A Deep Temporal Fusion Framework for Scene Flow Using a Learnable Motion Model and Occlusions

René Schuster[1] Christian Unger[2] Didier Stricker[1]
[1]DFKI - German Research Center for Artificial Intelligence [2]BMW Group
firstname.lastname@{dfki,bmw}.de

Abstract

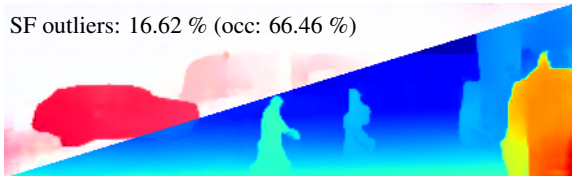
Motion estimation is one of the core challenges in computer vision. With traditional dual-frame approaches, occlusions and out-of-view motions are a limiting factor, especially in the context of environmental perception for vehicles due to the large (ego-) motion of objects. Our work proposes a novel data-driven approach for temporal fusion of scene flow estimates in a multi-frame setup to overcome the issue of occlusion. Contrary to most previous methods, we do not rely on a constant motion model, but instead learn a generic temporal relation of motion from data. In a second step, a neural network combines bi-directional scene flow estimates from a common reference frame, yielding a refined estimate and a natural byproduct of occlusion masks. This way, our approach provides a fast multi-frame extension for a variety of scene flow estimators, which outperforms the underlying dual-frame approaches.

1. Introduction

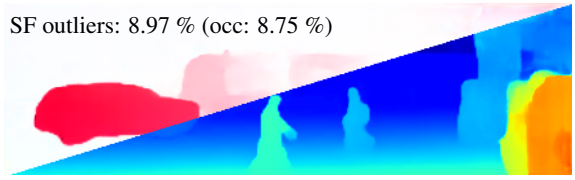
The estimation of motion is important in many applications such as autonomous or assisted driving, robot navigation, and others. A representation of motion in 2D image space (optical flow) is only a proxy for real world motion in the 3D world. Scene flow is the estimation of 3D geometry and 3D motion and as such a much richer and realistic representation. However, due to its higher complexity and its requirements on sensors, it is less often applied. Since the beginnings of scene flow estimation, major progress has been achieved. Most recently, data-driven deep learning approaches have pushed the limits of scene flow estimation even further [1, 8, 12, 21, 30]. These approaches achieve state-of-the-art results at run times close to real time. Yet, none of these deep learning methods utilizes a multi-frame setup which was shown to improve over a conceptually similar dual-frame approach for heuristic algorithms [18, 23, 26, 29]. Many of these traditional, heuristic approaches use the additional information from multiple views as a kind of regularization during matching, making them more complex and reliable on specific, simplified motion models (*e.g.* a constant motion assumption). At the same time, all previous approaches (even multi-frame

(a) Reference Image

(b) Fusion Weights



(c) Dual Frame Result from [21]



(d) Our Fusion Result

Figure 1: Our deep temporal fusion (DTF) refines an initial dual-frame estimate by combination with an inverted backward scene flow. The fusion is realized as a pixel-wise weighted averaging and thus yields (soft) occlusion maps. This way, the initial results are significantly outperformed, especially in the difficult occluded areas.

based) perform considerably worse in occluded areas (cf. Table 2), which suggests that there is a lot of unused potential in multi-frame scene flow estimation.

More generic concepts for learning-based multi-frame settings were proposed in the context of optical flow [10, 14, 19, 20]. But these methods do not model the underlying issue of occlusions at all, or tackle the estimation of occlusions by bi-directional flow estimation (twice as much effort).

In our work, we present the first deep fusion strategy for scene flow which is using a trainable, flexible motion model that exploits the geometric 3D information for self-supervised estimation of occlusion during temporal fusion (see Figure 2). Our framework overcomes some issues of previous work by the following contributions:

1. It introduces a dedicated sub-network to temporally invert motion in the opposite direction of the target flow using a learned, flexible model of motion.

2. It combines an initial estimate of forward scene flow with the inverted backward scene flow using a weighted average which results in the estimation of occlusions without explicit supervision.

3. This way, the fused results show superior performance over the underlying dual-frame scene flow algorithms, especially in occluded areas.

Additionally, our framework can be used together with any auxiliary scene flow estimator.

2. Related Work

Scene Flow Estimation. The history of scene flow estimation began with early variational methods inspired by optical flow estimation [5, 27]. Many variants were presented for different sensor setups like RGBD [4, 7]. But all those methods are subjected to the requirements of the variational framework (small motions, good initialization) or of the hardware (*e.g.* indoor environment for active depth cameras). Within the classical sensor setup of stereo cameras, a big step forward was achieved by the introduction of the piece-wise rigid scene model [2, 11, 17, 28, 29]. However, these heuristic approaches presume local planarity and rigidity and lead to considerably long computation times.

A boost in run time was achieved with the introduction of the first deep learning algorithms due to the massive parallelization on GPUs. At the same time, many of the newly proposed deep neural networks reached state-of-the-art results despite the lack of realistic, labeled training data [1, 8, 12, 21, 30]. Yet, no existing deep learning architecture for scene flow estimation makes use of the multi-frame nature of image sequences, which naturally exist in realistic applications. Our approach fills this gap with a trainable, generic multi-frame solution for scene flow estimation.

Classical, heuristic approaches have shown that the transition from a single temporal frame pair to two (or more) is expected to improve the results [18, 23, 26, 29]. However, all of these methods model the temporal relation of neighboring time frames as constant motion. Our proposed framework distills a generic motion model from data.

Deep Multi-Frame Models for Optical Flow. For optical flow there exists some previous work on deep multi-frame neural networks. MFF [20] computes forward flow for two consecutive time steps together with a backward flow for the central frame. The backward flow is used to warp the previous forward motion towards the reference frame realizing a constant motion assumption. A fusion network then combines the initial forward prediction and the warped one. Occlusions are not modeled explicitly here. ContinualFlow [19] uses previous flow estimates as additional input during the estimation for the current time step. Here, occlusions are learned as attention maps in a self-supervised manner similar to MaskFlownet [31] or PWOC-3D [21], but based on a cost volume instead of image features. ProFlow [14] proposes an online inverter for motion that is trained for every frame on the fly. In our work, we adopt this idea to avoid warping, but we only train a single inverter once to further avoid the re-training on every sample and the explicit estimation of occlusions at an early stage. In SelFlow [10] as in ProFlow also, occlusions are detected by a forward-backward consistency check. SelFlow uses the additional multi-frame information by constructing cost volumes for forward and backward direction which are then used for the flow estimation.

Our work gets rid of any consistency checks, avoids warping to shift the handling of occlusions to a later stage, and learns a dedicated universal model for the inversion of motion. Contrary to all mentioned cases, we propose a deep multi-frame model for the more complex problem of scene flow estimation.

3. Deep Multi-Frame Scene Flow

Consider a stream of stereo image pairs I_t^l and I_t^r for left and right camera at a given time t. For our framework, we tackle the problem of scene flow estimation with respect to a reference view (left at time t) into the future (time $t+1$). While dual-frame solutions only consider the four images at these two time steps, a multi-frame method incorporates information from at least one additional time (usually $t-1$ to avoid delay in the prediction and account for the symmetry in motion). Our framework builds on this exact setup using three stereo pairs at time $t-1$, t, and $t+1$. The idea is outlined in Figure 2 and can be summarized as follows. We use an arbitrary auxiliary model for scene flow estimation to predict forward ($t \rightarrow t+1$) and backward ($t \rightarrow t-1$) scene flow with respect to our reference view. This avoids

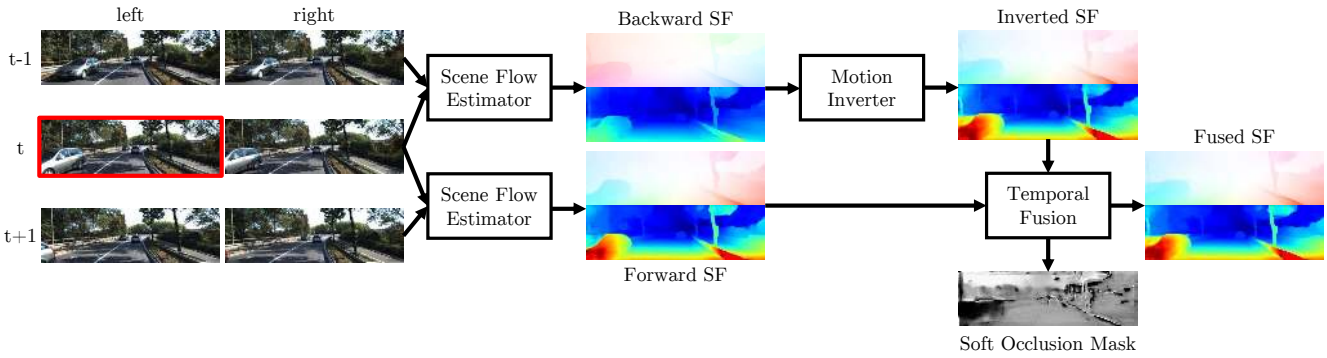


Figure 2: Overview of our proposed framework for deep temporal fusion with our trainable motion model.

any form of warping and thus postpones the problem of occlusions. Then, we learn a motion model that transforms the backward estimate into a forward motion. Finally, a temporal fusion module combines the forward and transformed backward estimate to obtain a refined result. For the fusion, we use a strategy of weighted averages. This implicitly yields soft occlusion maps for the two motion directions without explicit supervision on occlusions. The underlying dual-frame model that we mainly use is PWOC-3D [21] due to its simple training schedule compared to other approaches. However, in our experiments (Section 4.2) we show that our framework is not limited to this model. The novel sub-networks for motion inversion and fusion are presented in more detail in the next sections.

3.1. Temporal Scene Flow Inversion

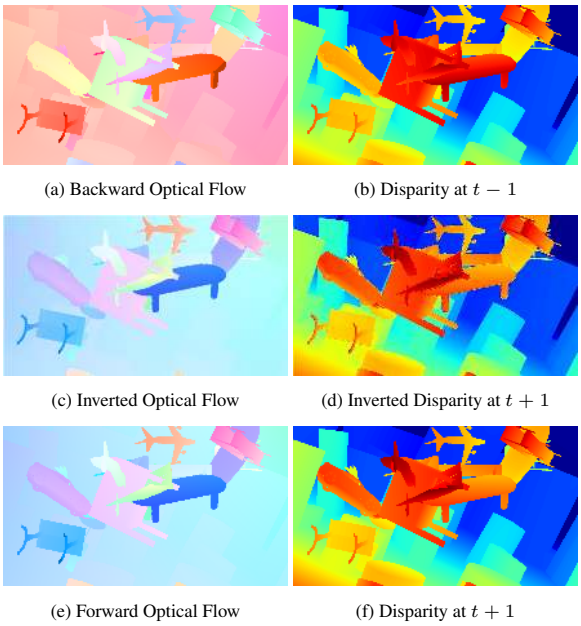
Instead of a constant motion assumption, which is often applied in previous work, we create and train a compact neural network that utilizes a learned motion model to temporally invert scene flow. Our architecture is inspired by the inversion module of [14] but we make it deeper since for our framework we want to learn a generic model that can invert motion for arbitrary sequences without the need of re-training on every frame. In detail, the inversion sub-network consists of 4 convolutional layers with kernel size 3×3 and a fifth one with a 7×7 kernel and output feature dimensions of $16, 16, 16, 16, 4$ respectively. The last layer is activated linearly. Similarly to [14], we equip our inverter with a mechanism for spatial variance by concatenating the input scene flow with normalized ($[-1, 1]$) spatial image coordinates of x- and y-direction. This way and together with the depth information from the backward scene flow, the inversion network is able to operate fully in (hypothetical) 3D space. For a qualitative impression of our inverter, Figure 3 visualizes the results for a validation sample.

3.2. Deep Forward-Backward Fusion

After the prediction of scene flow in the forward and backward direction (using the same reference frame) and inverting the backward estimate, we can merge the forward and inverted backward prediction. The refined results can potentially overcome errors in difficult regions of occlusion or out-of-view motion, because occlusions occur rarely across multiple views [23]. Our fusion strategy follows a weighted average approach, where a fusion module predicts pixel-wise weights (that sum up to one) for the combination of the original forward estimate and the inverted backward scene flow. Interestingly, these weights correspond to (soft) occlusion masks, revealing the main reason why the inverted backward motion should be preferred over a forward dual-frame estimate (cf. Figures 1 and 2). While the direct prediction of a refined (or residual) scene flow during fusion is also possible, this would neither model the underlying issue nor produce occlusion masks.

For our fusion module, we adopt the architecture of the context network of PWC-Net [25] and PWOC-3D [21]. It consists of seven convolutional layers with a kernel size of 3×3, output depth of $32, 64, 128, 128, 64, 32, 2$, and dilation rates of $1, 2, 4, 8, 16, 1, 1$ respectively. The last layer predicts pseudo probabilities in a one-hot encoding for the forward and inverted backward scene flow which are used for weighted averaging after a softmax activation. As input for this module, we concatenate the forward and inverted backward estimate.

Described above is a simple baseline for temporal fusion of scene flow (*basic*). Within the experiments in Section 4.4 we will compare different variants of our fusion module. Though the network can detect occlusion based on the depth (disparity) and motion of neighboring pixels, it can not estimate out-of-view motion without knowing where the field of view ends. This information could be guessed from the padding during convolution, however for more explicit modeling we again feed additional spatial information to the module, similar as with the inverter. We denote this variant as *spatial*. Another variant is again motivated by the issue of occlusion. Since in multiple views different parts of a reference image are occluded, we argue that the predicted occlusion masks (fusion weights) should



(a) Backward Optical Flow (b) Disparity at $t-1$

(c) Inverted Optical Flow (d) Inverted Disparity at $t+1$

(e) Forward Optical Flow (f) Disparity at $t+1$

Figure 3: An example of the learned inversion of motion on data of FlyingThings3D [16]. The left and right columns show the optical flow and disparity at $t+1$ components of the scene flow. The first and last rows give the ground truth in backward and forward direction respectively. The center row presents the results of our generic motion inverter.

differ for the different components of the scene flow, *e.g.* between left and right view of a stereo camera, there are no occlusions due to motion. Therefore this variant is predicting a separate occlusion map for each channel of our scene flow representation (in image space) and is depicted as *4ch* since it predicts fusion weights for four scene flow channels (two for optical flow and two for initial and future disparities). Lastly, we combine both strategies and name the combination *spatial-4ch*. In Figures 1 and 2, the occlusion maps (fusion weights) for the *basic* variant are shown for the sake of clarity and space.

4. Experiments

Our experiments and results are split into three sets with the following main intentions. First of all, we validate that the overall framework improves over the initial dual-frame estimates of different auxiliary scene flow models. Secondly, we compare our work to existing multi-frame scene flow algorithms using the official public KITTI benchmark [3, 17]. Lastly, our goal is to validate each step of our framework separately by means of an extensive ablation study.

As metric, the common KITTI outlier rate is used which classifies per-pixel estimates as outliers if they deviate more than 3 px and 5 % from the ground truth. This metric is computed for the different components of our scene flow, *i.e.* initial disparity (*D1*), next disparity (*D2*), optical flow (*OF*), or for the entire scene flow (*SF*) if either of the three previous components is classified as an outlier. All outlier rates are averaged over all valid ground truth pixels of the respective data split.

4.1. Data Sets and Training

Data Sets. For most of our experiments, the well-known KITTI data set is used [3, 17]. However, it is limited in size and thus inappropriate for the training of deep neural networks. Despite some success on unsupervised scene flow estimation [6] or knowledge distillation from teacher networks [1, 8], transfer learning by pre-training and fine-tuning is the most common strategy to overcome this issue [15, 21, 24, 25]. The one large-scale data set which provides labeled data for scene flow is FlyingThings3D (FT3D) [16]. In this work, it is also used for pre-training of some parts of the pipeline.

For validation, we split 20 sequences from the KITTI *training* subset as in [21] and the last 50 sequences from each subset *A*, *B*, and *C* of the FlyingThings3D *train* set.

Training and Implementation Details. Where required, the auxiliary scene flow estimators are initialized with the published pre-trained weights. We use the rich ground truth of FlyingThings3D [16] to separately pre-train the inverter on forward and backward ground truth motion with an L2-loss for 40 epochs with a batch size of 4 and an initial learning rate of 1×10^{-4} that we decrease to 5×10^{-5} and 1×10^{-5} after 20 and 30 epochs respectively. The rest of our pipeline is initialized from scratch.

Afterwards, we fine-tune our fusion pipeline on KITTI [17] for 100 epochs. The learning rate for fine-tuning starts at 5×10^{-5} and is again reduced after 75 epochs to 1×10^{-5}. Due to memory limitations, we use a batch size of 1 whenever the entire pipeline is used for training.

Unless mentioned otherwise, Leaky-ReLU [13] with a leak factor of 0.1 is used after each convolution. For all training stages, we use the ADAM optimizer [9] with its default parameters.

Our robust loss function for the 4-dimensional scene flow in image space is similar to the one in [21, 25] and defined by

$$\mathcal{L} = \frac{1}{N_{gt}} \cdot \sum_{\mathbf{x} \in gt} \left(|s(\mathbf{x}) - \hat{s}(\mathbf{x})|_1 + \epsilon \right)^{0.4} . \qquad (1)$$

Here s and $\hat{s}$ are the estimated and ground truth scene flow fields, $|\cdot|_1$ is the L_1-norm, $\epsilon = 0.01$ is a small constant for numerical stability, and the power of 0.4 gives less weight to strong outliers.

For the entire pipeline, we impose this loss on the forward estimate, the inverted backward scene flow, and the

Table 1: Comparison of our multi-frame fusion approach to the dual-frame results of the underlying auxiliary scene flow estimator for the entire image (*all*) and occluded areas only ($occ \in all \setminus noc$) on our KITTI validation split. The last column gives the maximum relative improvement of DTF over the respective dual-frame baseline.

Scene Flow Estimator	Setup	all				occ				max. rel. Improv.
		D1	D2	OF	SF	D1	D2	OF	SF	
SENSE [8]	Dual	0.97	2.22	3.00	4.04	2.08	8.23	7.19	11.84	
	Ours	0.97	1.66	3.01	3.52	2.05	4.81	7.21	8.57	41.6 %
OE [30]	Dual	1.11	2.58	5.56	6.61	2.53	7.34	15.06	17.73	
	Ours	1.12	2.46	5.46	6.39	2.54	6.97	14.57	16.86	5.0 %
DWARF [1]	Dual	2.35	3.49	7.07	8.16	3.94	7.59	17.70	19.63	
	Ours	1.17	2.63	5.64	6.75	2.82	7.54	14.90	17.82	50.2 %
PWOC-3D [21]	Dual	4.65	6.72	11.50	13.64	8.02	15.20	29.17	32.15	
	Ours	3.34	4.85	8.22	9.70	5.63	10.10	18.68	21.24	36.0 %
SFF [22]	Dual	6.61	10.28	12.39	15.76	9.94	19.57	26.08	30.74	
	Ours	6.04	9.03	11.43	14.30	8.77	15.91	22.85	26.25	18.7 %

(a) Reference Image (b) **Forward** Ground Truth (enhanced)

(c) SENSE [8] (d) OpticalExpansion [30] (e) DWARF [1]

(f) PWOC-3D [21], Original (g) PWOC-3D [21], Re-trained (h) SFF [22]

Figure 4: Visualization of the **backward** optical flow for different scene flow estimators. Most auxiliary estimators used in our experiments have difficulties with backward motion because they do not perform actual matching but rather rely on the image information of the reference frame alone, especially for street surfaces. Significant improvements are noticeable once the backward branch gets trained end-to-end in our framework (g), even though backward ground truth is not available.

final fusion:

$$\mathcal{L}_{total} = \mathcal{L}_{fw} + \mathcal{L}_{inv} + \mathcal{L}_{fused} \tag{2}$$

This multi-stage loss avoids that during training the fusion flips to one side and does not recover because the other side would not receive any updates anymore.

4.2. Comparison to the Auxiliary Estimators

In Table 1 we validate that our deep temporal fusion framework surpasses a diverse set of underlying dual-frame estimators in terms of scene flow outliers. Especially in the difficult areas of occlusion, our approach achieves significantly better results, reducing the scene flow outlier rate by up to ~ 30 %. The fusion improves the scene flow estimates for non-occluded areas also, resulting in an overall improvement over *all* image areas. For OpticalExpansion (OE) [30], the relative improvement is less compared to other auxiliary estimators. This has two reasons. First of all, some scene flow algorithms are heavily biased towards forward motions (cf. Figure 4) and therefore provide much less reliable information for fusion in the backward branch. Secondly, the estimate of motion-in-depth from OE is depending a lot on the optical flow estimate, which amplifies the previous limitation and expands it over the complete scene flow estimation in backward direction. The first reason additionally motivates an end-to-end training of the fusion framework

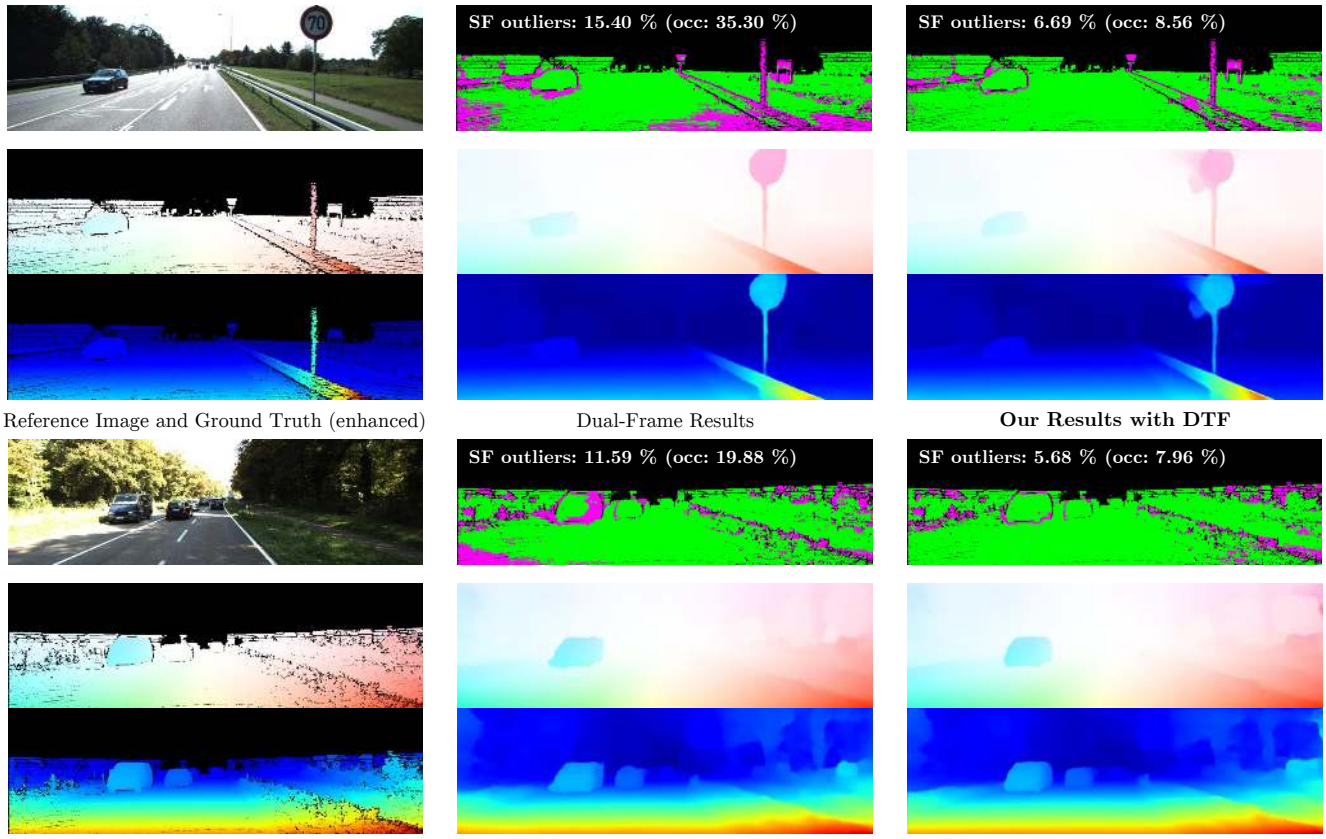


Figure 5: Visual comparison of our deep multi-frame fusion framework to the auxiliary dual-frame model PWOC-3D [21]. Scene flow results are shown by optical flow and disparity at $t + 1$. The error maps indicate scene flow outliers in magenta and inliers in green. Notice the improvements in occluded areas (*e.g.* in front and around of vehicles) or the out-of-view occlusions due to ego-motion (*e.g.* the close-by part of the guardrail in the first example and the lower image corners).

together with the auxiliary estimator. This is performed for PWOC-3D [21] because it is most easy to train. The other auxiliary estimators are used as off-the-shelf replacements with the officially provided pre-trained weights. Our framework is even able to improve non-learning-based results from SFF [22], with a noticeable margin of more than 10 % in occluded areas. Here, we account the smaller relative improvements to the ego-motion model that is applied in SFF which is able to estimate out-of-view motions in forward direction for the background more reliably. A visual comparison between PWOC-3D and the multi-frame extension by our framework is conducted in Figure 5.

4.3. Comparison to State-of-the-Art

To check the generalization of our model on more unseen data, we submit results obtained with our deep multi-frame model to the KITTI online benchmark. The results for all multi-frame methods and related dual-frame baselines are presented in Table 2. Due to the limited number of training samples on KITTI, some over-fitting can be observed when comparing the numbers to the results on our validation split. However, improvements over the underlying dual-frame models (SENSE and PWOC-3D) are still evident, again with margins of $\sim$ 15 - 20 % in occluded areas. Since KITTI evaluates the submitted results only for non-occluded (*noc*) and all valid pixels, the results for occluded areas (*occ*) are reconstructed from the available data. To this end, we compute the ratio of non-occluded image areas on the KITTI *training* set (84.3 %), and use this distribution to estimate results for only occluded areas for the KITTI *testing* set based on the benchmark results for non-occluded (*noc*) and *all* areas according to the following formula:

$$occ_r = \frac{all_r - noc_r \cdot 0.843}{0.157} \tag{3}$$

for the regions $r \in \{bg, fg, all\}$. This strategy reveals that even for the top performing multi-frame methods, moving vehicles which leave the field of view are the most challenging areas. In these regions (*occ-fg*), our fusion approach achieves top performance. It furthermore performs significantly better in foreground regions than the other multi-frame methods. Lastly, we highlight that since ours is the

Table 2: Results of the KITTI scene flow benchmark for all multi-frame approaches. We also provide results for the auxiliary scene flow methods used in our pipeline and conceptual dual-frame counterparts for other multi-frame methods, where existent. Scene flow outlier rates (*SF*) are presented for foreground (*fg*), background (*bg*), and all regions, as well as for non-occluded areas (*noc*), occluded areas only (*occ*, details in the text), and the union (*all*).

	Method	SF Outliers [%]									Run Time
		occ			noc			all			
		bg	fg	all	bg	fg	all	bg	fg	all	[s]
multi-frame	PRSM [29]	**12.36**	37.65	**15.74**	5.54	17.65	7.71	**6.61**	20.79	**8.97**	300
	DTF+SENSE (**Ours**)	16.37	**37.49**	19.65	6.69	**9.72**	**7.23**	8.21	**14.08**	9.18	0.76
	OSF+TC [18]	15.46	43.98	19.49	**5.52**	15.57	7.32	7.08	20.03	9.23	3000
	SFF++ [23]	26.40	48.36	30.91	9.84	21.04	11.55	12.44	25.33	14.59	78
	DTF+PWOC (**Ours**)	31.91	51.14	34.29	8.79	21.01	10.98	12.42	25.74	14.64	**0.38**
	FSF+MS [26]	21.59	65.48	27.63	9.23	28.03	12.60	11.17	33.91	14.96	2.7
dual-frame	SENSE [8]	17.22	44.86	21.63	6.71	10.02	7.30	8.36	15.49	9.55	0.32
	OSF [17]	15.01	47.98	19.41	5.52	22.31	8.52	7.01	26.34	10.23	3000
	PWOC-3D [21]	41.20	47.52	41.62	9.29	18.03	10.86	14.30	22.66	15.69	0.13
	SFF [22]	25.58	63.26	30.76	10.04	26.51	12.99	12.48	32.28	15.78	65
	PRSF [28]	41.09	58.82	42.80	8.35	26.08	11.53	13.49	31.22	16.44	150

Table 3: Evaluation of intermediate results in our pipeline on our KITTI validation split. For this experiment, PWOC-3D [21] is the auxiliary estimator and is trained end-to-end. The inversion module is separately evaluated on FlyingThings3D.

Output	all				occ			
	D1	D2	OF	SF	D1	D2	OF	SF
forward (fw)	3.47	5.83	8.95	10.76	5.89	14.39	23.17	26.93
inverted backward (bw-inv)	4.15	6.00	20.34	22.14	6.64	9.92	31.74	33.81
constant linear inversion (FT3D)	–	**1.27**	47.16	47.18	–	–	–	–
our inverter (FT3D)	2.19	3.25	**41.98**	**42.34**	–	–	–	–
fw + bw-inv + oracle	2.63	3.91	6.25	7.51	4.53	8.40	16.39	18.43
fw + bw-inv + fusion-basic	**3.22**	4.90	9.01	10.48	4.88	10.23	19.27	21.66
fw + bw-inv + fusion-spatial	3.48	5.51	8.85	10.55	6.13	13.66	22.23	25.40
fw + bw-inv + fusion-4ch	3.34	4.85	**8.22**	**9.70**	5.63	10.10	18.68	21.24
fw + bw-inv + fusion-spatial-4ch	3.43	**4.84**	8.67	10.19	**5.45**	**9.25**	**18.46**	**20.82**

first deep method for multi-view scene flow estimation, our run time is close-to real time and thus 2 to 5 orders of magnitude faster than that of most other multi-view methods. The inversion and fusion without auxiliary scene flow estimation takes 0.12 seconds. We use a Nvidia RTX 2080 Ti for inference.

4.4. Ablation Study

For completeness, each part of our framework is evaluated separately in Table 3. The first two rows show the results for the forward prediction and the inverted backward scene flow after end-to-end training. We can see that within our multi-view training, the plain forward prediction is already improved over the dual-frame baseline (cf. Table 1). Further, the results of the backward branch after inversion indicate that the motion inversion of optical flow is a bottleneck. Yet, for occluded areas the inversion outperforms the forward prediction already in terms of change of disparity, validating its importance. Both of these observations are confirmed by an evaluation of the inverter only on data of FlyingThings3D [16] as shown in the fourth row of Table 3 (cf. Figure 3) compared to a naïve constant linear motion assumption in 2D. This is, optical flow and change of disparity are multiplied by -1. Our learned motion model outperforms the constant motion model in terms of optical flow. Though, one might doubt whether the quality of the inversion is good enough to improve the forward prediction. Therefore, we compute an *oracle* fusion using the ground truth to select the better estimate from the forward and inverted backward branch. This experiment produces a theoretical bound for our fusion module and makes apparent that the inverted backward scene flow contains a lot of valu-

able information. Within the last four rows of Table 3 we compare the different variants of our fusion module as described in Section 3.2. The results in occluded areas reveal that all variants including the *basic* one effectively tackle the problem of occlusion. Among all, the *spatial* version performs the worst unless combined with the *4ch* variant. However, we could observe stronger over-fitting for this model with most representation power (and highest number of parameters). As a result, over the entire image area, the fusion module using four weight channels performs the best. Worth highlighting is that our fusion results in occluded areas reach the level of the oracle prediction almost.

5. Conclusion

In this work we have presented a straight-forward integration of multiple frames to improve scene flow estimates for a wide range of dual-frame algorithms. Significant improvements could be achieved by inverting the backward motion of the reference view and fusing it with an initial forward estimate. Moreover, our fusion strategy of weighted averages yields additional estimates of occlusion maps without the need for bi-directional consistency checks.

The experiments reveal that the inversion of optical flow is a limiting factor of the proposed approach, thus for future work we plan to equip the motion inverter with more domain knowledge to overcome this limitation and further to apply end-to-end training with other more complicated auxiliary estimators.

Acknowledgement

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

References

[1] Filippo Aleotti, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning end-to-end scene flow by distilling single tasks knowledge. In *Conference on Artificial Intelligence (AAAI)*, 2020.

[2] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D Flow: Dense 3-d motion estimation using color and depth. In *International Conference on Robotics and Automation (ICRA)*, 2013.

[5] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *International Conference on Computer Vision (ICCV)*, 2007.

[6] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense RGB-D scene flow. In *International Conference on Robotics and Automation (ICRA)*, 2015.

[8] Huaizu Jiang, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz. Sense: A shared encoder network for scene-flow estimation. In *International Conference on Computer Vision (ICCV)*, 2019.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2015.

[10] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. SelFlow: Self-supervised learning of optical flow. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Zhaoyang Lv, Chris Beall, Pablo F Alcantarilla, Fuxin Li, Zsolt Kira, and Frank Dellaert. A continuous optimization approach for efficient and accurate scene flow. In *European Conference on Computer Vision (ECCV)*, 2016.

[12] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep Rigid Instance Scene Flow. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

[14] Daniel Maurer and Andrés Bruhn. ProFlow: Learning to predict optical flow. In *British Machine Vision Conference (BMVC)*, 2018.

[15] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision (IJCV)*, 2018.

[16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas 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.

[17] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] Michal Neoral and Jan Šochman. Object Scene Flow with Temporal Consistency. In *Computer Vision Winter Workshop (CVWW)*, 2017.

[19] Michal Neoral, Jan Šochman, and Jiří Matas. Continual occlusion and optical flow estimation. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[20] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[21] Rohan Saxena, René Schuster, Oliver Wasenmuller, and Didier Stricker. PWOC-3D: Deep occlusion-aware end-to-end scene flow estimation. In *Intelligent Vehicles Symposium (IV)*, 2019.

[22] René Schuster, Oliver Wasenmüller, Georg Kuschk, Christian Bailer, and Didier Stricker. SceneFlowFields: Dense interpolation of sparse scene flow correspondences. In *Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[23] René Schuster, Oliver Wasenmüller, Christian Unger, Georg Kuschk, and Didier Stricker. SceneFlowFields++: Multi-frame matching, visibility prediction, and robust interpolation for scene flow estimation. *International Journal on Computer Vision (IJCV)*, 2020.

[24] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of CNNs for optical flow estimation. *arXiv preprint arXiv:1809.05571*, 2018.

[25] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Fast Multi-frame Stereo Scene Flow with Motion Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *International Conference on Computer Vision (ICCV)*, 1999.

[28] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *International Conference on Computer Vision (ICCV)*, 2013.

[29] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. *International Journal of Computer Vision (IJCV)*, 2015.

[30] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[31] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I-Chao Chang, and Yan Xu. MaskFlownet: Asymmetric feature matching with learnable occlusion mask. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.