s feature matches up with its corresponding pixel’s feature in the next frame. Specifically, at the  $l^{th}$  level, the obtained flow from the  $l + 1^{th}$  level,  $F_{12}^{l+1}$ , is up-sampled and up-scaled to the current level,  $F_{12}^{up}$ . It warps with  $f_2^l$  towards the first image, and the warped feature map  $f_{w2}^l$  will be obtained. After that, the cost volume  $cv^l$  can be calculated as the correlation between  $f_1^l$  and  $f_{w2}^l$ . For the top level, there is no optical flow from the upper level, so the cost volume is just the correlation between  $f_1^l$  and  $f_2^l$ .

**Optical Flow Estimator.** The concatenation of  $F_{12}^{up}, f_1^l$  and  $cv^l$  is used as the input of the flow estimator. The flow estimator is a six-layer convolutional neural network (CNN). The numbers of the output channels for each layer are 128, 128, 96, 64, 32, 2, respectively. To reduce the computational complexity incurred by the introduced attention mechanism, we follow the architecture of the flow estimator adopted in [18]. In specific, different from PWC-Net [16], the inputs of each convolutional layer are only the outputs of the previous two layers. Besides, the flow estimators at different levels of the pyramid network share the same parameters, while PWC-Net is not in this case.

**Context Network.** After obtaining the predicted flow, a con-

text network is used for refinement. The context network is a CNN implemented by the dilated convolutions, which can enlarge the receptive field to further extract more contextual information [16]. The dilation constants for each level are set to 1, 2, 4, 8, 16, 1, 1, respectively. The inputs of this sub-network are the outputs of the last two layers of the flow estimator. Finally, the refined flow  $F_{12}^l$  is obtained by the context network.

### 2.3. The Unsupervised Learning Strategy

There is no ground-truth optical flow for LiDAR image sequences. Therefore, our model is trained in an unsupervised way. Motivated by the work [20], we adopt the loss function designed for unsupervised optical flow learning tasks. It is to measure the difference between the reference image and the reconstructed one. The reconstructed image is obtained by back-warping the target image towards the reference image based on the estimated flow. Specifically, for the two input LiDAR images  $I_1$  and  $I_2$ , at the  $l^{th}$  level,  $I_1$  and  $I_2$  are down-sampled to match with the estimated flow  $F_{12}^l$ . The sampled images are denoted as  $I_1^l$  and  $I_2^l$ . Then the reconstructed image  $I_r^l$  can be calculated as

$$I_r^l = \text{backwarp}(I_2^l, F_{12}^l). \quad (3)$$

### 2.4. Implementation Details

**The Dataset.** We conduct the experiments on the KITTI dataset [21]. It provides the odometry benchmark, which contains 22 sequences of the point clouds collected by the LiDAR sensors, covering diverse traffic scenes. For each sequence, we project the point clouds into LiDAR image sequences using the method illustrated in Section 2.1, as [1–4, 8]. Then, we divide each three consecutive LiDAR images into a triplet. Specifically, 16 sequences are selected as the training set, while three sequences are chosen as the validation set. The remaining sequences are used as the test set. For each triplet, two range images are used as the inputs to predict the optical flow.

**The Loss Function.** Based on the illustration of the unsupervised loss function described in Section 2.3, different terms are used for the training and fine-tuning of our model. For the training process, following the main pipeline of the multi-scale loss proposed in [14], we use the weighted sum of losses from seven pyramid levels to guide the optical flow learning. The training loss can be formulated as:

$$\mathcal{L}_t = \sum_{l=1}^7 \alpha_l \|I_r^l - I_1^l\|_2, \quad (4)$$

where  $\alpha_1$  to  $\alpha_7$  are set as 0.3, 0.06, 0.08, 0.1, 0.12, 0.14, 0.2, respectively, and  $I_r^l$  is from Equation (3). For the fine-tuning process, recent studies [22, 23] use an occlusion map in the loss function, so as to exclude the effects of occluded pixels.

Inspired by this, in the reconstruction process, we use a binary map  $M_b^l$  generated from  $I_1^l$  to indicate the existence of a point in the space. Modified from Equation (3), the reconstructed image is represented as

$$I_b^l = \text{backwarp}(I_2^l, F_{12}^l) \times M_b^l. \quad (5)$$

Then, the fine-tuning loss function is formulated as

$$\mathcal{L}_{ft} = \sum_{l=1}^7 \alpha_l \|I_b^l - I_1^l\|_2 + \gamma \|\theta\|_2, \quad (6)$$

where  $\theta$  is the learnable weight of this model. In our experiments, we set the regularization parameter  $\gamma$  to  $1e-6$ .

**The Training Strategy.** Firstly, the proposed model is trained from scratch for 60 epochs, with a batch size of 4 and the initial learning rate is  $1e-4$ . The learning rate will decay by 0.1 for every 20 epochs. We adopt the ADAM optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 1e-7$ . Next, the network is fine-tuned with a batch size of 1, using the modified loss function, Equation (6), for another 40 epochs. The initial learning rate is set to  $0.5e-4$ , and it will decay by 0.5 when the validation loss has stopped decreasing for 4 epochs. The ADAM optimizer has the same parameter setting for both training and fine-tuning processes.

### 3. EXPERIMENTAL RESULTS

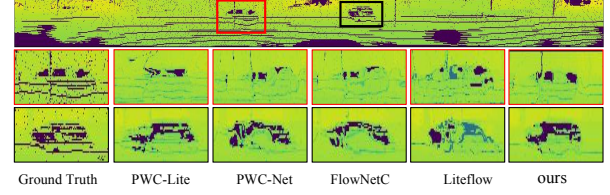
We evaluate the performance of our model on the test set. Several mainstream flow estimation models designed for the normal color images, are compared quantitatively. The experiments are conducted using an Intel Core i7-7700K CPU with a GTX 1080Ti GPU.

#### 3.1. Evaluation Metrics

It is known that the endpoint error (EPE) [20] is used as the metric for the supervised optical flow learning tasks. However, there is no ground-truth flow annotations for the LiDAR image sequences. Therefore, an ‘indirect’ metric is used for evaluation in this paper, i.e., Manhattan distance (i.e., L1 norm), measuring the error between the reference image and the reconstructed one. The reconstructed image is obtained as Equation (3).

#### 3.2. Performance Evaluation

From Table 1, it can be observed that our proposed method outperforms other methods in terms of the inference loss, especially with an improvement of 50% than the original PWC-Net. This result shows the effectiveness of introducing the attention mechanism into the feature extraction. Besides, the results indicate our method has fewer parameters. Note that compared with PWC-Lite, our method slightly increase the model size, but we deliver a better flow estimation. Moreover,



**Fig. 4:** Visual comparison for the reconstructed images. Our model delivers better perceptual quality than other methods.

**Table 1: Quantitative comparisons for the optical flow methods on the KITTI test dataset. ‘c’ means the model without fine-tuning and ‘ft’ means the model with fine-tuning.**

Method	# Param.	L1 loss
PWC-Net [16]	9.37M	3.1848
PWC-Lite [22]	<b>2.24M</b>	2.7664
FlowNetC [15]	39.18M	2.5031
LiteFlowNet [24]	5.38M	4.3231
Ours(c)	2.25M	1.9870
Ours(ft)	2.25M	<b>1.7568</b>

we also provide the results of our model with fine-tuning or not, for showing the effectiveness of adding the binary map in the loss function. Fig. 4 shows the comparison between the ground-truth and reconstructed frame using different flow estimation models. We select two frames from the test set, and set the first image as the ground truth. The reconstructed frame is obtained by back-warping the second frame based on the estimated optical flow. It can be found that compared with other methods, our proposed model can reconstruct a more accurate LiDAR image, especially for the object’s size and shape. These results all demonstrate the effectiveness of our flow estimation method.

### 4. CONCLUSION

In this paper, a novel flow estimation method is firstly proposed for the LiDAR image sequences, which exploits the attention mechanism strategy to further improve the quality of flow estimation. Moreover, to address the issue about the lack of the annotations of LiDAR images, the unsupervised learning strategy is introduced by minimizing the error between the reference image and the warped image obtained by the estimated optical flow. Extensive simulation results on the KITTI dataset have clearly shown that for the optical flow estimation of the LiDAR image sequences, our proposed model outperforms the existing flow estimation models, even with much fewer parameters.

### 5. ACKNOWLEDGMENT

This work is supported by the Sichuan Science and Technology Program under grant No. 2018JY0610.



## 6. REFERENCES

- [1] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1887–1893.
- [2] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Jun. 2019, pp. 4376–4382.
- [3] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, "SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation," *arXiv e-prints*, p. arXiv:2004.01803, Apr. 2020.
- [4] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet ++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.
- [5] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, "Compressing continuous point cloud data using image compression methods," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1712–1719.
- [6] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, "Real-time streaming point cloud compression for 3D LiDAR sensor using U-Net," *IEEE Access*, vol. 7, pp. 113616–113625, Aug. 2019.
- [7] C. Tu, E. Takeuchi, A. Carballo, C. Miyajima, and K. Takeda, "Motion analysis and performance improved method for 3D LiDAR sensor data compression," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 243–256, Jan. 2021.
- [8] X. Sun, S. Wang, M. Wang, Z. Wang, and M. Liu, "A novel coding architecture for LiDAR point cloud sequence," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5637–5644, Oct. 2020.
- [9] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 475–480, 2005.
- [10] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Rigid scene flow for 3D LiDAR scans," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1765–1770.
- [11] A. K. Ushani, R. W. Wolcott, J. M. Walls, and R. M. Eustice, "A learning approach for real-time temporal scene flow estimation from LIDAR data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 5666–5673.
- [12] A. Behl, D. Paschalidou, S. Donné, and A. Geiger, "PointFlowNet: Learning representations for 3D scene flow estimation from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7954–7963.
- [13] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 529–537.
- [14] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
- [15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," pp. 1647–1655, 2017.
- [16] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8934–8943.
- [17] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
- [18] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6489–6498.
- [19] J. Xu, R. Ranftl, and V. Koltun, "Accurate optical flow via direct cost volume processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5807–5815.
- [20] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2017, pp. 1495–1501.
- [21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3354–3361.
- [22] P. Liu, M. Lyu, I. King, and J. Xu, "Selfflow: Self-supervised learning of optical flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4571–4580.
- [23] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2018, pp. 7251–7259.
- [24] T. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8981–8989.