PWOC-3D: Deep Occlusion-Aware End-to-End Scene Flow Estimation

Rohan Saxena[1,2] René Schuster[1] Oliver Wasenmüller[1] Didier Stricker[1,3]

***Abstract*— In the last few years, convolutional neural networks (CNNs) have demonstrated increasing success at learning many computer vision tasks including dense estimation problems such as optical flow and stereo matching. However, the *joint* prediction of these tasks, called scene flow, has traditionally been tackled using slow classical methods based on primitive assumptions which fail to generalize. The work presented in this paper overcomes these drawbacks efficiently (in terms of speed and accuracy) by proposing PWOC-3D, a compact CNN architecture to predict scene flow from stereo image sequences in an end-to-end supervised setting. Further, large motion and occlusions are well-known problems in scene flow estimation. PWOC-3D employs specialized design decisions to explicitly model these challenges. In this regard, we propose a novel self-supervised strategy to predict occlusions from images (learned without any labeled occlusion data). Leveraging several such constructs, our network achieves competitive results on the KITTI benchmark and the challenging FlyingThings3D dataset. Especially on KITTI, PWOC-3D achieves the second place among end-to-end deep learning methods with 48 times fewer parameters than the top-performing method.**

I. INTRODUCTION

In robot navigation, particularly in an autonomous driving pipeline, estimating the motion of other traffic participants is one of the most crucial components of perception. Scene flow is one such reconstruction of the complete 3D motion of objects in the world. Due to the rich perceptual information it provides, it serves as a foundation for several high-level driving tasks in Advanced Driver Assitance Systems (ADAS) and autonomous systems.

Owing to the increased complexity of a full 3D reconstruction, the projection of this 3D motion on the image plane (called optical flow) often serves as an approximate proxy for motion sensing and perception in intelligent vehicles. Consequently, individual components of scene flow, namely optical flow and stereo disparity, have received significantly more attention in computer vision research.

However, the joint estimation of these tasks as scene flow has multiple benefits: Firstly, stereo matching is essentially a special case of optical flow where pixel matching is constrained along the epipolar line. Using a shared representation for these related tasks can result in fewer trainable parameters with a smaller memory footprint and faster training/inference with fewer computational resources. Secondly, the principle of multi-task learning [1], [2] states that jointly training this shared representation improves generalization by leveraging domain-specific information across tasks, with the training signals for one task serving as an inductive bias for the other.

[1]German Research Center for Artificial Intelligence - DFKI, Kaiserslautern, Germany, `firstname.lastname@dfki.de`
[2]Birla Institute of Technology and Science - BITS Pilani, India
[3]University of Kaiserslautern - TUK, Kaiserslautern, Germany

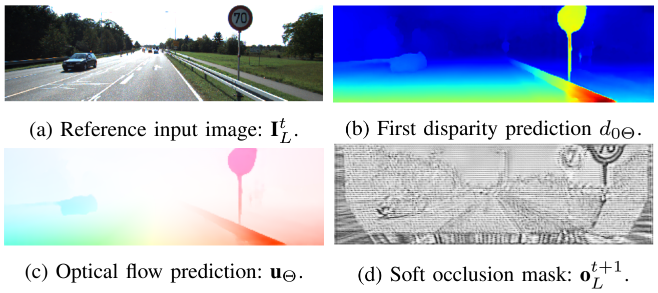

(a) Reference input image: $\mathbf{I}_L^t$. (b) First disparity prediction $d_{0\Theta}$. (c) Optical flow prediction: $\mathbf{u}_\Theta$. (d) Soft occlusion mask: $\mathbf{o}_L^{t+1}$.

Fig. 1. Example predictions of our PWOC-3D network on the KITTI benchmark. PWOC-3D uses CNNs to predict scene flow in an efficient end-to-end fashion. The soft occlusion map (d) is predicted by our novel self-supervised occlusion reasoning mechanism, which is leveraged to improve scene flow estimates.

Our scene flow approach also offers the advantages of robustness and speed over earlier approaches, both of which are critical for deployment in intelligent vehicles. Firstly, classical methods (particularly variational techniques) are based on data consistency assumptions like constancy of brightness or gradient across images, or the smoothness and constancy of motion within a small region. Such primitive assumptions fail to generalize due to varying illumination (shadows or lighting changes), occlusions or large displacements; all of which are prevalent on the dynamic road scenes. Since such scenes are a common scenario for intelligent vehicles, these methods are rendered ineffective on them. In contrast, our network does not make *any* such assumptions and can be used to estimate scene flow in a wide variety of environments with complex motions. Secondly, autonomous driving systems and embedded devices require real-time estimation of scene flow. The traditional approaches are iterative and can take anywhere between 1-50 minutes to process a set of images. In comparison, our end-to-end CNN can perform the same task with a single forward pass of the network in less than 0.2 seconds.

Our network also contains specialised constructs to handle specific challenges in a typical scene flow pipeline. Firstly, inspired by the work of PWC-Net [3] for optical flow, we employ a coarse-to-fine estimation scheme using a spatial pyramid and warp image features at each pyramid level using the intermediate flow estimate from the previous level to handle large motion. Secondly, we propose a novel self-supervised strategy to predict dense occlusion maps from images without using any labelled occlusion data and use these to improve scene flow estimates (cf. Figure 1). To the best of our knowledge, ours is the first method to reason about occlusion using a single flow prediction and without any occlusion ground truth. Previous methods [4], [5], [6] require at least bidirectional flow (forward and backward preditions) to model occlusion, thus our network reduces the effort by half compared to these methods.

Our network design demonstrates that embedding vision techniques, which leverage the underlying domain-knowledge of the problem and geometry of the scene, within differentiable CNNs produces better results than either approach has been able to single-handedly achieve. We demonstrate the performance of our method on the KITTI benchmark, where our method achieves the second highest accuracy among end-to-end CNN methods with 48 times fewer parameters than the top-performing method [7].

II. RELATED WORK

Conventional Scene Flow. First scene flow approaches were inspired by variational methods for optical flow [8]. These approaches frame scene flow estimation as global minimization problem and optimize it using variational calculus. The total energy or cost consists of a data term and a smoothness term, which is then optimized iteratively using the Euler-Lagrange equations. The data term typically models a means of 3D reconstruction of the scene by incorporating a cost based on the similarity of pixel intensities at the reference image and the other images warped towards the reference image using flow predictions. There are various works in this area with different types of underlying assumptions, such as [9], [10], [11], [12], [13], [14], [15].

To overcome the data consistency assumptions of variational methods, various techniques have explored segmenting the scene into rigid planes and estimating piecewise rigid scene flow. The work of [16] decomposes the scene into such piecewise rigid planes (superpixels) and employs a discrete-continuous CRF to optimize scene flow. [17] further combines the scene flow model with discrete optical flow estimates [18] to handle large motion. The work of [19] improves this model by propagating the object labels and their motion information from the previous frame to produce temporally consistent scene flow. The work of [20] also uses superpixel segmentation and formulates scene flow using a factor graph, which enables decomposition of the problem into photometric, geometric and smoothing constraints. Some methods have also used a similar formulation of the problem while exploiting multiple views. For instance, [21] also employs rigid plane segmentation, but in a multi-frame setting, and performs segment-level occlusion reasoning. The work of [22] decomposes multi-frame scene flow into rigid and non-rigid optical flow and stereo matching, combined with ego-motion estimation and motion segmentation. All methods in this category are greatly restricted in applicability due to the rigid motion formulation, which suits only the KITTI dataset well (since its ground truth was constructed by modelling dynamic objects as rigid). On the other hand, our method can be applied in a general scenario with non-rigid and non-planar objects moving in arbitrary 3D paths. This is particularly important for intelligent vehicles which encounter non-rigid objects such as motorbikes, cyclists and pedestrians in dynamic road scenes. SceneFlowFields [23] uses a sparse-to-dense approach instead of rigid-planar assumptions for regularization to achieve a better generalization. Though the accuracy of this methods is competitive, the speed is still far from real time.

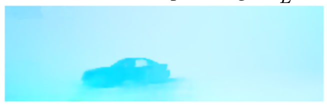
Semantic Scene Flow. More recent methods explore CNNs for semantic segmentation over superpixel segmentation. The work of [24] studied different granularities of instance recognition (including segmentation) using CNNs and explored how they could be employed to improve scene flow predictions from a CRF model. [25] similarly leverages instance-level semantic segmentation cues from a CNN to improve piecewise-rigid scene flow estimates using a cascade of CRFs. This class of methods will be heavily biased on the instance-level recognition dataset that is used to train the CNNs. In the settings described, the instances were obtained from the same dataset as the scene flow benchmark (KITTI), which is not the case in general.

End-to-end CNN for Optical Flow. The incorporation of a spatial pyramid and warping at different pyramid levels in an end-to-end CNN for optical flow estimation was first introduced in SPyNet [26]. PWC-Net [3] built upon SPyNet's pipeline by replacing the latter's image pyramid with a feature pyramid learned using a CNN. A cost volume was used to predict optical flow instead of plain features, and an additional network was used to refine predictions from the last pyramid level. Our PWOC-3D uses the PWC-Net architecture as a skeleton, but differs from it in several ways. Firstly, PWOC-3D reasons about the full 3D motion of objects rather than just 2D optical flow. This is made possible through several key design decisions: We construct four image pyramids (one for each image in the stereo sequence), we define 1D (for disparity) and 2D (for optical flow) versions of the warping and cost volume operations in our network. The 1D operations leverage the epipolar constraint for rectified stereo images to limit computation. Secondly, we employ a feature pyramid network (FPN) [27] to construct the feature pyramids instead of PWC-Net's generic CNN feature extractor (as illustrated in Figure 3), with significant improvement in results. Thirdly, PWOC-3D explicitly reasons about occlusion via our novel self-supervised method of predicting occlusion directly from images (without any occlusion ground truth), and exploits this understanding to improve scene flow predictions. The entire PWOC-3D pipeline is described in detail in the Section III.

End-to-End CNN for Scene Flow. Currently, there are only a couple of other end-to-end CNN architectures published for scene flow. The first was proposed alongside the FlyingThings3D [28] dataset primarily as a proof-of-concept of the utility of the dataset. This network contained roughly three times the number of trainable parameters of FlowNet [29]. In contrast, our method outperforms it while being smaller than a single FlowNet model. The second end-to-end CNN work was presented in [7]. This network used three separate processing pipelines to predict optical flow, initial and final disparities respectively. It was able to demonstrate

(a) Reference input image: $\mathbf{I}_L^t$.

(b) Left image from $t+1$: $\mathbf{I}_L^{t+1}$.


(c) Our predicted optical flow from $\mathbf{I}_L^t$ to $\mathbf{I}_L^{t+1}$.

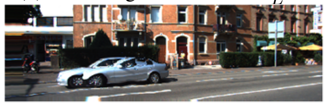

(d) $\mathbf{I}_L^{t+1}$warped towards $\mathbf{I}_L^t$ using our predicted optical flow.

Fig. 2. The adverse effect of occlusion on the warping operation. In (d), there are 2 cars visible: The car on the right is the 'true' car. The part of the road which is visible in $\mathbf{I}_L^t$ and occluded by the car in $\mathbf{I}_R^t$ is static, due to which the occluding area of the car is reproduced incorrectly from $\mathbf{I}_L^t$.

competitive performance on the KITTI benchmark, although with the parameters of ten FlowNet models. In contrast, our fast and compact network is a close second place to this method with 48 times fewer parameters.

Occlusion Handling for Flow. MirrorFlow [4] predicted bidirectional flow using variational methods and used it to warp both images towards each other. A forward-backwards consistency check was imposed on these warps. Areas which did not pass this check were considered occluded. This led them to predict consistent occlusion maps in both directions. UnFlow [5] used a very similar formulation of the problem: Bidirectional flow estimation, forward-backward consistency check, occlusion estimation. The difference here was that a FlowNet was used to predict flow instead of variational methods, and occluded areas were masked from contributing to the reconstruction loss function. Wang et al. [6] also used the same basic pipeline as UnFlow. This work proposed a different method of predicting occlusion maps based on warping a constant grid using the predicted flow.

All previous methods predicted occlusion using bidirectional flow. In contrast, PWOC-3D estimates occlusions in three images $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$ without any labeled occlusion data while computing only the forward direction flow.

III. METHOD

Our PWOC-3D pipeline involves extracting a feature pyramid for each of the four images $\mathbf{I}_L^t$, $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$. The features of $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$ at a particular pyramid level (except the top, i.e. lowest resolution) are warped towards the features of $\mathbf{I}_L^t$ using the flow estimates from the upper pyramid level. Based on the warped features, occlusion maps are predicted for $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$. A cost volume is then constructed using the features of $\mathbf{I}_L^t$ and each of the warped features of $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$ considering the predicted occlusions. Afterwards, a scene flow estimator network is used to predict scene flow using these cost volumes. Finally, a context network with dilated convolutions is used to refine the scene flow estimates. The complete overview of the end-to-end architecture is given in Figure 3. Figure 4 shows a detailed view of the pipeline at a particular pyramid level l.

Feature Pyramids. In PWC-Net [3], a simple feedforward strided CNN is used to construct feature pyramids for both

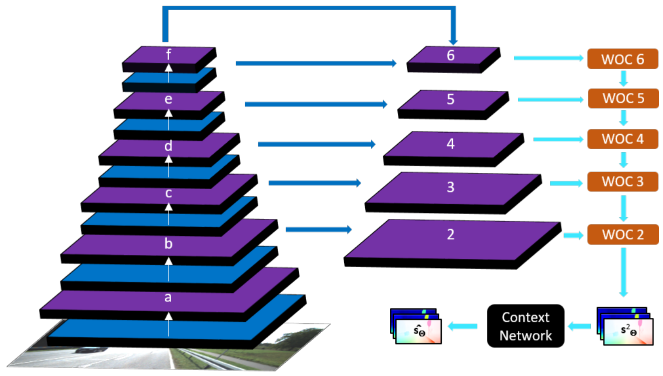


Fig. 3. Visualization of the flow of information across pyramid levels in our entire PWOC-3D pipeline. PWC-Net uses only the levels b, c, d, e, f as a feature pyramid, while we use levels $2, 3, 4, 5, 6$. The orange boxes represent warping, occlusion estimation, cost volume computation, and scene flow prediction for one level of the pyramid as shown in Figure 4.

the input images. However, using the different feature maps of a generic CNN in this manner is not an optimal strategy. This is because the high-resolution feature maps from the first few layers of the network contain well-localized, but semantically weak features; while low-resolution maps from deeper layers contain processed and semantically strong features which are not well-localized with the original image due to strided subsampling of the convolution operation [30].

The FPN [27] work proposes to overcome this problem by incorporating additional connections in the network as shown in Figure 3. In addition to the backbone bottom-up pathway (which computes a feature hierarchy as in a standard CNN), top-down pathways and lateral residual connections are introduced. The top-down pathway produces higher resolution semantically stronger feature maps, while the lateral skip connections (from layers lower in the pipeline) reinforce localization of features with respect to the input images. Combined, this mechanism leads to the feature map at each pyramid level being well-localized and semantically strong.

Since our architecture uses a coarse-to-fine estimation approach, especially with predictions from higher up in the pyramid being used in lower levels, consistency of semantic strength and spatial localization of the features across levels becomes particularly important. Thus, our work explores the role of FPN-like connections in the PWOC-3D pipeline.

For each of the 4 input images $\mathbf{I}_L^t$, $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$, we use the same network (from Figure 3) to construct 4 six-layered feature pyramids (denoted as $\mathbf{c}_L^t, \mathbf{c}_R^t, \mathbf{c}_L^{t+1}, \mathbf{c}_R^{t+1}$ respectively), with each subsequent pyramid level having half the resolution (in each dimension) of its predecessor, so that the subsampling factor at layer l is 2^l for each dimension. We start processing from the topmost level and continue the coarse-to-fine estimation scheme until feature pyramid level 2. That is, PWOC-3D produces scene flow estimations at $1/4$th of the input resolution in each dimension. We upsample the prediction using bilinear interpolation to obtain full-scale scene flow.

Warping. At every pyramid level l, the feature maps of $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$ (denoted as ${}_l\mathbf{c}_R^t$, ${}_l\mathbf{c}_L^{t+1}$, ${}_l\mathbf{c}_R^{t+1}$ respectively) are

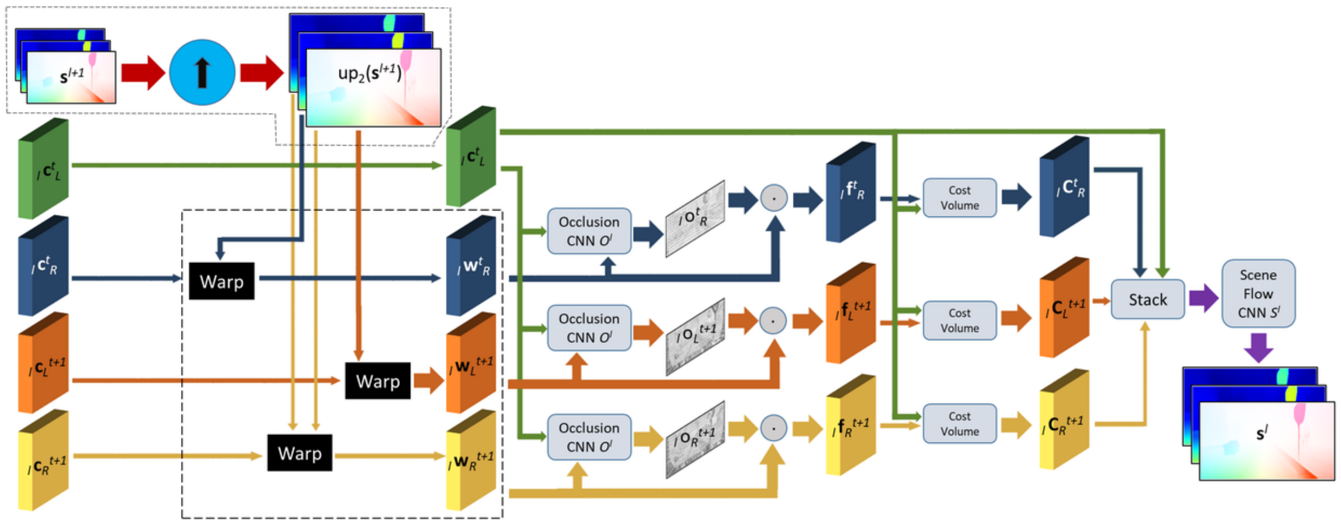


Fig. 4. An overview of the inference pipeline of PWOC-3D at pyramid level l. The operations within the dashed boundary denote the warping operations, which are present at every pyramid level except the topmost.

warped towards the reference image $\mathbf{I}^t_L$. Let the scene flow estimate at level l be denoted as $\mathbf{s}^l = (u^l, v^l, d^l_0, d^l_1)^\top$, the images are warped as follows:

- ${}_l\mathbf{c}^t_R$ is warped towards $\mathbf{I}^t_L$ using the disparity d_0^{l+1}, a 1D warping:

$$ {}_l\mathbf{w}^t_R(\mathbf{x}) = {}_l\mathbf{c}^t_R\left(\left(x - \text{up}_2\left(d_0^{l+1}\right)(\mathbf{x}),\ y\right)^\top\right), \tag{1}$$

  where $\text{up}_2(d_0^{l+1})$ denotes the predicted disparity map from level $l{+}1$ which has been upsampled by a factor of 2 using bilinear interpolation, and $\mathbf{x} = (x, y)^\top$ denotes the pixel index.
- ${}_l\mathbf{c}_L^{t+1}$ is warped towards $\mathbf{I}^t_L$ using optical flow $\mathbf{u}^{l+1} = (u^{l+1}, v^{l+1})^\top$, a 2D warping:

$$ {}_l\mathbf{w}_L^{t+1}(\mathbf{x}) = {}_l\mathbf{c}_L^{t+1}\left(\mathbf{x} + \text{up}_2\left(\mathbf{u}^{l+1}\right)(\mathbf{x})\right). \tag{2}$$

- ${}_l\mathbf{c}_R^{t+1}$ is warped towards $\mathbf{I}^t_L$ using optical flow $\mathbf{u}^{l+1} = (u^{l+1}, v^{l+1})^\top$ and disparity d_1^{l+1}, a modified 2D warping:

$$ \begin{aligned} &{}_l\mathbf{w}_R^{t+1}(\mathbf{x}) = \\ &{}_l\mathbf{c}_R^{t+1}\Big(\left(x - \text{up}_2\left(d_1^{l+1}\right)(\mathbf{x}) + \text{up}_2\left(u^{l+1}\right)(\mathbf{x}),\right. \\ &\quad \left. y + \text{up}_2\left(v^{l+1}\right)(\mathbf{x})\right)^\top\Big). \end{aligned} \tag{3}$$

Occlusion Mechanism. Occlusion, which is omnipresent in real-world dynamic scenes, plays an important role in the estimation of scene flow. Firstly, it leads to incorrect matching costs being computed since the object of interest is occluded from view. Secondly, the lack of information about the occluded area can throw off a naïve method because tracking the occluded pixels directly is impossible, and we must leverage other information to estimate this occluded motion. Thirdly, a considerable area of the reference image is occluded from view when it moves out of the field of view of the camera due to motion of the camera itself. This ego-motion is an inherent characteristic of autonomous driving systems. Thus, failing to account for occlusion has significant drawbacks. Occlusion also has an adverse effect on the (1D and 2D) warping operation, as illustrated in Figure 2.

PWOC-3D employs a novel strategy to handle occlusion by learning an occlusion model conditioned on the input images. The occlusion mechanism is explained using $\mathbf{I}^t_R$, but also applies in an analogous manner to the images $\mathbf{I}_L^{t+1}$ and $\mathbf{I}_R^{t+1}$. Specifically, occlusion in the image $\mathbf{I}^t_R$ with respect to the reference image $\mathbf{I}^t_L$ is modeled at each pyramid level l as an occlusion map ${}_l\mathbf{o}^t_R(\mathbf{x})$ where ${}_l\mathbf{o}^t_R : \Omega \to [0, 1]$ and Ω denotes the image plane. Here 0 corresponds to occluded pixels while 1 corresponds to visible pixels. Since each pixel value is continuous, this is a soft occlusion map from which a hard occlusion map can be obtained by thresholding it appropriately to have discrete 0 and 1 values.

This soft occlusion map is incorporated into PWOC-3D by multiplying it pixel-wise (by broadcasting along the channel dimension) with the corresponding warped features to result in masked features ${}_l\mathbf{f}^t_R$:

$$ {}_l\mathbf{f}^t_R(\mathbf{x}) = {}_l\mathbf{c}^t_R(\mathbf{x}) \cdot {}_l\mathbf{o}^t_R(\mathbf{x}) \tag{4}$$

This has the effect of masking out occluded pixels from ${}_l\mathbf{w}^t_R$, leaving only the non-occluded areas. These masked warped features are then used to construct the cost volume. The occluded pixels having been masked to 0 results in the cost for these pixels to also be computed as 0. In the cost volume operation, a higher cost means a higher degree of matching between two pixels. Thus, a cost of 0 as computed for the occluded areas reflects on no matching at all, which is semantically correct, since a pixel occluded in one image must not match with any other pixel in another image.

Due to the coarse-to-fine estimation approach adopted by PWOC-3D, the occlusion model is also a multi-scale mechanism. A separate occlusion map is predicted at each pyramid level l for each of the warped image features (${}_l\mathbf{w}^t_R$, ${}_l\mathbf{w}_L^{t+1}$, ${}_l\mathbf{w}_R^{t+1}$), which masks out occluded areas for feeding to the respective cost volume at that level.

Learning Occlusions. For predicting occlusion, we train a separate network $\mathcal{O}^l$ at each scale l which maps the depthwise stacked feature maps ${}_l\mathbf{c}_L^t$ and ${}_l\mathbf{w}_R^t$ to the soft occlusion map ${}_l\mathbf{o}_R^t$. These stacked feature maps are used as input to the network because they provide sufficient information to predict occlusion: Regions without occlusion would have similar features in the feature map ${}_l\mathbf{c}_L^t$ and the warped features ${}_l\mathbf{w}_R^t$; whereas pixels which are visible in ${}_l\mathbf{c}_L^t$ but occluded in ${}_l\mathbf{c}_R^t$ would have a dissimilarity in the features in ${}_l\mathbf{w}_R^t$, which can enable the network to predict such pixels as occluded.

The design of the $\mathcal{O}^l$ network consists of six layers of convolution, all of which employ a kernel of size 3×3, a stride of 1 and a padding of 1. The channel dimensions of the occlusion estimator network in each layer are 128, 96, 64, 32, 16 and 1 respectively. The last layer uses sigmoid as an activation function to ensure that the occlusion predictions are in the range $[0, 1]$. All other layers use the leaky ReLU activation function.

This network is inserted in the PWOC-3D pipeline after the warping operation and before the cost volume construction stage at each pyramid level. After warping the three images towards the reference image, the reference image features are stacked with each of the other feature maps as $[{}_l\mathbf{c}_L^t,\ {}_l\mathbf{w}_R^t]$, $[{}_l\mathbf{c}_L^t,\ {}_l\mathbf{w}_L^{t+1}]$, $[{}_l\mathbf{c}_L^t,\ {}_l\mathbf{w}_R^{t+1}]$ and sequentially fed as input to the occlusion estimator network $\mathcal{O}^l$ at that pyramid level to obtain the occlusion maps ${}_l\mathbf{o}_R^t$, ${}_l\mathbf{o}_L^{t+1}$, ${}_l\mathbf{o}_R^{t+1}$ respectively. Since the estimation of occlusion for each of the 3 images requires learning the same underlying task (that of learning to match features in the stacked image feature maps) a single occlusion estimator network $\mathcal{O}^l$ is used at each pyramid level to predict occlusion for each of the three images separately.

To maintain consistency of occlusion predictions across scales, every occlusion network $\mathcal{O}^l$ (except the network at the highest pyramid level) receives as additional input from the network $\mathcal{O}^{l+1}$ (of the upper pyramid level), the output features ${}_{l+1}\mathbf{g}$ of its penultimate convolutional layer and the corresponding predicted occlusion map $\mathbf{o}$. That is, $({}_{l+1}\mathbf{g}_R^t,\ {}_{l+1}\mathbf{o}_R^t)$, $({}_{l+1}\mathbf{g}_L^{t+1},\ {}_{l+1}\mathbf{o}_L^{t+1})$ and $({}_{l+1}\mathbf{g}_R^{t+1},\ {}_{l+1}\mathbf{o}_R^{t+1})$ are the additional inputs to predict ${}_l\mathbf{o}_R^t$, ${}_l\mathbf{o}_L^{t+1}$ and ${}_l\mathbf{o}_R^{t+1}$ respectively.

The occlusion estimator networks at each level can learn to predict the occlusion weights based on the degree of similarity between the reference image features and the warped features. These weights are used to mask the incorrect matching costs in the cost volume, which results in more robust scene flow estimations. Thus, the network *supervises itself* while learning to estimate occlusions, with the goal of improving scene flow estimates (minimizing the error on scene flow predictions) without any labeled occlusion data. Note that the training of PWOC-3D requires ground truth scene flow data; only the estimation of the occlusion maps is self-supervised.

Cost Volume. We compute a cost volume using the reference image features ${}_l\mathbf{c}_L^t$ and the masked warped features ${}_l\mathbf{f}_R^t$, ${}_l\mathbf{f}_L^{t+1}$ and ${}_l\mathbf{f}_R^{t+1}$. In contrast, PWC-Net computed the cost volume using simply the warped features ${}_l\mathbf{w}_R^t$, ${}_l\mathbf{w}_L^{t+1}$ and ${}_l\mathbf{w}_R^{t+1}$, which made its predictions susceptible to problems caused by occlusions.

We construct only a partial cost volume by limiting the search range to a maximum displacement of d_{max} pixels around each pixel. For the 1D cost volume operation (between $({}_l\mathbf{c}_L^t,\ {}_l\mathbf{f}_R^t)$), we search for matches in the horizontal dimension only along the epipolar line, while for the 2D cost volume (between $({}_l\mathbf{c}_L^t,\ {}_l\mathbf{f}_L^{t+1})$ and $({}_l\mathbf{c}_L^t,\ {}_l\mathbf{f}_R^{t+1})$) we search in 2D space. We then organize the resulting cost volume as a 3D array of dimensions $H \times W \times D$ and $H \times W \times D^2$ for the 1D and 2D cost volumes respectively where H and W are the height and width of the feature maps respectively and $D = 2d_{\text{max}} + 1$.

The matching cost in the cost volume is computed as the correlation between the feature vectors of two pixels. Consider ${}_l\mathbf{c}_L^t,\ {}_l\mathbf{f}_R^t : \Omega \mapsto \mathbb{R}^c$, where Ω is the set of image pixel vectors and c is the number of channels of the feature maps. Then the correlation between two patches centred at pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ is computed as a vector of dimensionality D^2 where each individual pixel cost is given by:

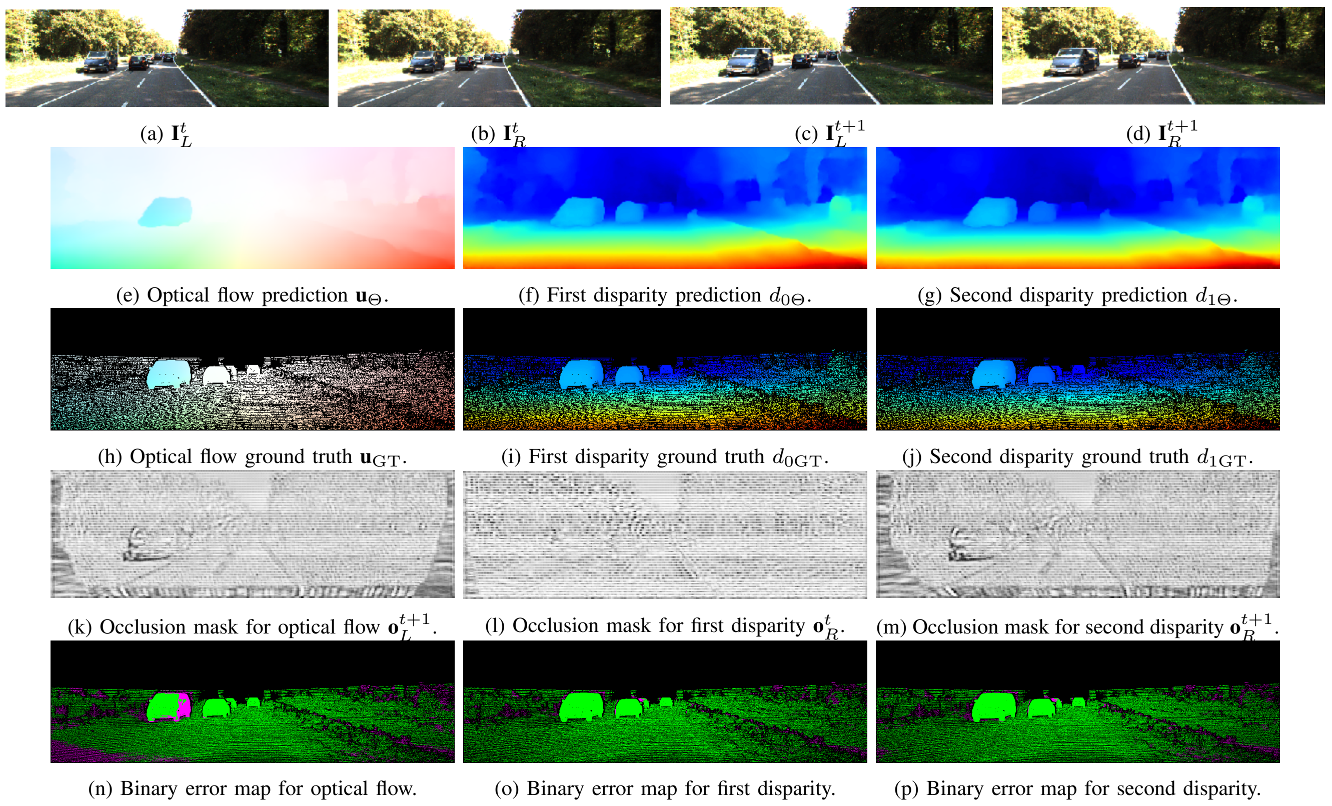
$$
{}_l\mathbf{C}_R^t(\mathbf{x}_1, \mathbf{x}_2|\mathbf{q}) = \frac{({}_l\mathbf{c}_L^t(\mathbf{x}_1))^\top {}_l\mathbf{f}_R^t(\mathbf{x}_2 + \mathbf{q})}{c}, \tag{5}
$$

where $\mathbf{q} \in \{(q_0, q_1)^\top : q_0, q_1 \in [-d_{\text{max}}, d_{\text{max}}]\}$.

Scene Flow Prediction. At every pyramid level l, we train a CNN $\mathcal{S}^l$ to estimate scene flow using the masked cost volume described above. The primary input to this network consists of the three warped and masked cost volumes corresponding to ${}_l\mathbf{f}_R^t$, ${}_l\mathbf{f}_L^{t+1}$ and ${}_l\mathbf{f}_R^{t+1}$ stacked one over the other along the channel dimension. The architecture of this network is similar to the occlusion estimator network: It consists of six layers of convolution filters with a kernel size of 3×3, a stride of 1 and a padding of 1. The channel dimensions of each layer are 128, 128, 96, 64, 32 and 4 respectively. Each layer employs the leaky ReLU activation, except the last layer which does not use any activation function to facilitate it to predict continuous scene flow values.

Similar to the occlusion estimator network, we pass as additional input to $\mathcal{S}^l$ (at each pyramid level l except the top), the features $\mathbf{h}^{l+1}$ from the penultimate convolutional layer of the network $\mathcal{S}^{l+1}$ (of the upper pyramid level) as well as its corresponding scene flow prediction $\mathbf{s}^{l+1}$. This maintains consistency across the pyramid levels and allows the entire framework to perform multi-scale reasoning.

Context Network. Similar to PWC-Net, our PWOC-3D employs a CNN with dilated convolutional layers to refine the flow estimation. The input to this network comprises the flow estimate $\mathbf{s}^2$ and the last feature map $\mathbf{f}^2$ from the scene flow estimator $\mathcal{S}^2$ of the lowest pyramid level. This context network consists of 7 convolutional layers with 3×3 filters with a stride and padding of 1. The number of filters at each layer are 128, 128, 128, 96, 64, 32 and 4 respectively. The dilation parameters used at each layer are 1, 2, 4, 8, 16, 1


(a) $\mathbf{I}_L^t$ (b) $\mathbf{I}_R^t$ (c) $\mathbf{I}_L^{t+1}$ (d) $\mathbf{I}_R^{t+1}$

(e) Optical flow prediction $\mathbf{u}_\Theta$. (f) First disparity prediction $d_{0\Theta}$. (g) Second disparity prediction $d_{1\Theta}$.

(h) Optical flow ground truth $\mathbf{u}_{\mathrm{GT}}$. (i) First disparity ground truth $d_{0\mathrm{GT}}$. (j) Second disparity ground truth $d_{1\mathrm{GT}}$.

(k) Occlusion mask for optical flow $\mathbf{o}_L^{t+1}$. (l) Occlusion mask for first disparity $\mathbf{o}_R^t$. (m) Occlusion mask for second disparity $\mathbf{o}_R^{t+1}$.

(n) Binary error map for optical flow. (o) Binary error map for first disparity. (p) Binary error map for second disparity.

Fig. 5. Visualization of predictions of PWOC-3D with occlusion masks and ground truth on a validation sample from KITTI. In the error maps (n), (o) (p), pixels which contribute to the KITTI outlier error are coloured in magenta, while those that do not are coloured green.

and 1 respectively. All layers use the leaky ReLU activation, except the last which does not employ any activation.

This network outputs a residual flow $\delta\mathbf{s}^2$ which is added to the scene flow prediction $\mathbf{s}^2$ to obtain the final prediction $\hat{\mathbf{s}}(\mathbf{x}) = \mathbf{s}^2(\mathbf{x}) + \delta\mathbf{s}^2(\mathbf{x})$.

Loss Function. To start the coarse-to-fine estimation scheme of PWOC-3D, we initialize the prediction at a hypothetical level 7 of the pyramid $\mathbf{s}^7$ with zeros. This has the effect that the features at the topmost level ($l = 6$) of the pyramid are not warped at all. Thus, the cost volume and occlusions are computed directly using the original feature maps. The rest of the pipeline progresses as described in the previous sections.

We employ a multi-scale weighted loss function with intermediate supervision which penalizes losses at each level of the pyramid. Let Θ denote the set of all trainable parameters in the entire network and let $\mathbf{s}^l_{\mathrm{GT}}$ be the ground truth scene flow field subsampled to the resolution of pyramid level l, leading to the loss function

$$\begin{aligned}\mathcal{L}(\Theta) = &\underbrace{\sum_{l=3}^{6} \alpha_l \sum_{\mathbf{x}} |\mathbf{s}^l_\Theta(\mathbf{x}) - \mathbf{s}^l_{\mathrm{GT}}(\mathbf{x})|_2}_{\text{pyramid levels except the lowest}} \\ &+ \underbrace{\alpha_2 \sum_{\mathbf{x}} |\hat{\mathbf{s}}_\Theta(\mathbf{x}) - \mathbf{s}^2_{\mathrm{GT}}(\mathbf{x})|_2}_{\text{bottom pyramid level 2}} + \underbrace{\gamma |\Theta|_2}_{\text{L2 regularization}} ,\end{aligned} \tag{6}$$

where $|\cdot|_2$ denotes the L2 norm of a vector.

This enables the entire PWOC-3D model: The feature pyramid network, the occlusion mechanism with its occlusion estimator networks at different pyramid levels, the scene flow estimator networks at each pyramid level, and the context network to be trained in an end-to-end manner.

IV. EXPERIMENTS AND RESULTS

Datasets. The primary focus of this work is to estimate scene flow for automotive applications, thus making the KITTI dataset [16] a natural choice. However, KITTI provides only 200 training sequences (with sparse ground truth only), which is not sufficient to train a deep neural network. To overcome this problem, we first pretrain PWOC-3D on the synthetic (but large) FlyingThings3D dataset [28], and then finetune this model on KITTI. This transfer learning approach helps avoiding the network from being overfit on KITTI. We train PWOC-3D for 760 epochs on FlyingThings3D and for 125 epochs on KITTI.

Training Details. We use the photometric data augmentation strategy of FlowNet [29], combined with random vertical flipping (the latter only for FlyingThings3D, and not for KITTI). Since FlyingThings3D also provides bidirectional scene flow annotations, we utilise this by random temporal flipping of the training sample from ($\mathbf{I}_L^t$, $\mathbf{I}_R^t$, $\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$) to ($\mathbf{I}_L^{t+1}$, $\mathbf{I}_R^{t+1}$, $\mathbf{I}_L^t$, $\mathbf{I}_R^t$). We refrain from geometric transformations such as rotation, translation, etc which would destroy the epipolar constraint for disparity estimation across the stereo pairs.

TABLE I

EXPERIMENTAL RESULTS OF PWOC-3D. WE SHOW ENDPOINT ERROR (EPE [PX]) AND KITTI OUTLIER ERROR (KOE [%]) ON TRAINING, VALIDATION, AND TEST SET FOR KITTI AND FLYINGTHINGS3D. WE EVALUATE DIFFERENT COMPONENTS OF OUR CONTRIBUTION AND COMPARE TO END-TO-END SCENE FLOW NETWORKS FROM PREVIOUS WORK.

	FlyingThings3D						**KITTI**					
	Training		**Validation**		**Testing**		**Training**		**Validation**		**Testing**	
Architecture	EPE	KOE	EPE	KOE	EPE	KOE	EPE	KOE	EPE	KOE	EPE	KOE
PWOC-3D (*basic*)	8.25	25.15	9.79	25.01	23.38	26.01	1.97	6.09	3.71	13.6	–	–
PWOC-3D + FPN	6.17	19.89	8.40	20.35	**21.86**	21.22	**1.76**	**5.32**	3.39	13.97	–	–
PWOC-3D + FPN + Occ	**5.86**	**18.30**	**8.06**	**18.93**	22.01	**19.90**	1.85	5.69	**3.22**	**12.55**	–	**15.69**
SceneFlowNet [28]	–	–	11.24	–	–	–	–	–	–	–	–	–
Occ-SceneFlow [7]	–	–	–	–	–	–	–	–	–	–	–	11.34

TABLE II

A SNAPSHOT OF THE KITTI SCENE FLOW BENCHMARK'S LEADERBOARD AT THE TIME OF SUBMISSION. OUR METHOD IS THE MOST EFFICIENT IN TERMS OF RUNTIME.

Method	**D1-bg**	**D1-fg**	**D1-all**	**D2-bg**	**D2-fg**	**D2-all**	**Fl-bg**	**Fl-fg**	**Fl-all**	**SF-bg**	**SF-fg**	**SF-all**	**Runtime**
ISF [24]	4.12	**6.17**	4.46	**4.88**	**11.34**	**5.95**	5.40	**10.29**	**6.22**	**6.58**	**15.63**	**8.08**	600 s
PRSM [31]	**3.02**	10.52	**4.27**	5.13	15.11	6.79	**5.33**	13.40	6.68	6.61	20.79	8.97	300 s
OSF+TC [19]	4.11	9.64	5.03	5.18	15.12	6.84	5.76	13.31	7.02	7.08	20.03	9.23	3000 s
OSF 2018 [17]	4.11	11.12	5.28	5.01	17.28	7.06	5.38	17.61	7.41	6.68	24.59	9.66	390 s
SSF [25]	3.55	8.75	4.42	4.94	17.48	7.02	5.63	14.71	7.14	7.18	24.58	10.07	300 s
OSF [16]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	18.92	7.83	7.01	26.34	10.23	3000 s
FSF+MS [22]	5.72	11.84	6.74	7.57	21.28	9.85	8.48	25.43	11.30	11.17	33.91	14.96	2.7 s
PWOC-3D (ours)	4.19	9.82	5.13	7.21	14.73	8.46	12.40	15.78	12.96	14.30	22.66	15.69	**0.13 s**
CSF [20]	4.57	13.04	5.98	7.92	20.76	10.06	10.40	25.78	12.96	12.21	33.21	15.71	80 s
SFF [23]	5.12	13.83	6.57	8.47	21.83	10.69	10.58	24.41	12.88	12.48	32.28	15.78	65 s
PR-Sceneflow [32]	4.74	13.74	6.24	11.14	20.47	12.69	11.73	24.33	13.83	13.49	31.22	16.44	150 s

The hyperparameter d_{max} in the cost volume layer is set to 4. The weights used in the loss function $\alpha_2, \alpha_3, \ldots, \alpha_6$ are 0.32, 0.08, 0.02, 0.01 and 0.005 respectively. The regularization parameter γ is set to 0. Further, the ground truth scene flow is downscaled by 20 as in [29], and is downsampled to different resolutions to compute the training signal at different scales. During inference, all scene flow predictions are made at the input resolution. We use the Adam [33] optimizer to train PWOC-3D with the default setting of hyperparameters as recommended in [33]. We use a learning rate of $\lambda = 10^{-4}$.

Quantitative Analysis. As evident from Table I, the PWOC-3D model with the occlusion mechanism and the improved feature pyramid is the best performing network among all the variants. This is due to its well-localized and semantically strong features and its ability to mask the incorrect matching costs from the cost volume, thus preventing them from having adverse effects on the reasoning of the network. Table I also shows the comparison to the only two other end-to-end scene flow networks. PWOC-3D outperforms [28] on FlyingThings3D in terms of endpoint error and is 48 times smaller than the network from [7] (PWOC-3D has only 8,046,625 trainable weights).

As depicted in Figure 5, the occlusion maps contain clear signs of the masking effect. Specifically, the occlusion map for d_0 shown in Figure 5(l) contains areas occluded only along the horizontal epipolar line, while the maps for optical flow and d_1 in Figures 5(k) and 5(m) respectively model the occlusion caused due to the ego-motion of the camera in addition to the occlusion arising because of motion. This demonstrates the significant impact of our occlusion mechanism in autonomous driving scenarios.

The network with the FPN shows improvement in performance over the network without. The endpoint error of all the networks on the test split of FlyingThings3D [28] is higher than that on the validation split as visible in Table I. This is because in the test set, FlyingThings3D contains a set of objects and backgrounds which are disjoint from the training set. Thus, the samples are considerably different from those that the networks have been trained on.

Another interesting result is that among all variants in Table I, the difference between the validation and training errors is the lowest for PWOC-3D with the feature pyramid connections and the occlusion mechanism, particularly on KITTI where the training data is very limited. Thus, our occlusion reasoning scheme also helps reduce overfitting.

Table II shows the ranking of our method among the top 10 published methods on the KITTI scene flow benchmark's leaderboard at the time of our submission. As visible, PWOC-3D has a significantly lower runtime (0.13 s per frame on a GeForce GTX 1080 Ti) than all other methods, thus making it suitable for real-time applications. Among the listed methods, PWOC-3D is the only approach with that property. Further, our approach is especially accurate in the important foreground regions of moving objects. In summary, PWOC-3D has a unique mixture of characteristics with competitive accuracy, small network size, and low runtime.

V. CONCLUSION

In this paper we proposed PWOC-3D, a novel end-to-end CNN pipeline to predict scene flow (optical flow and stereo disparity jointly) directly from stereo image sequences. Our approach was significantly more efficient than earlier classical approaches, and much more accurate than variational methods. Moreover, unlike most previous techniques, PWOC-3D does not make any assumptions about the consistency or smoothness of motion, or the rigidity of objects. This makes our method more general and applicable to realistic scenarios where such assumptions do not hold, e.g. highly dynamic road scenes.

Moreover, PWOC-3D employs special constructs such as pyramid processing, warping and occlusion reasoning to tackle common challenges in scene flow like large motion and occlusions. In this regard, we proposed a novel self-supervised scheme to estimate occlusion from images without any labeled occlusion data. PWOC-3D demonstrates competitive results on the KITTI benchmark and the FlyingThings3D dataset. Notably, our method has significantly fewer parameters than contemporary methods and achieves second place on KITTI among end-to-end deep learning methods with 48 times fewer parameters than the top-performing method. PWOC-3D, along with our self-supervised occlusion scheme, can be combined with an unsupervised reconstruction loss (similar to [5], [6]) to result in a fully self-supervised end-to-end CNN which predicts scene flow. This requires a detailed and comprehensive study and is left for future work.

Acknowledgment

This work was partially funded by the Federal Ministry of Education and Research Germany as part of the project VIDETE (01IW18002).

References

[1] R. Caruana, "Multitask learning," *Machine learning*, 1997.
[2] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[3] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[4] J. Hur and S. Roth, "Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation," in *International Conference on Computer Vision (ICCV)*, 2017.
[5] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Conference on Artificial Intelligence (AAAI)*, 2018.
[6] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[7] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *European Conference on Computer Vision (ECCV)*, 2018.
[8] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, 1981.
[9] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *International Conference on Computer Vision (ICCV)*, 2007.
[10] E. Herbst, X. Ren, and D. Fox, "RGB-D flow: Dense 3-d motion estimation using color and depth," in *International Conference on Robotics and Automation (ICRA)*, 2013.
[11] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *International Journal of Computer Vision (IJCV)*, 2013.
[12] J. Park, T. H. Oh, J. Jung, Y.-W. Tai, and I. S. Kweon, "A tensor voting approach for multi-view 3d scene flow estimation and refinement," in *European Conference on Computer Vision (ECCV)*, 2012.
[13] X. Zhang, D. Chen, Z. Yuan, and N. Zheng, "Dense scene flow based on depth and multi-channel bilateral filter," in *Asian Conference on Computer Vision (ACCV)*, 2012.
[14] D. Ferstl, C. Reinbacher, G. Riegler, M. Rüther, and H. Bischof, "aTGV-SF: Dense variational scene flow through projective warping and higher order regularization." in *International Conference on 3D Vision (3DV)*, 2014.
[15] J. Quiroga, T. Brox, F. Devernay, and J. Crowley, "Dense semi-rigid scene flow estimation from rgbd images," in *European Conference on Computer Vision (ECCV)*, 2014.
[16] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[17] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.
[18] M. Menze, C. Heipke, and A. Geiger, "Discrete optimization for optical flow," in *German Conference on Pattern Recognition (GCPR)*, 2015.
[19] M. Neoral and J. Šochman, "Object Scene Flow with Temporal Consistency," in *Computer Vision Winter Workshop (CVWW)*, 2017.
[20] Z. Lv, C. Beall, P. F. Alcantarilla, F. Li, Z. Kira, and F. Dellaert, "A continuous optimization approach for efficient and accurate scene flow," in *European Conference on Computer Vision (ECCV)*, 2016.
[21] C. Vogel, K. Schindler, and S. Roth, "3D scene flow estimation with a piecewise rigid scene model," *International Journal of Computer Vision (IJCV)*, 2015.
[22] T. Taniai, S. N. Sinha, and Y. Sato, "Fast multi-frame stereo scene flow with motion segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[23] R. Schuster, O. Wasenmüller, G. Kuschk, C. Bailer, and D. Stricker, "SceneFlowFields: Dense interpolation of sparse scene flow correspondences," in *Winter Conference on Applications of Computer Vision (WACV)*, 2018.
[24] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?" in *International Conference on Computer Vision (ICCV)*, 2017.
[25] Z. Ren, D. Sun, J. Kautz, and E. Sudderth, "Cascaded scene flow prediction using semantic segmentation," in *International Conference on 3D Vision (3DV)*, 2017.
[26] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[28] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[29] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *International Conference on Computer Vision (ICCV)*, 2015.
[30] R. Schuster, O. Wasenmüller, C. Unger, and D. Stricker, "SDC - Stacked dilated convolution: A unified descriptor network for dense matching tasks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[31] C. Vogel, K. Schindler, and S. Roth, "3D scene flow estimation with a piecewise rigid scene model," *International Journal of Computer Vision (IJCV)*, 2015.
[32] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *International Conference on Computer Vision (ICCV)*, 2013.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations (ICLR)*, 2015.