M-FUSE: Multi-frame Fusion for Scene Flow Estimation

Lukas Mehl Azin Jahedi Jenny Schmalfuss Andrés Bruhn
Institute for Visualization and Interactive Systems, University of Stuttgart
{lukas.mehl,azin.jahedi,jenny.schmalfuss,andres.bruhn}@vis.uni-stuttgart.de

Abstract

Recently, neural network for scene flow estimation show impressive results on automotive data such as the KITTI benchmark. However, despite of using sophisticated rigidity assumptions and parametrizations, such networks are typically limited to only two frame pairs which does not allow them to exploit temporal information. In our paper we address this shortcoming by proposing a novel multi-frame approach that considers an additional preceding stereo pair. To this end, we proceed in two steps: Firstly, building upon the recent RAFT-3D approach, we develop an improved two-frame baseline by incorporating an advanced stereo method. Secondly, and even more importantly, exploiting the specific modeling concepts of RAFT-3D, we propose a U-Net architecture that performs a fusion of forward and backward flow estimates and hence allows to integrate temporal information on demand. Experiments on the KITTI benchmark do not only show that the advantages of the improved baseline and the temporal fusion approach complement each other, they also demonstrate that the computed scene flow is highly accurate. More precisely, our approach ranks second overall and first for the even more challenging foreground objects, in total outperforming the original RAFT-3D method by more than 16%. Code is available at `https://github.com/cv-stuttgart/M-FUSE`.

1. Introduction

Estimating the 3D motion field of objects in the 3D world from stereo or RGBD image sequences, the so-called scene flow, is one of the fundamental tasks in computer vision. Its fields of application range from robotics and automotive scenarios [20] over markerless motion capture for virtual and augmented reality [36] to action recognition and intention prediction [39].

Early works go back to the seminal approach of Vedula *et al*. [37] in the late nineties and since then variational methods have been among the leading techniques to solve this task; see *e.g*. [7, 38, 40]. Only recently, four years after their first application to scene flow estimation [19], neural networks have been able take the lead in dedicated benchmarks such as KITTI [20]; see *e.g*. the approaches in [13, 34, 42, 43]. This comparably late success of neural networks, however, is not surprising: Scene flow estimation has more degrees of freedom than other correspondence problems that only work in 2D or 1D, such as optical flow and stereo, hence solving this task requires more sophisticated ideas and more complex network architectures.

One way to deal with these additional degrees of freedom is to use semantic information. This information can be given in terms of object models [20] or instance segmentations [2, 15, 43]. Another way is to rely on point-wise [34] or segment-wise rigidity priors [4, 15, 20, 38], or to explicitly learn segmenting rigid motions [43]. In combination with semantic information such rigidity estimates allow to assign rigid motions to all independently moving objects and to the background [15, 43]. And finally, it is also possible to reduce the difficulty of the problem. This can either be done by decoupling stereo and 3D motion estimation [1, 13, 34, 40, 42], which also enables the use of dedicated state-of-the-art algorithms for stereo, or by directly relying on RGBD footage [4, 5, 23, 24], *e.g*. by using time-of-flight cameras or LiDAR [13].

In view of the aforementioned progress in neural networks for scene flow estimation, it is remarkable that currently leading methods [1, 12, 13, 15, 34, 42, 43] do not exploit potentially valuable temporal information to further improve the results. In fact, while differing in the actual inputs – monocular images [1, 42], stereo pairs [12, 13, 15, 42, 43, 34], RGBD images [34] or LiDAR point clouds [13] – all leading networks are restricted to the standard two-frame setting. In this context, it is also surprising that the best multi-frame scene flow method on the KITTI benchmark is still a classical variational method which dates back to 2015 [38]. This illustrates that developing suitable multi-frame extensions of existing network architectures is indeed a difficult task.

The latter observation is also reflected in the recent literature on multi-frame scene flow networks [27, 8]. On the one hand, on the KITTI benchmark, only slight improvements of 2%-4% have been reported compared to the underlying two-frame baselines[1]. Evidently, for recent multi-frame architectures, the often much larger training gains do not generalize well to the actual test data. On the other hand, the proposed multi-frame concepts were either not incorporated into state-of-the-art baselines [27] or they were developed for the even more challenging self-supervised monocular setting [8]. This in turn gives an explanation for the relatively poor overall performance of recent multi-frame methods compared to currently leading supervised two-frame approaches. And finally, as of today, the accuracy of leading two-frame approaches in general has improved by a factor two compared to the baseline in [27]; see *e.g.* [13, 43, 33]. This in turn raises the question if suitable multi-frame extensions can be developed at all, if the underlying baseline already provides a sufficiently high accuracy.

Contributions. In our paper, we show that multi-frame ideas are still valuable in the context of recent high-accuracy networks. Building upon the RAFT-3D method [34], we present a novel multi-frame approach that allows to leverage the performance of current two-frame techniques. In this context we make the following contributions: (i) We propose a multi-frame architecture that particularly exploits the advantages of the underlying RAFT-3D architecture by combining a $SE(3)$ based prediction step with a U-Net based fusion architecture. In this context, we also improve the underlying two-frame baseline by substituting the employed stereo approach. (ii) Performing ablation studies and further experiments based on fourfold cross validation, we illustrate the benefits of the different architectural components of our method. In this way, we identify a fusion strategy that generalizes well to the test data. (iii) With improvements of 9% for the baseline and 16% for the overall approach, we report much larger performance gains than existing multi-frame networks from the literature. These gains also lead to highly competitive results, eventually ranking second in the KITTI scene flow benchmark.

2. Related work

Multi-frame scene flow. Regarding the use of multiple time frames for scene flow estimation, one can mainly distinguish three types of methods. Like our approach, most of them rely on a three frame setting that has proven to be a good compromise between available temporal information and efficiency for both optical flow [17, 18, 25, 41] and scene flow [8, 21, 27, 28, 32].

(i) On the one hand, there are approaches that explicitly model multi-frame scene flow in terms of an *energy minimization* framework. Such approaches are the method of Vogel *et al.* [38] that, based on piece-wise rigidity assumption, enforces a consistent piece-wise planar segmentation over time, the method of Golyanik *et al.* [4] that follows a similar idea but relies on RGBD data instead of stereo sequences, the method of Taniai *et al.* [32] which fuses estimates from optical flow and multi-frame time stereo, and the method of Neoral and Šochman [21] that extends the two-frame scene flow approach of Menze and Geiger [20] by additionally propagating object labels over time.

(ii) On the other hand, there are *sparse-to-dense* methods that speed up the computation of energy-based methods by considering sparse matching strategies followed by a robust interpolation step. Such a method is the approach of Schuster *et al.* [28], which performs a sparse multi-frame matching relying on the assumption that the 3D motion in terms of the scene flow is constant over time.

(iii) And finally, also *neural networks* gained recently popularity in the context of multi-frame scene flow. Such methods include another approach of Schuster *et al.* [27] that predicts the forward from the backward flow based on a small learned motion inverter and subsequently fuses both flows using a convex fusion step, and the self-supervised monocular approach of Hur and Roth [8] that uses a convolutional LSTM to encourage consistency over time.

While our multi-frame method is also based on a neural network that fuses flow estimates, its underlying strategy differs significantly from the one in [27]. On the one hand, our method not only relies on a much more advanced baseline, *i.e.* RAFT-3D. Its entire architecture is also specifically tailored towards this baseline; *e.g.* our method exploits both the local $SE(3)$ parametrization as well as the valuable inputs of RAFT-3D's recurrent unit when adaptively integrating temporal information. On the other hand, instead of learning a motion model via a small motion inverter that is naturally limited in its generalization capabilities and subsequently restricting the fusion to a convex combination, our method predicts the motion using a SE(3)-based extrapolation and then considers a more generalized U-Net based fusion step. While the SE(3)-based prediction holds in many scenarios, the generalized fusion step allows to implicitly learn possibly required corrections of this prediction.

Multi-frame optical flow. In contrast to scene flow estimation (3D motion), there is already an extensive literature on *multi-frame neural networks* in the closely related field of optical flow estimation (2D motion = projected scene flow). In this context, one can roughly distinguish three strategies:

(i) A first common strategy is to rely on a relaxed version of a *constant motion model*. This can either be achieved by adding a temporal smoothness constraint to the loss such as the unsupervised method of Janai *et al.* [10] or, even less strict, by initializing the estimation with the motion-compensated flow from the previous time step (warm start) as employed by the method of Teed and Deng [33].

(ii) A second strategy is to estimate multiple flows in a

[1] We considered from both papers the best overall results in terms of the standard SF-all outlier measure: [27]: DTF-SENSE (9.18) vs. SENSE (9.55), [8]: Multi-Mono-SF-ft (33.09) vs. Self-Mono-SF-ft (33.88).

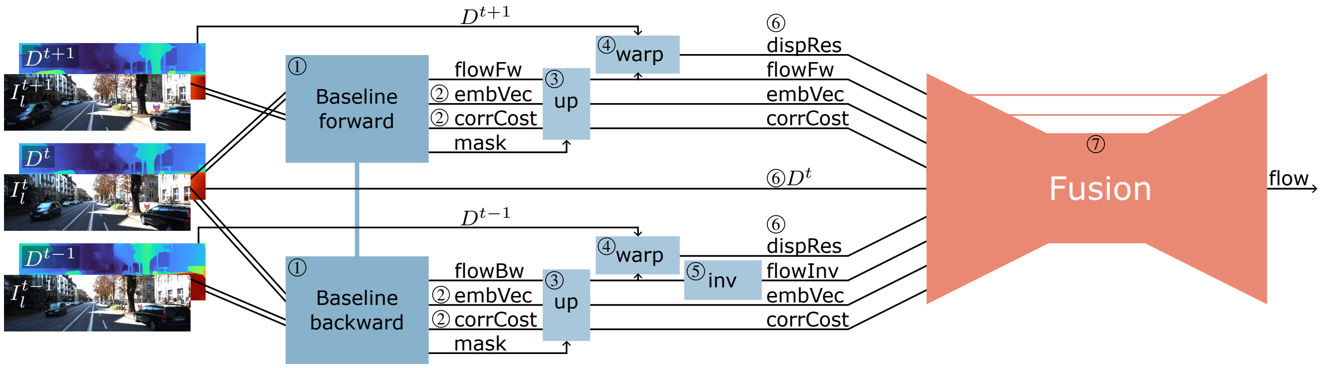

Figure 1. Overview of our M-FUSE approach (see Sec. 3.2). We employ two shared instances of our baseline model to predict forward ($t \rightarrow t+1$) and backward ($t \rightarrow t-1$) scene flow as well as additional features used in our fusion U-Net to predict the final flow estimate.

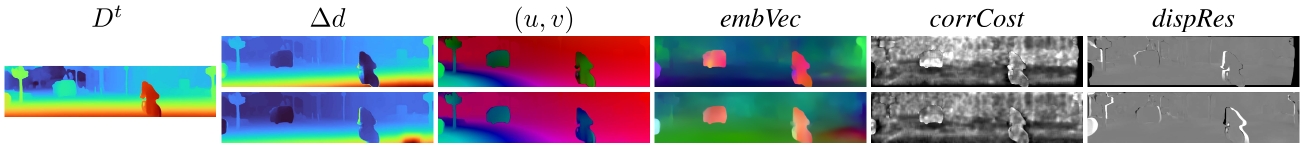

Figure 2. Input features to our fusion module. *From left to right:* Disparity in the reference frame D^t, disparity change Δd, optical flow (u, v), rigid-motion embedding vectors, correlation cost, disparity residuals. *Top row:* features from forward direction, *bottom row:* backward direction. We use color visualizations for disparity/optical flow, and PCA to reduce the 16-channel embedding vectors to RGB.

joint recurrent unit ($\neq$ shared recurrent unit). A corresponding self-supervised approach has been proposed by Liu *et al*. [14] who estimate forward and backward flows with a double cost volume, which allows to learn flow in occluded regions without an explicit motion model.

(iii) A third strategy seeks to improve the estimation by incorporating a *learned motion model*. To this end, Maurer and Bruhn [17] proposed an approach that integrates predictions from a motion model that is learned with a small neural network on the fly. Similarly, Stone *et al*. [30] incorporated this idea in an unsupervised method, where the motion model helps to teach the flow in occluded regions. Alternatively, instead of learning the model directly, Ren *et al*. [25] fuse a forward flow estimate based on a constant motion model, but allow corrections in a learned follow-up fusion step. To this end, they employ the fusion module from FlowNet2 [9] originally designed for combining small and large displacements.

While our approach is similar to [25] in the sense that it also relies on a U-Net architecture, it generalizes the underlying ideas to the scene flow setting, *i.e*. a setting where the motion in a 3D scene is unprojected and hence more meaningful – in contrast to optical flow. This offers new possibilities such as predictions by more realistic motion models (constant scene flow, constant $SE(3)$ motion) as well as a much better guided fusion (having access to optical flow and disparity). Moreover, unlike [25, 30] our approach is specifically tailored towards the baseline network, *i.e*. RAFT-3D, explicitly exploiting the underlying network characteristics (parametrization, cost volume, convex upsampling, high resolution disparity). As our experiments show, such a tight integration is required to further improve state-of-the-art two-frame scene flow networks. Finally, in contrast to [25] our approach predicts the forward from the backward flow instead of using the previous forward flow. This, in turn, avoids registering the information from the previous time frame (optical flow, fusion features) via motion-based interpolation (warping).

3. Approach

We propose a neural network for scene flow estimation from a triplet of stereo frames. Given the three stereo frames $(I_l^{t-1}, I_r^{t-1}), (I_l^t, I_r^t)$ and (I_l^{t+1}, I_r^{t+1}), our goal is to estimate the four-dimensional scene flow (u, v, d, d') [7] between frame t and frame $t+1$. Here, (u, v) denotes the optical flow and d and d' are the disparities at time t and $t+1$ registered to the reference frame I_l^t.

Following recent works [1, 13, 34, 42], we thereby decouple the disparity estimation from the recovery of the 3D motion. To this end, we precompute disparities for each stereo frame using a dedicated stereo method, yielding D^{t-1}, D^t and D^{t+1}. This allows us to directly take over the final estimate for d from D^t. Hence, the scene flow problem reduces to estimating (u, v, d'). Please note that d' cannot be taken over directly from D^{t+1} since d' is registered to I_l^t while D^{t+1} is registered to I_l^{t+1}. However, knowing D^t, we can easily convert between d' and the change in disparity Δd, with $d' = D^t + \Delta d$ when estimating the scene flow.

Using this notation, let us now explain our two-frame baseline and subsequently our full multi-frame network.

3.1. Improved two-frame baseline

We base our work on the recent two-frame approach RAFT-3D [34], which first uses an off-the-shelf stereo estimation network to compute left-right disparities and then estimates the scene flow while keeping the reference disparity D^t fixed. The approach employs a recurrent neural network which operates at 1/8th of the original resolution, where the final result is obtained by a learned convex upsampling [33]. Notably, RAFT-3D predicts the scene flow in terms of a field of $SE(3)$ transformation matrices, and afterwards translates them to the standard parametrization (u, v, d'). Building upon this work, we proceed in two steps. Making use of recent progress in the field of stereo estimation, we first exchange RAFT-3D's stereo estimation network, GANet [44] with the recently well-performing LEAStereo [3]. Subsequently, with the improved stereo results, we fully retrain RAFT-3D using their provided code. This way, we obtain an improved two-frame baseline serving as a building block for our multi-frame network.

3.2. Multi-frame fusion network

Conceptually, our multi-frame fusion network consists of two shared instances of our improved two-frame baseline and a fusion module predicting the final scene flow. More precisely, given two initial motion estimates for the forward and backward flow in low resolution, we derive low-resolution features which we adaptively upsample and subsequently combine with high-resolution features, eventually fusing forward and inverted backward flow estimates in a feature-guided high-resolution fusion module. In the following, we discuss all steps of our approach in detail; see Figure 1 for a complete overview.

① **Initial scene flow estimation.** Initially, our improved two-frame baseline predicts forward ($t \rightarrow t+1$) and backward ($t \rightarrow t-1$) scene flow at 1/8th of the full image resolution; as in the original RAFT-3D method [34]. It predicts flow estimates in terms of a field of $SE(3)$ transformation matrices and a weighting mask for convex upsampling.

② **Low-resolution features.** In order to later guide our fusion of flow estimates, we consider two features derived from the specific architecture of our baseline. First, rigid-motion embedding vectors (*embVec*) are essential to our baseline method, as they are used for a soft-grouping of pixels that belong to objects with the same rigid motion [34]. Since this segmentation information can be valuable for the fusion of forward and backward flow estimates, we utilize these features as an input to our fusion module. To this end, we extract the 16-channel prediction of the rigid-motion embedding vectors by the network for both the forward and the backward baselines. Second, the cost volume is at the core of recent motion estimation algorithms [31, 33, 34] since it assigns matching costs to potential flow estimates. In order to better guide our fusion module, we look up the correlation costs (*corrCost*) for the current flow estimates in forward and backward direction, which provides supporting information on the quality of the estimates. Note that we omit the multi-scale pyramid and the spatially extended lookup employed by [33, 34] and only extract a single cost value per pixel for the central location.

③ **Joint convex upsampling.** So far, the flow predictions as well as the extracted features are given on 1/8th of the original resolution. As the next step, we will hence exploit the convex upsampling mask predicted by the baseline networks in order to obtain flow predictions and features on the original high resolution. This proceeding offers three advantages: We can utilize disparity maps, which are given at the original resolution, we can perform backward-to-forward prediction at the original resolution and we can ultimately fuse flows at the original resolution.

④ **High-resolution features.** With the correlation cost at hand, we have matching information based on image features, but so far we do not make use of any disparity cues to guide the fusion. In order to create meaningful features, we first convert the upsampled forward and backward transformation fields to optical flow and disparity estimates which yields $(u_{\mathrm{fw}}, v_{\mathrm{fw}}, d'_{\mathrm{fw}})$ and $(u_{\mathrm{bw}}, v_{\mathrm{bw}}, d'_{\mathrm{bw}})$, respectively. Then, we warp the initial high-resolution disparity estimates D^{t+1} and D^{t-1} using these optical flows such that they are aligned with the reference frame and subtract the corresponding disparity estimates in order to compute disparity residuals (*dispRes*) for both directions as

$$\mathcal{W}(D^{t+1}, u_{\mathrm{fw}}, v_{\mathrm{fw}}) - d'_{\mathrm{fw}} \ , \tag{1}$$

$$\mathcal{W}(D^{t-1}, u_{\mathrm{bw}}, v_{\mathrm{bw}}) - d'_{\mathrm{bw}} \ , \tag{2}$$

where $\mathcal{W}(D, u, v)$ denotes backward-warping of D using the optical flow (u, v). If the correct scene flow is given, the residuals are 0 for non-occluded pixels [34].

⑤ **Backward-to-forward prediction.** In our initial flow estimation, we predicted backward flow pointing towards the previous frame. In order to obtain a meaningful prediction in forward direction, we utilize the $SE(3)$ motion parametrization of the upsampled scene flow and invert the backward transformations with a differentiable matrix inversion [35]. Note that this inversion in matrix space is capable of performing true inversion of rotational motion rather than simple linear inversion in the standard scene flow representation, which only flips the sign. Subsequently, we convert the matrix representation of the forward and the inverted backward flow to optical flow and disparity change, which is the parametrization we employ for the fusion.

⑥ **Fusion inputs.** As a final step, we concatenate the forward and backward flow and all features which we provide to the fusion module, yielding 43 channels that are visualized in Figure 2. Summarizing, we employ the disparity in the reference frame D^t and for forward and backward direction scene flow estimates $(u, v, \Delta d)$, rigid-motion embedding vectors, correlation costs and disparity residuals.

⑦ **Fusion module.** With a rich set of inputs at hand, we apply our fusion module to predict a final scene flow estimate. The fusion module is a CNN that uses a U-Net architecture [26], employing three depth levels with channel sizes 64, 128 and 256, where each level in the downsampling as well as in the upsampling branch consists of two 3×3 convolutional layers with stride 1 and zero-padding, to preserve image dimensions. The downsampling uses the same convolutional layers with a stride of 2 and upsampling employs transposed convolutions with kernel size 4, stride 2 and zero-padding of 1. Similar to the original U-Net, we use residual connections between the downsampling and the upsampling branch, however, instead of concatenation, we add the upsampled tensor to the skip connection. After each convolutional layer, a LeakyReLU activation [16] with slope 0.1 is applied. Finally, the three-channel output is predicted with one 3×3 convolution without activation. Please note that in contrast to [27] our generalized fusion is not restricted to a convex combination, *i.e.* a linear blending of predictions and forward flow. Hence, it implicitly allows to perform corrections when performing the fusion, in case predictions or forward flow are not accurate.

3.3. Supervision

Our network predicts the scene flow as triplet $(u, v, \Delta d)$. We compute the target disparity as $d' = d + \Delta d$ and supervise our training using a robustified sublinear L1 loss, reading

$$\mathcal{L}_{\text{fuse}} = \sum_{\mathbf{x}} \left(\alpha \cdot |d' - d'_{\text{gt}}| + |u - u_{\text{gt}}| + |v - v_{\text{gt}}| + \epsilon \right)^{\gamma} . \quad (3)$$

In all our experiments, we chose $\epsilon = 0.01$ and $\gamma = 0.4$. We additionally introduce a weighting parameter α to balance the loss components for disparity and optical flow and set it to 2. Finally, we also utilize the multi-iteration loss from RAFT-3D [34] $\mathcal{L}_{\text{R3D}}$ that computes the $L1$ norm of optical flow and disparity change with a per-iteration weight. We apply it directly to the output of the forward baseline and obtain a total loss of $\mathcal{L} = \mathcal{L}_{\text{fuse}} + \mu \cdot \mathcal{L}_{\text{R3D}}$, with $\mu = 0.1$.

4. Experiments

We implemented our model in PyTorch [22] and initialized the fusion module's weights with the normal distributed initialization by He *et al.* [6] for convolutions, and zero-initialization for biases. For the two-frame baseline, we used code provided by the authors [34].

Training details. For the two-frame baseline, we followed the original training of RAFT-3D [34] with 200K steps pretraining on FlyingThings3D [19] and 50K steps finetuning on the KITTI *train* split [20]; the latter using our improved disparity estimates [3]. For training our multi-frame method, we initialized our shared forward and backward model with the pretrained two-frame baseline and also finetuned for 50K steps on the KITTI *train* split – this time, however, dividing the 50K steps in two stages. First, for 10K steps, we kept the parameters of the shared baseline models fixed in order to pretrain the fusion module. Then, for the remaining 40K steps, we trained our entire model end-to-end. Thereby, we used the Adam optimizer [11] with the same linear-decay learning rate strategy [29] as RAFT-3D [34], employing maximum learning rates of $5 \cdot 10^{-4}$ and $1 \cdot 10^{-4}$ for the two finetuning stages. During all stages, we trained on a single NVIDIA A100 GPU with batch size 4. Moreover, we utilized spatial and photometric augmentations [34] with crop size 256×960.

4.1. Benchmark results

In our first experiment, we compare the accuracy of our multi-frame scene flow method to that of other recent scene flow approaches from the literature. To this end, we computed the scene flow for the KITTI *test* split both with our novel M-FUSE approach as well as with its underlying two-frame baseline and submitted the corresponding flow fields to the official benchmark [20]. To this end, we can not only show total improvements but also investigate the influence of the improved stereo method we employ. Table 1 shows the obtained results together with the results of the ten top-ranked published scene flow methods. Thereby, it lists the standard outlier rates *D1* and *D2* for the disparities at time t and $t + 1$, the optical flow error *Fl* and the scene flow error *SF*. These errors are evaluated on *all* pixels, as well as separately for static *background* (bg) objects only moving due to camera motion and for dynamic *foreground* (fg) objects that move independently; see [20] for details. Additionally, for each outlier rate, the table shows relative improvements of the baseline and our method with respect to RAFT-3D as well as the relative improvements of our multi-frame approach compared to the two-frame baseline.

As one can see, already our baseline (RAFT-3D, with LEAStereo) shows significantly improved results compared to the original RAFT-3D approach (with GANet). In this context, the total gain of 9.9% can be mainly attributed to strong improvements in the background region. Our full M-FUSE approach then improves these results even further, outperforming RAFT-3D by 16.3%. Thereby, it also shows strong gains in the foreground, which are due to the consideration of multi-frame information (see M-FUSE vs. Baseline). As a result, on the KITTI benchmark our method ranks second for all pixels, and first for foreground regions.

In Figure 3 we analyze the multi-frame improvements for four exemplary KITTI sequences. In accordance with

Table 1. Top ranking non-anonymous submissions to the KITTI benchmark.

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
DTF_SENSE [27]	2.08	3.13	2.25	4.82	9.02	5.52	7.31	9.48	7.67	8.21	14.08	9.18
PRSM [38]	3.02	10.52	4.27	5.13	15.11	6.79	5.33	13.40	6.68	6.61	20.79	8.97
Binary TTC [1]	1.48	3.46	1.81	3.84	9.39	4.76	5.84	8.67	6.31	7.45	13.74	8.50
Stereo expansion [42]	1.48	3.46	1.81	3.39	8.54	4.25	5.83	8.66	6.30	7.06	13.44	8.12
ISF [2]	4.12	6.17	4.46	4.88	11.34	5.95	5.40	10.29	6.22	6.58	15.63	8.08
ACOSF [12]	2.79	7.56	3.58	3.82	12.74	5.31	4.56	12.00	5.79	5.61	19.38	7.90
UberATG-DRISF [15]	2.16	4.49	2.55	2.90	9.73	4.04	3.59	10.40	4.73	4.39	15.94	6.31
RAFT-3D [34]	1.48	3.46	1.81	2.51	9.46	3.67	3.39	8.79	4.29	4.27	13.27	5.77
RigidMask+ISF [43]	1.53	3.65	1.89	2.09	8.92	3.23	2.63	7.85	3.50	3.25	13.08	4.89
CamLiFlow [13]	1.48	3.46	1.81	**1.92**	8.14	**2.95**	**2.31**	**7.04**	**3.10**	**2.87**	12.23	**4.43**
M-FUSE (ours)	**1.40**	**2.91**	**1.65**	2.14	**8.10**	3.13	2.66	7.47	3.46	3.43	**11.84**	4.83
Baseline	**1.40**	**2.91**	**1.65**	1.97	9.22	3.17	2.98	9.51	4.06	3.53	13.57	5.20
Baseline Improvements (Baseline vs. RAFT-3D)	5.4%	15.9%	8.8%	21.5%	2.5%	13.6%	12.1%	-8.2%	5.4%	17.3%	-2.3%	9.9%
Multi-frame Improvements (M-FUSE vs. Baseline)	-	-	-	-8.6%	12.1%	1.3%	10.7%	21.5%	14.8%	2.8%	12.7%	7.1%
Overall Improvements (M-FUSE vs. RAFT-3D)	5.4%	15.9%	8.8%	14.7%	14.4%	14.7%	21.5%	15.0%	19.3%	19.7%	10.8%	16.3%

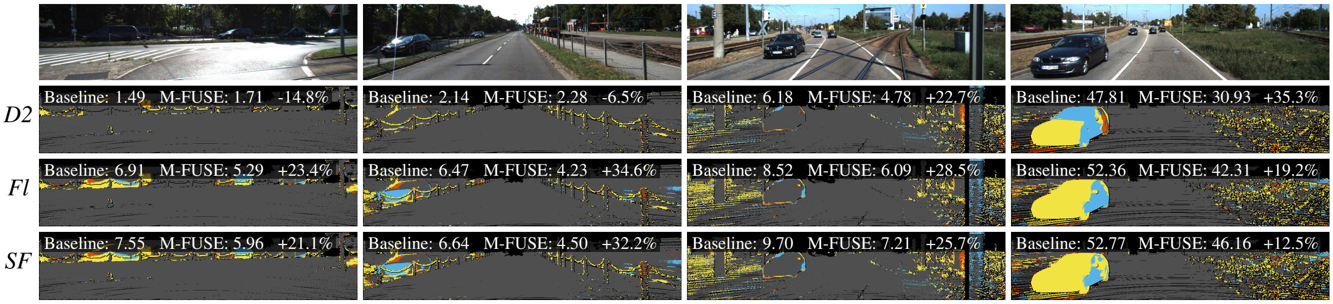
D2
Fl
SF
Baseline: 1.49 M-FUSE: 1.71 -14.8%
Baseline: 2.14 M-FUSE: 2.28 -6.5%
Baseline: 6.18 M-FUSE: 4.78 +22.7%
Baseline: 47.81 M-FUSE: 30.93 +35.3%
Baseline: 6.91 M-FUSE: 5.29 +23.4%
Baseline: 6.47 M-FUSE: 4.23 +34.6%
Baseline: 8.52 M-FUSE: 6.09 +28.5%
Baseline: 52.36 M-FUSE: 42.31 +19.2%
Baseline: 7.55 M-FUSE: 5.96 +21.1%
Baseline: 6.64 M-FUSE: 4.50 +32.2%
Baseline: 9.70 M-FUSE: 7.21 +25.7%
Baseline: 52.77 M-FUSE: 46.16 +12.5%

Figure 3. Qualitative evaluation of multi-frame improvements for four sequences of the KITTI benchmark (M-FUSE vs. baseline). *From top to bottom:* reference frame, change in the outlier errors *D2*, *Fl* and *SF*. *Grey:* Both methods are inliers, *blue:* M-FUSE is inlier and two-frame baseline is outlier, *red:* two-frame baseline is inlier and M-FUSE is outlier, *yellow:* both methods are outliers.

the numbers in Table 1, we observe that (i) multi-frame improvements are strongest for the optical flow error compared to the disparity error and (ii) the improvements are most prominent on the individually moving foreground objects. Figure 4 visually shows improvements for background regions (*top*) and foreground objects (*bottom*).

4.2. Ablations

We ablate our model architecture in Table 2. For these and all following experiments, we perform 4-fold cross validation on the KITTI *train* split for more reliable evaluations with only limited data available. Note that we omit the error measure for *D1* in the tables since it is identical.

Feature aggregation. Our U-Net computes additive increments for previous layers at the same resolution, which leads to a residual structure. We compare this approach to the strategy presented in [26], where feature maps are concatenated and not summed up before a convolution. Using additive residual connections slightly improves results.

Fusion module depth. In a second study, we ablate the depth of our fusion U-Net by comparing variants with two, three and four levels. While a two-level U-Net still gives on-par results in the *D2* error, our three-level U-Net outperforms both the other networks in the *Fl* and *SF* errors.

Additional fusion inputs. Our fusion network makes use of forward and backward scene flow estimates. In a larger set of experiments we determined which additional inputs to

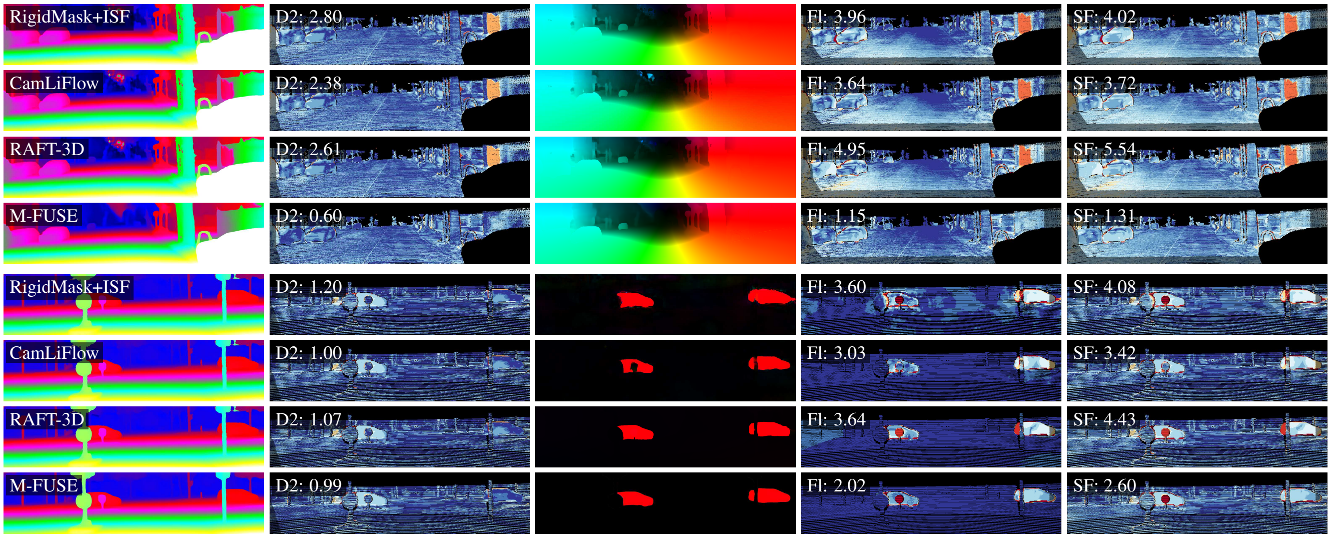

Figure 4. Qualitative comparison of our method, the original RAFT-3D, as well as the two top-performing approaches from the literature for two scenes using the visualizations provided by the KITTI benchmark [20]. *From left to right:* Target disparity visualization, corresponding *D2* error plot, optical flow visualization, corresponding *Fl* error plot, combined *SF* error plot.

Table 2. Ablation study. We show 4-fold cross validation results on KITTI *train* in terms of the D2, Fl and SF errors [20] as well as the number of parameters in millions.

	D2	Fl	SF	#param
two-frame	1.81	3.67	4.07	
Feature aggregation				
concat.	2.08	3.42	3.99	2.56
add (ours)	**1.99**	**3.21**	**3.82**	2.38
Fusion module depth				
2 levels	**1.99**	3.40	4.02	0.53
3 levels (ours)	**1.99**	**3.21**	**3.82**	2.38
4 levels	2.06	3.34	4.02	9.79
Additional fusion inputs				
none	2.72	3.33	4.62	2.36
corrCost,dispRes	**1.87**	3.36	3.83	2.36
corrCost,embVec	2.29	3.71	4.58	2.38
dispRes,embVec	1.99	3.43	3.97	2.38
corrCost,dispRes,embVec (ours)	1.99	**3.21**	**3.82**	2.38
corrCost,dispRes,embVec, I_l^t	2.10	3.27	3.96	2.38

our fusion module are useful. To this end, we compare our set of additional inputs (correlation costs, disparity residuals and rigid motion embedding vectors) to omitting all of them (none) and to omitting each of them individually, to assess their individual contribution. The results show that omitting all additional features significantly worsens results, which indicates that valuable information is contained in our set of features. Further, we see that omitting the rigid motion embedding vectors gives inconclusive results compared to our method, with superior *D2* results but a worse *Fl* error. The disparity residuals seem most essential: When removed, the resulting quality lowers significantly for all measures. Further, removing correlation costs has a slight negative impact. Additionally, we investigated if adding the reference frame I_l^t to the set of inputs [25] is helpful. However, this did not improve results any further, presumably because the correlation cost already provides sufficient information.

We show additional ablations with less conclusive results in the supplementary material.

4.3. Scene flow parametrization

Next, we investigate the influence of the underlying scene flow parametrization in our fusion module. Table 3 compares the image-space parametrizations of optical flow and target disparity (u, v, d') against optical flow and the change in disparity $(u, v, \Delta d)$. Additionally, we investigate the world-space 3D motion vector parametrization (x, y, z).

Evidently, the image-space parametrizations outperform the 3D vector parametrization by a large margin in the optical flow error *Fl*. We argue that this is due to the error measures that are employed in scene flow estimation, which also work in image space. Among the image-space parametrizations, using the change in disparity instead of the target disparity yields better results. Presumably, predicting the disparity change (motion) instead of the target disparity (structure) bears a greater resemblance to the optical flow, which renders this strategy superior for the joint prediction.

Table 3. Influence of the scene flow parametrization.

	D2	Fl	SF
(u, v, d')	2.06	3.49	4.13
$(u, v, \Delta d)$ (ours)	**1.99**	**3.21**	**3.82**
(x, y, z)	2.04	8.50	8.79

Table 4. Comparison of multi-frame strategies

	D2	Fl	SF
two-frame	**1.81**	3.67	4.07
warm-start (inv. backward)	2.59	5.29	5.73
warm-start (fw-warped prev.)	2.23	4.48	4.88
learned inv + mask fusion	2.06	3.96	4.39
specialized U-Net (bw-warped prev.)	2.10	3.78	4.26
specialized U-Net (inv. backward)	2.01	3.59	4.05
M-FUSE	1.99	**3.21**	**3.82**

Table 5. Parameter Count and Timing

	stereo		scene flow		total	
RAFT-3D	6.6M	4.0s	44.9M	0.4s	51.4M	4.4s
Baseline	1.8M	0.8s	44.9M	0.4s	46.7M	1.2s
M-FUSE	1.8M	1.2s	47.2M	1.3s	49.0M	2.5s

4.4. Comparison of multi-frame strategies

Finally, we compare our approach to three multi-frame strategies available in the literature: Warm-starting the method, a learned inversion with mask-based fusion and a specialized U-Net with additional inputs; see Table 4.

First, the warm-start initialization strategy has been shown to be highly successful in recent recurrent networks [33]. We considered two variants for our baseline approach: For one, we used the matrix-inverted backward flow as an initialization, in contrast to the identity matrix initialization from [34]. For the other, we initialized with the previous forward scene flow that is forward-warped in the corresponding Lie algebra [35] using the estimated optical flow. In Table 4, both approaches perform considerably worse than the two-frame baseline, even though forward-warping yields better results than inverted backward flow. This is in line with previous studies, where warm-start on the KITTI dataset did not yield improvements [33].

Second, we considered a recent strategy that relies on a learned backward-to-forward inverter [17, 27] followed by a predicted fusion mask that linearly combines forward and backward estimates [27]. We reimplemented the inversion and fusion module from [27] and pretrained the former, before using these modules in our method. For comparability, we adapted the modules to our three-channel prediction case, keeping D^t fixed. In Table 4, this strategy clearly yields worse results than our approach. We attribute this to the simplistic structure of the motion model and the restrictive convex combination of flow inputs.

Third, we investigated a strategy that employs the specialized fusion U-Net from FlowNet2 [9] for fusing optical flow estimates [25] guided by a brightness constancy map and the reference image. To this end, we extended this fusion module to the scene flow setting and embedded it in our approach. For a fair comparison, we also added disparity residuals to its fusion inputs. We evaluated two variants, one with backward-warping the previous flow estimate as in [25], and one with inverted backward flow, as in our approach. While only the approach with inversion is able to reach results on-par with the two-frame baseline, both cannot keep up with the results achieved by our method.

4.5. Timing and parameter counts

Table 5 shows that our method takes a total of around 2.5s per frame for inference on a NVIDIA GeForce RTX 2080 Ti with 51.4M parameters. The runtime is composed of 1.2s for stereo (3×0.4s LEAStereo) and 1.3s for scene flow (2×0.4s RAFT-3D baseline + 0.5s fusion). While runtime and parameter count is increased compared to the two-frame baseline, our method is still faster and more parameter-efficient than the original RAFT-3D approach due to the fast and lightweight LEAStereo method [3].

5. Conclusion

We proposed a novel multi-frame scene flow approach that leverages the performance of recent high accuracy two-frame methods. To this end, we developed an improved RAFT-3D baseline and embedded it into a U-Net-based fusion approach that adaptively integrates temporal information by combining an $SE(3)$-based extrapolation of the backward flow with the jointly estimated forward flow. The achieved results clearly demonstrate that our strategy of explicitly tailoring our architecture towards the underlying baseline pays off. With more than 16% improvements compared to the original RAFT-3D approach, they show significantly larger improvements than other multi-frame networks in the literature. Moreover, in absolute accuracy our method ranks second in the public KITTI benchmark, clearly outperforming all other multi-frame approaches.

Acknowledgements. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (B04). A.J. and J.S. acknowledge support from the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

References

[1] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Binary TTC: A temporal geofence for autonomous navigation. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12946–12955, 2021.

[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 *Proc. International Conference on Computer Vision (ICCV)*, pages 2574–2583, 2017.

[3] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. In *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, pages 22158–22169, 2020.

[4] Vladislav Golyanik, Kihwan Kim, Robert Maier, Matthias Nießner, Didier Stricker, and Jan Kautz. Multiframe scene flow with piecewise rigid motion. In *Proc. International Conference on 3D Vision (3DV)*, pages 147–160, 2017.

[5] Jens-Malte Gottfried, Janis Fehr, and Christoph. S. Garbe. Computing rangeflow from multi-modal Kinect data. In *Proc. International Symposium on Visual Computing (ISVC)*, pages 758–767. Springer, 2011.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[7] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–7, 2007.

[8] Junhwa Hur and Stefan Roth. Self-supervised multi-frame monocular scene flow estimation. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2684–2694. Springer, 2021.

[9] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[10] Joel Janai, Fatma Güney, Anurag Ranjan, Michael J. Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proc. European Conference on Computer Vision (ECCV)*, LNCS 11220, pages 713–731. Springer, 2018.

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

[12] Congcong Li, Haoyu Ma, and Qingmin Liao. Two-stage adaptive object scene flow using hybrid CNN-CRF model. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 3876–3883, 2021.

[13] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. Camliflow: Bidirectional camera-LiDAR fusion for joint optical flow and scene flow estimation. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5791–5801, 2022.

[14] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. SelFlow: self-supervised learning of optical flow. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4571–4580. Springer, 2019.

[15] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proc. Conference on Computer Vision and Pattern Recognition (CPVR)*, pages 3614–3622, 2019.

[16] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. Workshop on Deep Learning for Audio, Speech and Language Processing (ICML-W)*, 2013.

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

[18] Daniel Maurer, Nico Marniok, Bastian Goldluecke, and Andrés Bruhn. Structure-from-motion-aware patchmatch for adaptive optical flow estimation. In *Proc. European Conference on Computer Vision (ECCV)*, LNCS 11212, pages 575–592. Springer, 2018.

[19] 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 *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[20] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015.

[21] Michael Neoral and Jan Šochmann. Object scene flow with temporal consistency. In *Proc. Computer Vision Winter Workshop (CVWW)*, pages 1–9, 2017.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

[23] Yi-Ling Qiao, Lin Gao, Yu-Kun Lai, Fang-Lue Zhang, Mingzhe Yuan, and Shihong Xia. SF-Net: learning scene flow from RGB-D images with CNNs. In *Proc. British Machine Vision Conference (BMVC)*, 2018.

[24] Julian Quiroga, Thomas Brox, Frédéric Devernay, and James Crowley. Dense semi-rigid scene flow estimation from RGBD images. In *Proc. European Conference on Computer Vision (ECCV)*, LNCS 8695, pages 567–582. Springer, 2014.

[25] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yand, Erik B. Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *Proc. Winter Con-*

ference on Applications of Computer Vision (WACV), pages 2077–2086, 2019.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[27] René Schuster, Christian Unger, and Didier Stricker. A deep temporal fusion framework for scene flow using a learnable motion model and occlusions. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*, pages 247–255, 2021.

[28] 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 of Computer Vision (IJCV*, 128(2):527–546, 2020.

[29] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Proc. Defense + Commercial Sensing*, volume 11006, pages 369–386. SPIE, 2019.

[30] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised *raft* with full-image warping. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2021.

[31] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[32] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3939–2948, 2017.

[33] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, LNCS 12347, pages 402–419. Springer, 2020.

[34] Zachary Teed and Jia Deng. RAFT-3D: Scene flow using rigid-motion embeddings. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8375–8384. Springer, 2021.

[35] Zachary Teed and Jia Deng. Tangent space backpropagation for 3D transformation groups. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10338–10347, 2021.

[36] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics*, 6(31):187:1–187:11, 2012.

[37] Sundar Vedula, Simon Baker, Robert Collins, Takeo Kanade, and Peter Rander. Three-dimensional scene flow. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 722–729, 1999.

[38] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115(1):1–28, 2015.

[39] Pichao Wang, Wanqing Li, Zhimin Gao, Yuyao Zhang, Chang Tang, and Philip Ogunbona. Scene flow to action map: a new representation for RGB-D based action recognition with convolutional neural networks. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2017.

[40] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011.

[41] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6911–6920, 2017.

[42] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1334–1343, 2020.

[43] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1266–1275, 2021.

[44] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–194. Springer, 2019.