Improving Optical Flow on a Pyramid Level – Supplementary Material

Markus Hofinger[‡], Samuel Rota Bulò[†], Lorenzo Porzi[†], Arno Knapitsch[†], Thomas Pock[‡], Peter Kontschieder[†]

Facebook[†], Graz University of Technology[‡]
{markus.hofinger,pock}@icg.tugraz.at[‡]

This document contains supplementary material for the paper 'Improving Optical Flow on a Pyramid Level'. The structure of this supplementary document is the following:

- Further insights and experiments on gradient stopping:
  - Variance analysis
  - Smoothness analysis
  - Cross-task evaluation - stereo
  - Cross-architecture evaluation - PWC
- Details on Flow Cues:
  - Notation
  - Detailed explanation
  - Ablations on flow cues
- Further Analysis:
  - Ablation on Data distillation
  - Ablation on extending search range
  - Qualitative comparisons of validation epe changes
  - Histograms of errors
  - Qualitative training results for KITTI (images)
  - Qualitative training results for MPI-Sintel (images)
  - Qualitative Results on KITTI
  - Qualitative Results on MPI-Sintel
  - Sidenote on D2V and V2D operation with warping vs. sampling

A Further insights on Gradient Stopping

In this section we provide additional empirical insights to what is described in Section 3.3 in the main submission document, *i.e.*, why stopping the optical flow gradients between the levels is beneficial. We will do this by showing that the variance of the model parameter gradients over the training set is reduced (sec A.1), and the Lipschitzness of the parameter gradients is improved as well, while still providing a descent direction (sec A.2). For stochastic gradient descent methods these properties lead to improved convergence, which is what we already observed in the main paper (Fig. 5). Finally, we show that gradient stopping also leads to improved convergence for the different task of stereo estimation (sec A.3) as well as for a different architecture like PWC-Net (implemented in a different code-base (sec A.4).

A.1 Gradient stopping - variance analysis

It is known [9] that for stochastic gradient descent methods the rate of convergence decreases with increasing variance of the gradients over the training set. We can show empirically that stopping the optical flow gradients between levels (see Fig. 4 in the main paper) leads to a reduced variance of the gradients w.r.t the whole training dataset when compared to the baseline model. To ensure a fair and valid comparison, both model versions use identical parameters Θ and are fed with the exact same data batches ξ_n all the time. The variance over each epoch is computed independently for every single parameter using Welford's online variance computation algorithm [11] in a numerically stable variant. After each epoch, the mean of these P single parameter variances is computed for each model as

$$\sigma^2 = \frac{1}{P} \sum_{\theta \in \Theta} \underset{n \in N}{\mathrm{VAR}_{\mathrm{Welford}}} (\nabla f_\theta(\xi_n)) \tag{1}$$

and shown in Fig. 1. After gradient variances are computed a standard training is performed for 1 epoch, and the parameters of both models are update with the new parameters of the baseline model to ensure that the only difference in the gradient variance comes from the gradient computation itself. As can be seen in Fig. 1 our proposed partial gradient stopping truly reduces the gradient variance w.r.t the model parameters and the training dataset. This leads to the improved rate of convergences for our proposed partial gradient stopping over the baseline.

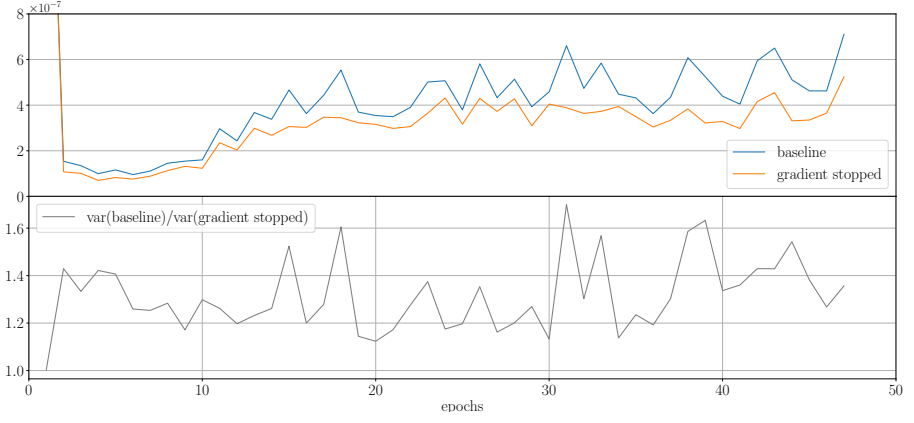


Fig. 1: Gradient variance for a HD^3 baseline model vs. a model with the proposed gradient stopping. The baseline has a higher gradient variance over the training data, which leads to slower convergence.

A.2 Gradient stopping - smoothness analysis

Here we will show that stopping the partial optical-flow gradient between the levels also leads to a better Lipschitzness of the gradients of the loss also known as β-smoothness, while still providing a descent direction. It is well known that the rate of convergence increases if the function has a low curvature which corresponds to a low β-smoothness. We follow the approach of [7] that estimate 'effective' β-smoothness (β_{eff}) by measuring the l_2 gradient change over difference in parameters, as they move along the gradient direction in the optimization.

$$\beta_{\text{eff}} = \frac{\|\nabla f(\xi, \Theta_1) - \nabla f(\xi, \Theta_2)\|_2}{\|\Theta_1 - \Theta_2\|_2} \tag{2}$$

We ensure a fair comparison between the baseline model and our version with partial flow gradient stopping by evaluating the gradient functions with the exact same parameters Θ_i and data batches ξ_n for both model versions at all times. Fig. 2 (top) shows that on average β_{eff} is lower which corresponds to

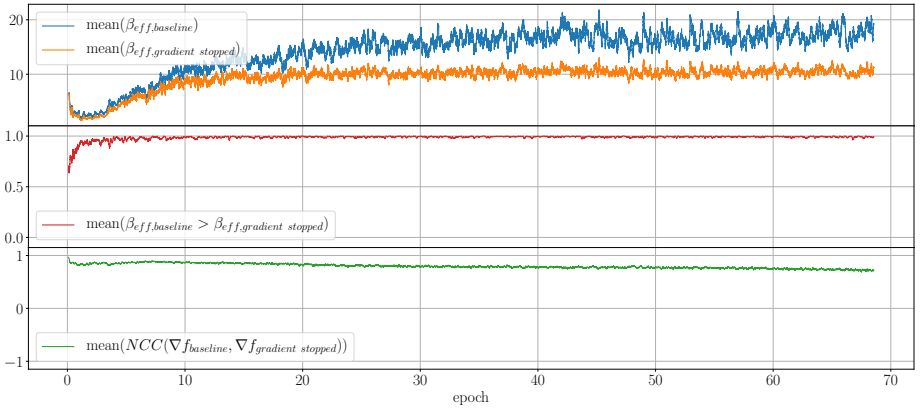


Fig. 2: Partial gradient stopping vs. Lipschitzness of gradients. Top: Average of the effective β-smoothness shows that model with gradient stopping is smoother (lower β_{eff}) than the baseline; Middle: Percentage of how often gradient stopping leads to smoother results; Bottom: Positive normalized cross correlation between the model parameter gradients indicates that it is still a descent direction.

a lower curvature. This is confirmed by the center plot that directly compares β_{eff} for both models in every iteration before averaging the result. The lower plot shows that the normalized cross correlation (NCC) of the gradients for the parameters of both models are positively correlated. This is in contrast to the NCC of the partial optical flow gradients (Fig. 5, main paper) between the levels. Therefore, stopping the partial optical flow gradients between the levels, reduces intermediate parts that oppose each other, which in turn leads to better final gradients at the model parameters. The latter are still positively correlated with the original parameter gradients, which shows that they still

provide a descent direction, but with better convergence properties, as shown by our various insights. Finally, based on these analyses and the improved results obtained in our experimental section we conclude the importance of blocking partial optical flow gradients across levels in a pyramidal setting for improved convergence.

A.3 Gradient stopping on depth estimation – HD^3, depth from Stereo.

With this experiment we show that stopping the partial optical flow gradients between the levels also works for stereo estimation. We use the Stereo training setup of HD^3 in their original publicly available codebase[1] and run a training on the Flying Things Stereo dataset. We choose to use the original version of the code base just with gradient stopping added, and keep the original training procedure that trains only on the *left* disparity. We do this to show that the effect of gradient stopping is not just limited to the simultaneous forward and backward training used in the main paper, but is a more general one.

Again, we find significant improvements with our proposed partial flow gradient stopping, as can be seen in Fig. 3, which leads to an improvement of ≈10% on the final EPE. This confirms that gradient stopping also works for stereo estimation networks. Furthermore, it verifies that gradient stopping does not require joint forward- and backward flow training as used in the flow ablations in the main paper, but also leads to significant gains for a standard forward-only training.

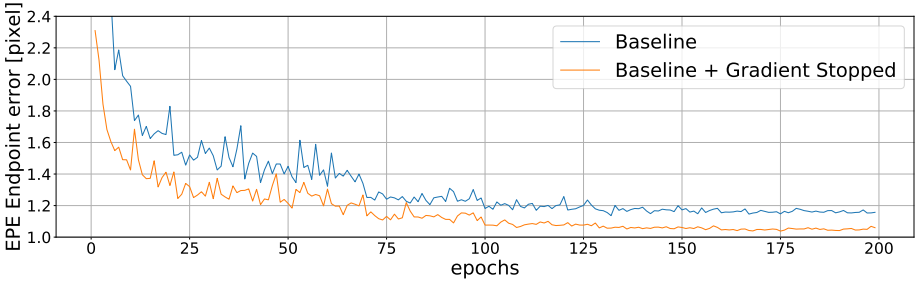


Fig. 3: Improving HD^3 Stereo estimation with gradient stopping. Curves show validation Endpoint error (EPE) after each training epoch. Simple gradient stopping leads to faster convergence of the EPE

A.4 Gradient stopping on different architectures estimation – Improving PWC-Net Optical Flow.

With this experiment we show that this behaviour is not limited to HD^3 but also applies to other networks as PWC. We use the PWC-Net implementation from

[1] HD^3 codebase : https://github.com/ucbdrive/hd3/

the official IRR-PWC [2] publicly available code base[2], and run a training on the Flying Chairs dataset using their provided data processing and augmentation strategy, and follow all default settings for training. We run two experiments, the baseline and an experiment where we apply gradient stopping at the upsampling layer within the pyramid structure used therein. In direct comparison we found both, significantly improved reduction of the training loss for the final high-resolution level as well as the validation EPE (Fl-all is not reported from their inference code).

Fig. 4 shows the validation EPE of an exemplary experimental result on the PWC flow Network. As can be seen, applying gradient stopping leads to a faster convergence of the EPE. This immediately leads to initial gains of more than 10% at 20 epochs and 6% at 100 epochs. Therefore, lower EPE values can be reached faster. We kept the original learning rate schedule for comparability, but even in this setting that was optimized for the original baseline, a difference of approximately 2% remains after 200epoch. Gradient stopping shows a clear positive impact, even though the used PWC-variant directly regresses the flow at each level, whereas the HD[3] baseline that was used for many comparisons in the main paper uses residual estimates together with the D2V and V2D operations. This shows that stopping the gradients for the flow at the upsampling layer leads to a faster decrease of the EPE also across multiple types of optical flow networks.

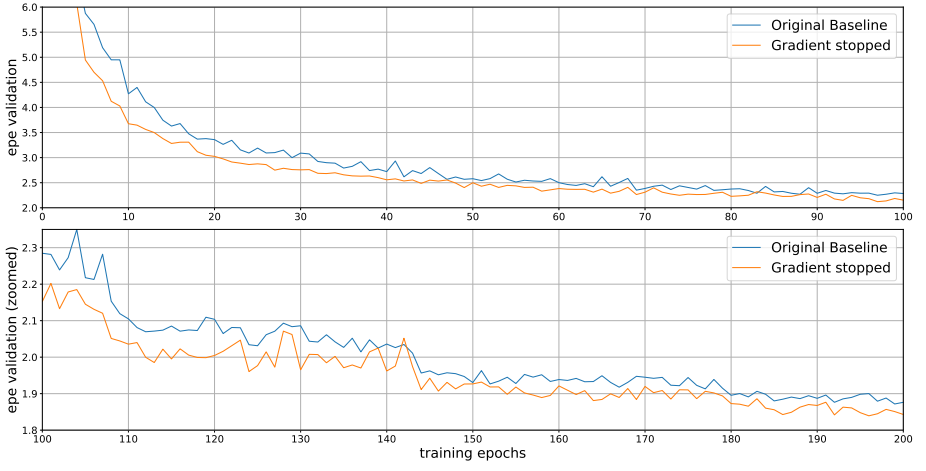


Fig. 4: Improving PWC-Net with gradient stopping. Training with gradient stopping vs. original. Gradient stopping leads to faster decrease for the validation EPE.

[2] IRR-PWC codebase: https://github.com/visinf/irr

B Further details on Flow Cues

B.1 Notation

To simplify equations in the following section, we define a few additional terms on top of the main paper. Given a pixel $x \in \mathcal{I}_1^l$ we denote by $x_{1\to2} \in \mathbb{R}^2$ the matching position of x in I_2^l (in absolute terms), *i.e.* $x_{1\to2} = x + F_{1\to2}^l(x)$. Similarly, for the opposite direction, we define $y_{2\to1} \in \mathbb{R}^2$ for pixels $y \in \mathcal{I}_2$.

Details on the Flow Cues module The use of prior knowledge when computing optical flow has been widely explored in classical methods. Recently, [2] successfully used forward-backward flow warping as feature for occlusion upsampling. Although this feature is hand crafted it is very valuable, as it provides cues that would otherwise be hard to learn for a convolutional network since it can connect completely different locations on the coordinate system of I_1 and I_2. Classic approaches like inverse flow estimation [4] show that there are even more cues that can potentially be of interest. We therefore propose to combine multiple of these cues, which can be mutually beneficial, and make them explicitly available to the network as cheaply computable features to directly improve flow predictions.

In order to do so, our architecture keeps jointly track of the forward and backward flows by exploiting Siamese modules with shared parameters, with features from I_1 and I_2 being fed to the two branches in mirrored order. A downside is its increased memory consumption, which we noticeably mitigate by adopting In-Place Activated BatchNorm [6] throughout our networks. Without additional connections, the Siamese modules compute the forward $F_{1\to2}$ and backward $F_{2\to1}$ flow mappings in a completely independent way. However, in practice the true flows are strongly tied to each other, although they reside on different coordinate systems. We therefore provide the network with a Flow Cue Module that gives each branch different kind of cues about its own and the other branch's flow estimates. Each of these cues represents a different mechanism to bring mutually supplementary information from one coordinate system to the other. For the sake of simplicity, we will always present the results of the cues in the coordinate system of the branch that operates on the features of I_1.

Forward-backward flow warping. Since both flow mappings are available, they can be used to bring one flow in the coordinate frame of the other via dense warping. For example, a forward flow estimate $F_{1\to2}^{\text{fb}}$ can be made from the backward flow $F_{2\to1}$ by warping it with the forward flow $F_{1\to2}$:

$$F_{1\to2}^{\text{fb}}(x) = -F_{2\to1}(x_{1\to2}) \tag{3}$$

The other direction $F_{2\to1}^{\text{fb}}$ can be computed in a similar way. Comparing the estimated results $F_{1\to2}^{\text{fb}}$ versus $F_{1\to2}$ can be used for consistency checks, and is used in unsupervised flow methods [5, 13] to estimate occlusions in a heuristic manner.

Reverse flow estimation [4]. In contrast to the previous cue, reverse flow estimation can be used to estimate the forward flow $F_{1\to2}$ directly from backward flow $F_{2\to1}$ alone, although in a non-dense manner. The reverse flow estimates are denoted by $F^{\text{rev}}_{2\to1}$ and $F^{\text{rev}}_{1\to2}$ and are obtained by

$$F^{\text{rev}}_{1\to2}(x) = -\frac{\sum_{y\in\mathcal{I}_2}\omega(x, y_{2\to1})F_{2\to1}(y)}{\omega_1(x)}\,, \tag{4}$$

where $\omega(x,x') = [1-|x_u-x'_u|]_+\,[1-|x_v-x'_v|]_+$ denotes the bilinear interpolation weight of x' relative to x, and

$$\omega_1(x) = \sum_{y\in\mathcal{I}_2}\omega(x, y_{2\to1}) \tag{5}$$

is a normalizing factor. In the dis-occluded areas where the denominator of Eq. (4) is 0, we define the flow values $F^{\text{rev}}_{1\to2}(x) = 0$. In occluded areas $F^{\text{rev}}_{1\to2}$ will become an average of the incoming flows. Similarly, we define $F^{\text{rev}}_{2\to1}$ by swapping 1 and 2 as well as x and y in Eq. (4).

Map uniqueness density [8, 10]. Provides information about occlusions and dis-occlusions and basically corresponds to ω_1 in Eq. (4) for image I_1. The value of $\omega_1(x)$ provides the (soft) amount of pixels in I_2 with flow vectors pointing towards $x \in \mathcal{I}_1$. Occluded areas will result in values ≥ 1 whereas areas becoming dis-occluded in values ≤ 1. ω_1 is therefore an indicator on where the reverse flow is more or less precise. Similarly, we have ω_2(x) for I_2.

Out-of-image occlusions. This represents an indicator function, *e.g.* $o_1 : \mathcal{I}_1 \to \{0,1\}$ for image I_1, providing information about flow vectors pointing out of the other image's domain, *i.e.*

$$o_1(x) = \mathbb{1}_{x_{1\to2}\notin\mathcal{I}_2} \tag{6}$$

and similarly we define $o_2 : \mathcal{I}_2 \to \{0,1\}$ for image I_2.

The Flow Cue Module. We show in Fig. 5 how the flow cues mutually benefit from one another in different areas. *E.g.*, the-out-of-image occlusions o_1 allow to differentiate which dis-occlusions in map uniqueness density ω_1 are real dis-occlusions, *i.e.* areas where the object moved away, and where the low density stems from flow vectors in the second image that are just likely not visible in the current crop.

We therefore provide the network with all the additional flow cues mentioned above, by stacking them as additional features together with the original forward flow $F_{1\to2}$ for the subsequent part of the network. Therefore, the network now has three differently generated flow estimates including its own prediction $F_{1\to2}$. The following layers can therefore reason about consistency and probable sources of outliers with a far better basis than one single cue alone could provide. Symmetrically, the same is done for the backward stream (see Fig. 5).

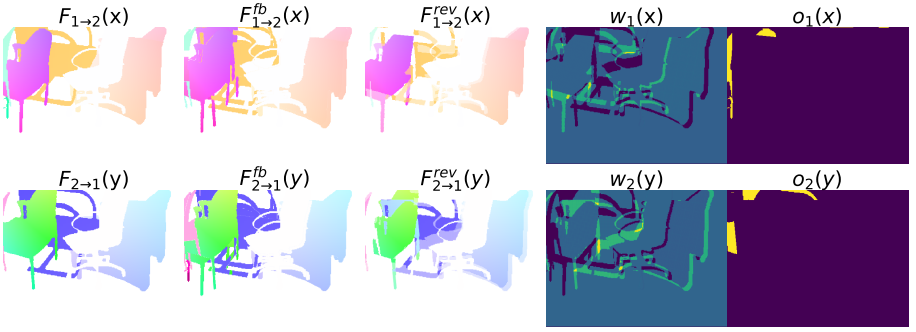


Fig. 5: Flow Cues module output illustration for a given optical flow input; Left to right: Input flow, forward-backward estimate, reverse flow estimate, map uniqueness density, out of image occlusions. Note the differences in the F^{fb} and F^{rev} and how

B.2 Ablations on Contributions of Flow Cues.

Here we evaluate the impact of our proposed flow cues in comparison to related ones from prior works [2, 3], demonstrating their effect on relevant error measures on the Flying Chairs2 dataset. The ablations are performed training on Flying Chairs 2. We use averages over the last 10 validation results to reduce the effect of single spikes. In Tab. 1 we list our findings, always on top of activating gradient stopping and SAMPLING due to its preferable behavior for estimating flow of fine-grained structures.

Providing *Mapping Occurrence Density* (MOD) [8, 10] as the only Flow Cue and hence information about the occlusions and dis-occlusions slightly degrades results in terms of both, EPE and Fl-all. When running the Sampling in combination with Forward-Backward flow warping (FWDBWDFW) we encounter a considerable reduction of errors – particularly on the Fl-all errors. Finally, when combining SAMPLING with all our proposed Flow Cues (ALL CUES), *i.e.* reverse flow estimation, mapping occurrence density, and out-of-image occlusions, we obtain the lowest errors.

Table 1: Ablation results on Flow Cues on top of Cost Volume Sampling and Gradient Stopping using CV-range of ±4 pixels

MOD	FWDBWDFW	ALL CUES	EPE	Fl-all
✗	✗	✗	1.208	6.192
✓	✗	✗	1.217	6.271
✗	✓	✗	1.202	6.171
✓	✓	✓	1.186	6.156

C Further Analysis

C.1 Details on Distillation

In this section we provide additional details on our distillation strategy. In contrast to [1] we don't want to transfer knowledge from a larger network or ensembles to smaller ones, but to transfer it from one domain to the other. We therefore avoid to keep all predictions, since some are completely off, and instead try to filter out the most trustworthy. Specifically, we apply the following filters, obtaining "pseudo ground-truth" annotations (Fig. 6 main paper, bottom right):

- We use forward $F_{1\to2}$ and backward $F_{2\to1}$ flows to estimate occlusions. Specifically, we regard a pixel $y \in I_1$ as not occluded if the following holds [5]

$$\|F_{1\to2}(y) + F_{2\to1}(y_{1\to2})\|^2 - 0.05 < 0.01 \left(\|F_{1\to2}(y)\|^2 + \|F_{2\to1}(y_{1\to2})\|^2\right) \quad (7)$$

- We compute the photometric error using SAD on a per pixel basis and determine a mask of good predictions by thresholding the error.
- We determine the confidence of the network using the method proposed in [12] and retain predictions with a confidence above 95%.
- We filter pixels that are more than 3 pixels away from the gt
- Finally, we combine all of the previous filters and apply an additional pruning using an erosion operation to remove small patches, in order to only keep regions with sufficient trustable data.

Since this is still a "pseudo ground-truth" we do not apply LMP on the distillation part $\mathcal{L}_D$ part of the loss but only on the supervised Loss $\mathcal{L}_S$

Ablations on distillation. Here we show ablation results for our distillation approach. We compare the results in Tab. 2 after standard pretraining on Flying Chairs and Flying Things 3D to a finetuning on KITTI with and without distillation. To gain insights on overfitting and generalization, we provide results on the training datasets as well as cross validation scores on different datasets. For completeness we also provide finetuning results of our retrained initial baseline, which uses IPABN with leakyRelu but none of the other improvements. The baseline uses the same training schedule, but the original 2k finetuning iterations on Kitti instead of early stopping, as it converges slower since it doesn't use gradient stopping.

It can be seen that standard finetuning leads to high gains on the training datasets, especially Kitti 2015 but also drastically reduces performance on the other non-finetuning datasets (which is not surprising). Compared to the baseline, the improved model already mitigates this reduction in generalization to some extent, while performing better on the target dataset. Using our proposed distillation approach further improves this generalization to unseen datasets. Interestingly, it even leads to a small improvement on the training dataset itself. Since we drastically filter the "pseudo ground-truth" it could mean that, this additional information acts like additional augmentation that benefits the finetuning.

Table 2: Ablation on Distillation for KITTI finetuning. Comparing pretraining vs. finetuning (FT) vs. finetuining using distillation (Dist). Non baseline models use CV-range ±8 and all proposed improvements (Highlighting **best** and second-best results).

			Kitti 2012			Kitti 2015			Flying Chairs2		Flying Things		Sintel final	
			EPE [1]	Fl-all [%]	Fl-all [%]	EPE [1]	Fl-all [%]	Fl-all [%]	EPE [1]	Fl-all [%]	EPE [1]	Fl-all [%]	EPE [1]	Fl-all [%]
baseline	Dist	FT	train		test	train		test	Flying Chairs2		Flying Things		Sintel	
✗	✗	✗	2.37	8.65 %	-	7.09	18.93 %	-	**2.31**	**8.6%**	**5.77**	**11.5%**	**4.68**	**11.4%**
✓	✗	✓	(0.85)	(2.35 %)	-	(1.38)	(4.41 %)	-	11.23	24.0%	61.34	47.7%	26.77	23.7%
✗	✗	✓	(0.77)	(1.91 %)	-	(1.18)	(3.58 %)	-	9.77	36.8%	27.92	46.2%	9.36	21.5%
✗	✓	✓	**(0.76)**	**(1.84 %)**	**2.25**	**(1.14)**	**(3.28 %)**	**6.35**	6.29	24.7%	27.57	38.0%	9.39	18.9%

C.2 Extending search range

Here we investigate the impact of extended search ranges in various configurations of our model. The base configuration always uses gradient stopping and SAD for the cost volume construction and is trained on Flying Chairs2 in forward and backward direction. Results for Flying Things 3D are presented to give an insight on generalization on the closest related dataset. We compare Cost Volume Sampling vs. Cost Volume Warping, both in combination with LMP.

What can be seen from the data, is that LMP clearly helps to improve EPE and Fl-all metrics in all cases. What can also be seen, is that in general extending the search range leads to better performance. However, for the combination of Sampling and LMP there is a gain of 0.06 in EPE when going from a range of ±4 to ±8, while for the same settings without LMP the total improvement is just 0.01 and with warping it is 0.04. We do not experience significant gains that warrant a search range extension of more than ±8 (see Tab. 3). We therefore recommend a version with ±8, Cost Volume Sampling and LMP as it leads to satisfactory performance.

Table 3: Extending the cost-volume range leads to lower errors, especially when combined with LMP and Sampling. Model was trained on Flying Chairs2.

Warp/ Sample	Range +/-	LMP	Flying Chairs2 EPE [1]	Flying Chairs2 Fl-all [%]	Flying Things EPE [1]	Flying Things Fl-all [%]
W	4		1.20	6.18%	14.84	25.25%
W	8		1.16	5.96%	14.20	24.18%
S	4		1.18	6.15%	15.14	25.00%
S	6		1.16	6.02%	13.92	24.13%
S	8		1.17	5.97%	13.46	23.52%
S	10		1.15	5.92%	15.06	23.44%
W	4	✓	1.17	6.00%	14.12	23.47%
W	8	✓	1.13	5.81%	13.49	23.07%
S	4	✓	1.17	5.97%	14.46	23.00%
S	6	✓	1.14	5.86%	13.41	23.05%
S	8	✓	**1.11**	**5.76%**	12.97	22.41%
S	10	✓	**1.11**	5.78%	**12.78**	**22.10%**

C.3 Histogram of Errors

Fig. 6 and Fig. 7 show the gains made over the KITTI training sequences as achieved with our submitted model that used all proposed improvements and CV-range of ±4. The gains are made visible in form of histograms, where the ground truth flow magnitude is used for the binning. As can be seen, our improvements are not limited to a single range of flow magnitudes but affect the whole spectrum of flow vectors. At this point we want to remind the reader, that adding our contributions hardly changes the number of learnable parameters (e.g. $\approx +1\%$ for HD^3) in the network. The gains therefore result from using the provided parameters more effectively.

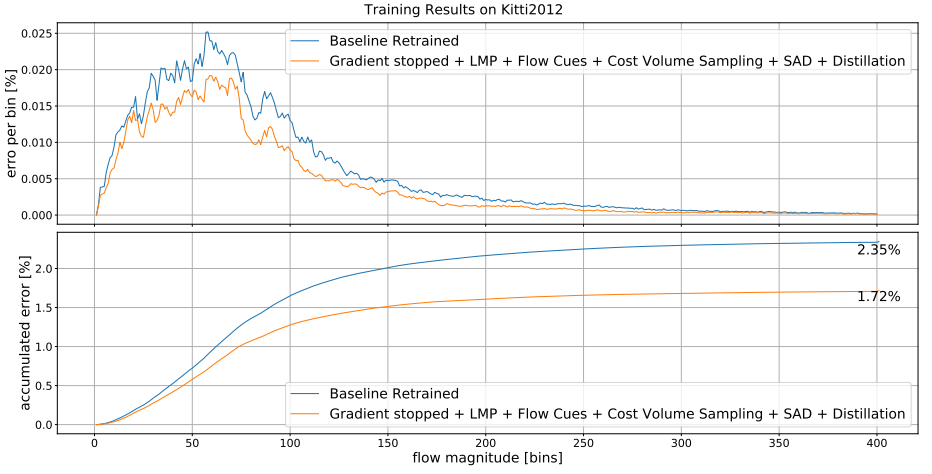


Fig. 6: Histogram of errors on the training data of KITTI 2012. The errors are grouped in bins according to the ground truth flow magnitude on which they occurred. Adding all our contributions consistently improves in all areas.

C.4 Qualitative Comparison of Training Convergence

Fig. 8 shows exemplary validation curves of an HD^3 type model during the Flying Things 3D pre-training. This is the last part of the pre-training stage before finetuning on KITTI or Sintel. These comparisons are qualitatively only, as they were conducted on center crops of the forward flow only, to keep extra computation effort during training low. We evaluate on the same validation split provided by the original HD^3 codebase.

The validation curves in Fig. 8 illustrate the overall behavior that we observed on the different datasets and models, when adding our different contributions. When adding gradient stopping to the baseline there is a significant drop in both EPE and Fl-all. Adding *Loss Max Pooling* (LMP) on top mostly affects the Fl-all by focusing on the remaining difficult examples. Adding our remaining

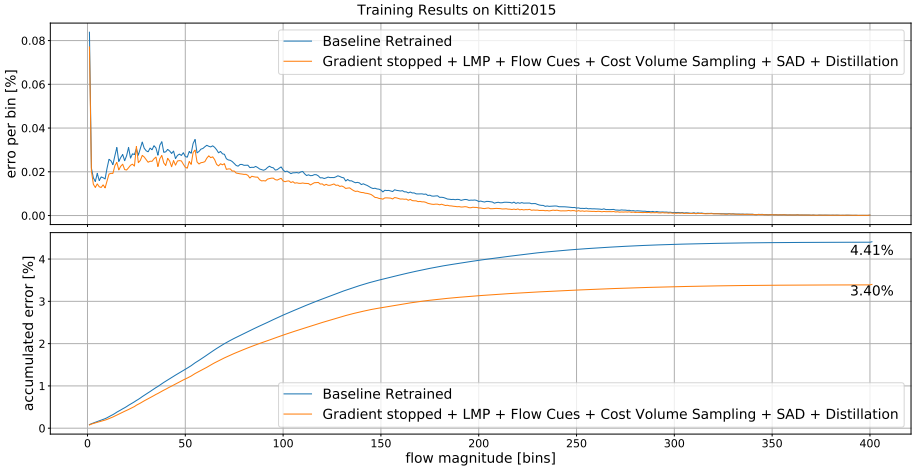


Fig. 7: Histogram of errors on the training data of KITTI 2015. The errors are grouped in bins according to the ground truth flow magnitude on which they occurred. Adding all our contributions consistently improves in all areas.

contributions (Data Distillation is only applied on KITTI) leads to an additional boost in performance on both EPE and Fl-all.

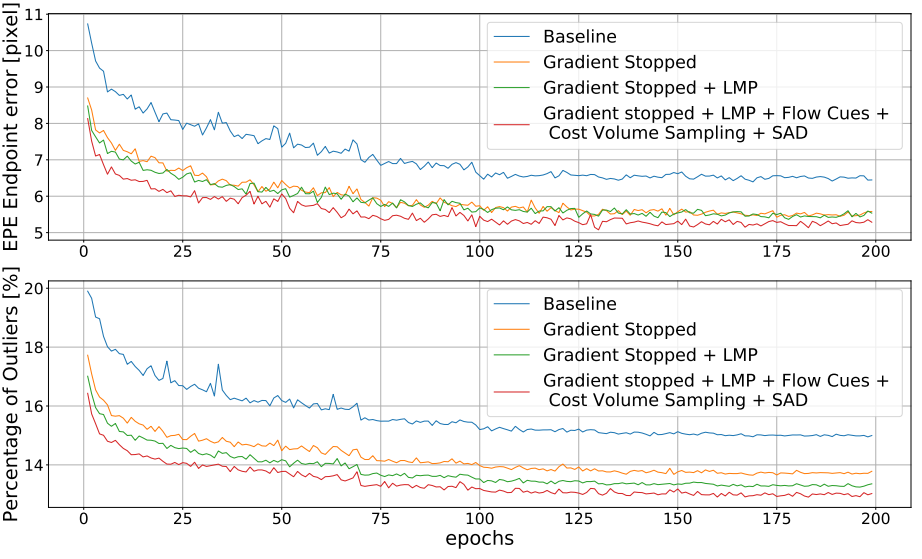


Fig. 8: Qualitative comparison of training curves on Things 3D pre-training for optical flow with and HD^3 type model (CVr±4). Large drop from Baseline to Gradient Stopped version on EPE and percentage of outliers (Fl-all). LMP improves mainly on Fl-all; adding all our remaining contributions gives additional boost on EPE and Fl-all.

C.5 Qualitative Comparisons of Training Results on KITTI

In this section various qualitative results on the KITTI training images will be shown. Fig. 9 shows comparisons between the baseline model as taken from the HD^3 modelzoo and our best model that uses all our contributions. What can be seen especially well in the error plots, is that our model improves a lot on the moving cars. Furthermore, it improves on fine details, which can e.g. be seen e.g. at the guard rails, where it manages to keep sharper edges and a more homogeneous background. At the same time, it does not suffer from the artifacts present in the top region of the baseline model. The figures are best viewed in high-resolution on a PC.

C.6 Results on MPI-Sintel

We outperform the state-of-the-art on the challenging MPI-Sintel Dataset. Fig. 11 shows the *Results and Rankings* for MPI-Sintel test results at the time of submission to the server. For more details please refer to the main paper.

Fig. 10 shows the comparison of a HD^3 baseline model and our improved baseline trained on the MPI-Sintel training sequence. As can be seen our improved model allows to preserve more fine details like the stick in the bamboo scene or the pike. Also, it seems to be better at detecting and correcting hardly connected moving backgrounds that seem to cause problems for the modelzoo baseline.

C.7 Sidenote: Sampling vs. Warping – HD^3's D2V and V2D Operations.

One of the key innovations in the HD^3 [12], was the introduction of the D2V and V2D operations that allow to transform match densities into vectors and vice versa. This operation is used for absolute and residual flows and implicitly assumes an equidistant fixed grid spacing for the flow. However, this assumption is actually not always valid since the warping operation can deform the space over which the search window operates in the warped image $I_{2\to1}$. I.e. a movement of a single pixel in the search window in $I_{2\to1}(x)$ can move the correspondence to a completely different position in $I_2(y)$ dependent on the flow $F_{2\to1}(x)$ that was used for the warping.

In the case of sampling, the equidistance of the grid is preserved, since it always uses a single flow vector $F_{2\to1}(x)$ as offset for the entire search window for each individual pixel. Therefore, the spacing of the search window stays equidistant w.r.t. $I_2(y)$ and hence for the D2V and V2D operations.

References

1. Hinton, G.E., Vinyals, S., Dean, J.: Distilling the knowledge in a neural network. In: Deep Learning Workshop, NIPS (2014)

Fig. 9: Comparisons on the KITTI 2015 training set. Theirs = HD^3 baseline, HD^3 with our modifications and contributions and a CV-range of ± 4.

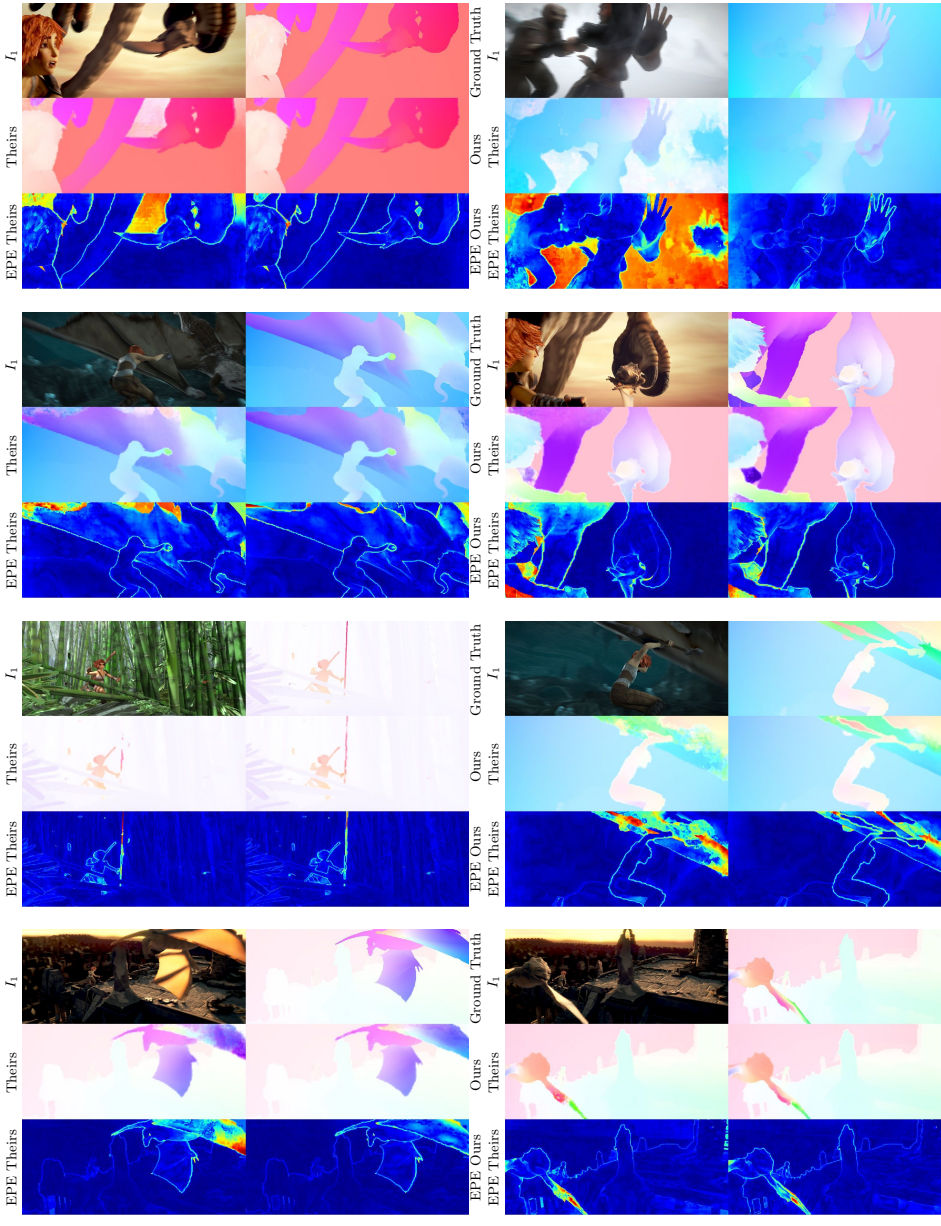
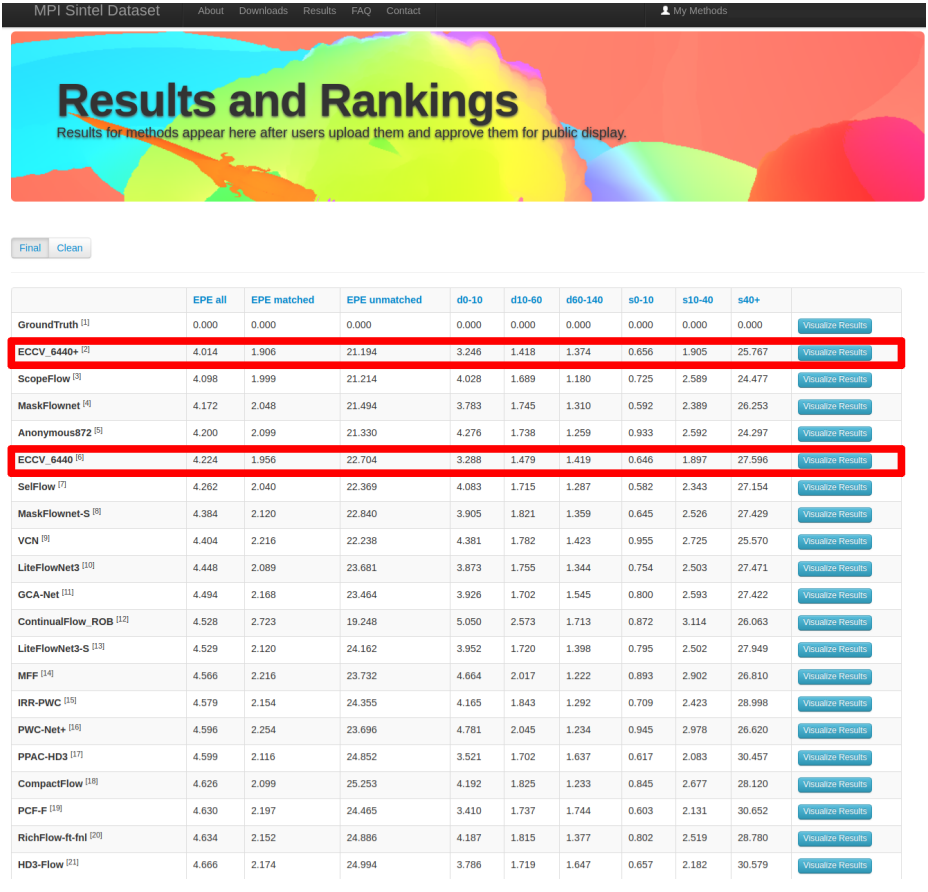


Fig. 10: Comparisons on the MPI-Sintel training set. Theirs = HD^3 baseline model from the modelzoo. Ours = Adding our contributions on top (Except for Data Distillation since Sintel has dense GT) and a CV-range of ± 4.



	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
ECCV_6440+ [2]	4.014	1.906	21.194	3.246	1.418	1.374	0.656	1.905	25.767	Visualize Results
ScopeFlow [3]	4.098	1.999	21.214	4.028	1.689	1.180	0.725	2.589	24.477	Visualize Results
MaskFlownet [4]	4.172	2.048	21.494	3.783	1.745	1.310	0.592	2.389	26.253	Visualize Results
Anonymous872 [5]	4.200	2.099	21.330	4.276	1.738	1.259	0.933	2.592	24.297	Visualize Results
ECCV_6440 [6]	4.224	1.956	22.704	3.288	1.479	1.419	0.646	1.897	27.596	Visualize Results
SelFlow [7]	4.262	2.040	22.369	4.083	1.715	1.287	0.582	2.343	27.154	Visualize Results
MaskFlownet-S [8]	4.384	2.120	22.840	3.905	1.821	1.359	0.645	2.526	27.429	Visualize Results
VCN [9]	4.404	2.216	22.238	4.381	1.782	1.423	0.955	2.725	25.570	Visualize Results
LiteFlowNet3 [10]	4.448	2.089	23.681	3.873	1.755	1.344	0.754	2.503	27.471	Visualize Results
GCA-Net [11]	4.494	2.168	23.464	3.926	1.702	1.545	0.800	2.593	27.422	Visualize Results
ContinualFlow_ROB [12]	4.528	2.723	19.248	5.050	2.573	1.713	0.872	3.114	26.063	Visualize Results
LiteFlowNet3-S [13]	4.529	2.120	24.162	3.952	1.720	1.398	0.795	2.502	27.949	Visualize Results
MFF [14]	4.566	2.216	23.732	4.664	2.017	1.222	0.893	2.902	26.810	Visualize Results
IRR-PWC [15]	4.579	2.154	24.355	4.165	1.843	1.292	0.709	2.423	28.998	Visualize Results
PWC-Net+ [16]	4.596	2.254	23.696	4.781	2.045	1.234	0.945	2.978	26.620	Visualize Results
PPAC-HD3 [17]	4.599	2.116	24.852	3.521	1.702	1.637	0.617	2.083	30.457	Visualize Results
CompactFlow [18]	4.626	2.099	25.253	4.192	1.825	1.233	0.845	2.677	28.120	Visualize Results
PCF-F [19]	4.630	2.197	24.465	3.410	1.737	1.744	0.603	2.131	30.652	Visualize Results
RichFlow-ft-fnl [20]	4.634	2.152	24.886	4.187	1.815	1.377	0.802	2.519	28.780	Visualize Results
HD3-Flow [21]	4.666	2.174	24.994	3.786	1.719	1.647	0.657	2.182	30.579	Visualize Results

Fig. 11: MPI-Sintel *Results and Rankings* - our method improves upon the state of the art. Screenshot taken on March 12, 2020. Short names have been updated after publication to also show IOFPL on benchmark server.

2. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: CVPR (2019)
3. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In: ECCV (2018)
4. Javier, S., AgustÃn, S., Nelson, M.: Direct estimation of the backward flow. Tech. rep., Institute for Systems and Technologies of Information, Control and Communication, Las Palmas de Gran Canaria (2013)
5. Liu, P., Lyu, M.R., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: CVPR (2019)
6. Rota Bulò, S., Porzi, L., Kontschieder, P.: In-place activated batchnorm for memory-optimized training of DNNs. In: (CVPR) (2018)
7. Santurkar, S., Tsipras, D., Ilyas, A., Mądry, A.: How does batch normalization help optimization? In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 2488–2498. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
8. Unger, M., Werlberger, M., Pock, T., Bischof, H.: Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In: (CVPR) (2012)
9. Wang, C., Chen, X., Smola, A.J., Xing, E.P.: Variance reduction for stochastic gradient optimization. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 181–189. Curran Associates, Inc. (2013), http://papers.nips.cc/paper/5034-variance-reduction-for-stochastic-gradient-optimization.pdf
10. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: CVPR. pp. 4884–4893 (06 2018). https://doi.org/10.1109/CVPR.2018.00513
11. Welford, B.P.: Note on a method for calculating corrected sums of squares and products. Technometrics **4**(3), 419–420 (1962). https://doi.org/10.1080/00401706.1962.10490022
12. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: CVPR (2019)
13. Yu, J.J., Harley, A.W., Derpanis, K.G.: Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: Computer Vision - ECCV 2016 Workshops, Part 3 (2016)