ScopeFlow: Dynamic Scene Scoping for Optical Flow

Aviram Bar-Haim[1] and Lior Wolf[1,2]
[1]Tel Aviv University
[2]Facebook AI Research

Abstract

We propose to modify the common training protocols of optical flow, leading to sizable accuracy improvements without adding to the computational complexity of the training process. The improvement is based on observing the bias in sampling challenging data that exists in the current training protocol, and improving the sampling process. In addition, we find that both regularization and augmentation should decrease during the training protocol.

Using an existing low parameters architecture, the method is ranked first on the MPI Sintel benchmark among all other methods, improving the best two frames method accuracy by more than 10%. The method also surpasses all similar architecture variants by more than 12% and 19.7% on the KITTI benchmarks, achieving the lowest Average End-Point Error on KITTI2012 among two-frame methods, without using extra datasets.

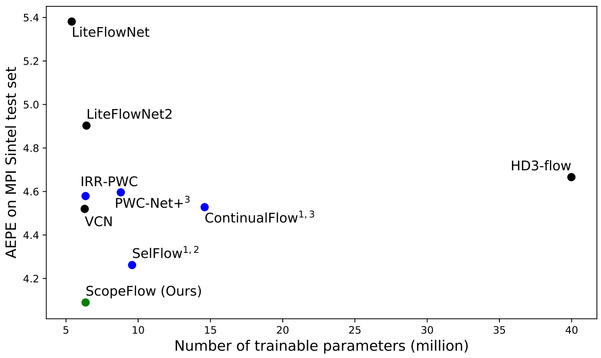

Figure 1. **Model size and accuracy trade-off.** Average-end-point-error of the leading methods on the MPI Sintel benchmark vs. the number of trainable parameters. PWC-Net based models are marked in blue. Our model is in the bottom left corner, achieving the best performance with low number of parameters during training. [1]Methods that use more than two frames. [2]SelFlow uses half of the parameters in test time. [3]Trained with additional datasets.

1. Introduction

The field of optical flow estimation has benefited from the availability of acceptable benchmarks. In the last few years, with the adoption of new CNN [15] architectures, a greater emphasis has been placed on the training protocol.

A conventional training protocol now consists of two stages: (i) pretraining on larger and simpler data and (ii) finetuning on more complex datasets. In both stages, a training step includes the following: (i) sampling batch frames and flow maps, (ii) applying photometric augmentations to the frames, (iii) applying affine (global and relative) transformations to the frames and flow maps, (iv) cropping a fixed size random crop from both input and flow maps, (v) feeding the cropped frames into a CNN architecture, and (vi) backpropagating the loss of the flow estimation.

While photometric augmentations include variations of the input image values, affine transformations are used to augment the variety of input flow fields. Due to the limited motion patterns represented by today's optical flow datasets, these regularization techniques are required for the data driven training. We chose the word *scoping*, to define the process of affine transformation followed by cropping, as this process sets the scope of the input frames.

To improve optical flow training, we ask the following questions: Q1. How do fixed size crops affect this training? Q2. What defines a good set of scopes for optical flow? Q3. Should regularization be relaxed after pretraining?

Our experiments employ the smallest PWC-Net [27] variant of Hur & Roth [11], with only 6.3M trainable parameters, in order to support low memory, real time processing. We demonstrate that by answering these questions and contributing to the training procedure, it is possible to train a dual frame, monocular and small sized model to outperform all other models on the MPI Sintel benchmark. The trained model improves the accuracy of the baseline model, which uses the same architecture, by 12%. See Fig. 1 for a comparison to other networks.

Moreover, despite using the smallest PWC-Net variant, our model outperformed all other PWC-Net variants on both KITTI 2012 and KITTI 2015 benchmarks, improving the baseline model results by 12.2% and 19.7% on the public test set, and demonstrating once more the power of using the improved training protocol.

Lastly, albeit no public benchmark is available for occlusion estimation, we compared our occlusion results to other published results on the MPI Sintel dataset, demonstrating more than 5% improvement of the best published F1 score.

Our main contributions are: (i) showing, for the first time, as far as we can ascertain, that CNN training for optical flow and occlusion estimation can benefit from cropping randomly sized scene scopes, (ii) exposing the powerful effect of regularization and data augmentation on CNN training for optical flow and (iii) presenting an updated generally applicable training scheme and testing it across benchmarks, on the widely used PWC-Net network architecture.

Our code is attached as supplementary and our models will be openly shared, in order to encourage follow-up work, to support reproducibility, and to provide an improved performance to off the shelf real-time models.

2. Related work

The deep learning revolution in optical flow started with deep descriptors [29, 6, 2] and densification methods [34]. Dosovitskiy *et al.* [4] presented FlowNet, the first deep end-to-end network for optical flow dense matching, later improved by Ilg *et al.* [12], incorporating classic approaches, like residual image warping. Ranjan & Black [24] showed that deep model size can be much smaller with a coarse to fine pyramidal structure. Hui *et al.* [9, 10] suggested a lighter version for FlowNet, adding features matching, pyramidal processing and features driven local convolution. Xu *et al.* [31] adapted semi-global matching [8] to directly process a reshaped 4D cost volume of features learned by CNN, inspired by common practices in stereo matching. Yang & Ramanan [32] suggested a method for directly learning to process the 4D cost volume, with a separable 4D convolution. Sun *et al.* [27] proposed PWC-Net, which includes pyramidal processing of warped features, and a direct processing of a partial layer-wise cost volume, demonstrated strong performance on optical flow benchmarks. Many extensions were suggested to the PWC-Net architecture, among them multi-frame processing, occlusion estimation, iterative warping and weight sharing [25, 23, 17, 11].

Pretraining optical flow models Today's leading optical flow learning protocols, include pretraining on large scale data. The common practice is to pretrain on the FlyingChairs [4] and then on FlyingThings3D [20] (FChairs and FThings). As shown by recent works [19, 12], the multistage pretraining ordering is critical. The FChairs dataset includes 21,818 pairs of frames, generated by CAD models [1], with flicker images background. FThings is a natural extension of the FChairs dataset, having 22,872 larger 3D scenes with more complex motion patterns. Hur & Roth [11] created a version of FChairs with ground truth occlusions, called FlyingChairsOcc (denoted FChairsOcc), to allow supervised pretraining on occlusion labels.

Datasets and benchmarks The establishment of larger complex benchmarks, such as MPI Sintel [3] and KITTI [7, 22], boosted the evolution of optical flow models. The MPI Sintel dataset was created from the Sintel movie, composed of 25, relatively long, annotated scenes, with 1064 training frames in total. The final pass category of Sintel is a challenging one, having many realistic effects to mimic natural scenes. The KITTI2012 dataset comprises 194 training pairs with annotated flow maps, while KITTI2015 has 200 dynamic color training pairs. Furthermore, some methods are using more datasets during the finetune process, such as HD1K [14], Driving and Monkaa [20].

Motion categories MPI Sintel provides a stratified view of the error magnitude of challenging motion patterns. The ratio of the best mean error for the small motion category (slower than 10 pixels per frame) to the large motion category (faster than 40 pixels per frame) is approximately x44. In Sec. 3, we present one possible theoretical explanation for the poor performance of state of the art methods in large motions, and suggest an approach to improve the accuracy of this pixels category.

Another example is the category of unmatched pixels. This category includes pixels belonging to regions that are visible only in one of two adjacent frames (occluded pixels). As expected, these pixels share much higher end-point-error than match-able pixels: the ratio of the best match-able EPE to the best non match-able is approximately 9.5.

Different deep learning approaches were suggested to tackle the problems of fast objects and occlusion estimation. Among the different solutions suggested were: occlusion based loss [28] and model [16, 17] separation, and multi-frame support for long-range, potentially occluded, spatio-temporal matches [25, 23]. We suggest a new approach for applying multiple strategies online. Our findings imply that the training can be improved by applying scene scope variations, while taking into account the probability of sampling valid examples from different flow categories.

Training procedure and data augmentation Fleet & Weiss [5] showed the importance of photometric variations, boundary detection and scale invariance to the success of optical flow methods. In recent years, the importance of the training choices attracted more attention [17]. Sun *et al.* [26] used training updates to improve the accuracy of the initial PWC-Net model by more than 10%, showing they could improve the reported accuracy of FlowNetC (a sub network of FlowNet) by more than 50%, surpassing FlowNet2 [10] performance, with their updated training protocol. Mayer *et al.* [19] suggests that no single best general-purpose training protocol exists for optical flow, and different datasets require different care. These conclusions are in line with our findings on the importance of proper training.

2.1. PWC-Net architectures

PWC-Net [27] is the most popular architecture for optical flow estimation to date, and many variants for this architecture were suggested [25, 21, 11, 17, 23]. PWC-Net architecture was built over traditional design patterns for estimating optical flow, given two temporally consecutive frames, such as: pyramidal coarse-to-fine estimation, cost volume processing, iterative layerwise feature warping and others.

Features warping In PWC-Net, a CNN encoder creates feature maps for the different network layers (scales). The features of the second image are backward warped, using the upsampled flow of the previous layer processing, for every layer l, except the last layer l_{Top}, by:

$$c_w^l(x) = c_2^l(x + up_{\times 2}(f^{l+1}(x))) \tag{1}$$

where x is the pixel location, $c_w^l(x)$ is the backward warped feature map of the second image, $f^{l+1}(x)$ is the output flow of the coarser layer, and $up_{\times 2}$ is the $\times 2$ up-sampling module, followed by a bi-linear interpolation.

Cost volume decoding A correlation operation applied on the first and backward warped second image features, in order to construct a cost volume:

$$cost^l(x_1, x_2) = \frac{1}{N}(c_1^l(x_1))^T c_w^l(x_2) \tag{2}$$

where $c_n^l(x) \in \mathbb{R}^N$ is a feature vector of image n.

The cost volume is then processed by a CNN decoder, in order to estimate the optical flow directly. In some variants of PWC-Net [23, 11] there is an extra decoder with similar architecture for occlusion estimation.

Iterative residual processing Our experiments employ the Iterative Residual Refinement proposed by Hur & Roth [11]. The reasons we chose to test our changes for the PWC-Net architecture on the IRR variant are: (i) IRR has the lowest number of trainable parameters among all PWC-Net variants, making a state of the art result obtained with proper training more significant, (ii) it uses shared weights that could be challenged with scope and scale changes, and if successful, it would demonstrate the power of a rich, scope invariant feature representations, (iii) this variant is using only two frames - demonstrating the power of dynamic scoping without long temporal connections, and (iv) the occlusion map allows the direct evaluation of our training procedure on occlusion estimation. Therefore, any success with this variant directly translates to real-time relatively low complexity optical flow estimation.

3. Scene scoping analysis

Due to the limited number of labeled examples available for optical flow supervised learning, most of the leading methods, in both supervised and unsupervised learning, are using cropping of a **fixed sized** randomly located patches. We are interested in understanding the chances of a pixel to be sampled, within a randomly located fixed size crop, as a function of its location in the image.

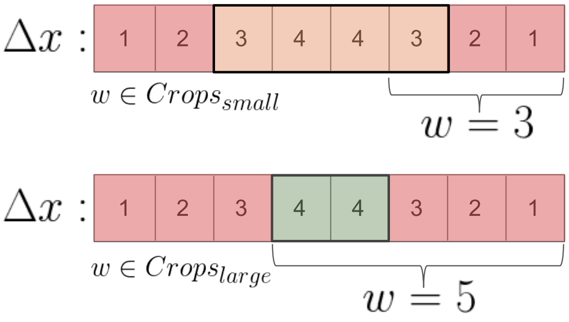

Figure 2. **Illustration of Lemma 1**. The probability for a pixel to be sampled within a valid random crop location, depends on the image width W, the crop width w and the distance to the closest border Δx. For both samples $W = 8$. Top: $w = 3$ ($w \in Crops_{small}$). Bottom: $w = 5$ ($w \in Crops_{large}$). Each pixel is labeled with Δx.

1D image random cropping statistics Consider a 1D image with a width W, a crop size w and a pixel location x. Let Δx denote the distance of the pixel from the closest border, and Δw denote the difference between the image width W and the crop size w. Let $Crops_{large}$ be the set of crop sizes with w larger than half of the width, $\frac{W}{2} < w \leq W$. Let $Crops_{small}$ be the complement set of crop sizes smaller or equal to half of the width, $0 < w \leq \frac{W}{2}$. Two instances of this setup are depicted in Fig. 2.

Using the notations above, pixels are separated into three different categories, described in the following lemma.

Lemma 1. *For an image size W and a randomly chosen crop of size $0 < w \leq W$ the probability of a pixel, with coordinate x and distance to the closest border Δx to be sampled, is as follows:*

$$P(x|W, w) = \begin{cases} 1 & \textit{if } \Delta w < \Delta x \\ \frac{w}{\Delta w + 1} & \textit{if } w \leq \Delta x \\ \frac{\Delta x}{\Delta w + 1} & \textit{otherwise} \end{cases} \tag{3}$$

where $\Delta w + 1$ is the number of valid crops.

Proof. For illustration purposes, the three cases are color coded, respectively, as green, orange, and red, in Fig.2. We handle each case separately. (i) **Green**: Every valid placement must leave out up to Δw pixels from the left or the right. Since Δx is larger than Δw, the pixel x must be covered. (ii) **Orange**: In this case, there are w possible locations for pixel x within the patch, all of which are achievable, since Δx is large enough. Therefore, w patch locations out of all possible patches contain pixel x. (iii) **Red**: In this case, the patch can start at any displacement from the edge that is nearer to x, that is not larger than Δx, and still cover pixel x. Therefore, there are exactly Δx locations out of the $\Delta w + 1$ possible locations. □

2D image random cropping statistics Since the common practice in current state-of-the-art optical flow training protocols is to crop a fixed sized crop, in the range $\frac{W}{2} < w \leq W$ ($w \in Crops_{large}$), we will focus in the reminder of this section on the green and red categories, which are the relevant categories for crop sizes with each dimension [h,w] larger than half of the corresponding image dimension (*i.e.* in $Crops_{large}$), and represent a cropping of more than a quarter of the image.

From the symmetry of our 1D random cropping setup, in both x and y axes, we can use Eq. 3 in order to calculate the probability of sampling pixels in a 2D image of size $[H, W]$, with a randomly located crop of a fixed size $[h, w]$. The probability of sampling a central (green) pixel remains 1, while the probability of sampling a marginal (red) pixel (x, y) in 2D, is given by:

$$P_{red}(x, y|H, h, W, w) = \frac{\min(\Delta x, \Delta w)\min(\Delta y, \Delta h)}{(\Delta w + 1)(\Delta h + 1)} \tag{4}$$

Where $\Delta h = H - h$, $\Delta w = W - w$ the difference between the image and the crop width and height, and $\Delta x, \Delta y$ represent the distance from the closest border, as before. Eq. 4 represents the ratio between the number of crop locations where a (marginal) pixel with $\Delta x, \Delta y$ is sampled to the number of all unique valid crop locations. An illustration of this sampling probability is demonstrated in Fig. 3 for varying ratios of crop size axes and image axes.

Fixed crop size sampling bias As in the 1D cropping setup, given an image of size $[H, W]$ and a crop size $[h, w]$, we can define a central area (equivalent to the green pixels in 1D), which will always be sampled. Respectively, we can define a marginal area (equivalent to the red pixels in 1D), where Eq. 4 holds.

Analyzing Eq. 4 we can infer the following: (i) in the marginal area, for a fixed crop size $[h, w]$, the probability of being sampled decreases quadratically along the image diagonal, when Δx and Δy both decrease together, and (ii) in the marginal area, for a fixed pixel, the probability of being sampled decreases quadratically when the crop size decreases (when Δw and Δh both decrease together).

Therefore, when using a fixed sized crop to augment a dataset with a random localization approach, there will be a dramatic sampling bias towards pixels in the center of the image, preserved by the symmetric range of random affine parameters. For example, with the current common cropping approach for the MPI Sintel data-set, the probability of the upper left corner pixel to be sampled in a crop equals $\frac{1}{(1024-768+1)(436-384+1)} = 0.000073\%$, while the pixels in the central $[332, 512]$ crop will be sampled in any randomized crop location.

This sampling bias could have a sizable influence on the training procedure. Fig. 3 illustrates the distribution of fast pixels in both MPI Sintel and KITTI datasets. Noticeably, pixels of fast moving objects (with speed larger than 40 pixels per frame) are often located at the marginal area, while slower pixels are more centered in the scene. This should not be a surprise, since (i) lower pixels belong to nearer scene objects and thus have a larger scale and velocity, and (ii) fast moving objects usually cross the image borders.

Moreover, many occluded pixels are also located close to the image borders. Therefore, increasing a crop size could also help to observe a more representative set of occlusions during training. Therefore, we hypothesized that larger crops can also improve the ability to infer occluded pixels motion from the context.

3.1. Scene scoping approaches

Fig. 3 shows the crop size effect on the probability to sample different motion categories. Clearly, the category of fast pixels suffers the most from reduction of the crop sizes. We tested four different strategies for cropping the scene dynamically (per mini batch) during training: (S1) fixed partial crop size (the common approach), (S2) cropping the largest valid crop size, (S3) randomizing crop size ratios from a pre-defined set with:

$$R_{fixed} = \{(0.73, 0.69), (0.84, 0.86), (1, 1)\} \tag{5a}$$

$$(r_h, r_w) = randchoice(R_{fixed}), \tag{5b}$$

where (r_h, r_w) are one of the three crop ratios, and strategy (S4) is a range-based crop size randomization:

$$s = randint(round(r_{min} \cdot S), round(r_{max} \cdot S)), \tag{6}$$

where s is the crop axis size (h or w), S is the full image axis size (H or W), and $[r_{min}, r_{max}]$ is the range of crop size ratios $\frac{s}{S}$ for sampling.

We also employ different affine transformations, and dynamically change the zooming range along the training, to enlarge the set of possible scene scopes, and improve the robustness of features to different scales. In Sec. 5.2 we describe the experiments done in order to find an appropriate approach for feeding the network with a diversity of scene scopes and reducing the inherent sampling bias explained in this section, caused by the current cropping mechanisms.

In addition to testing the scope modifications based on our analysis, we were also interested in testing different parameters of the training.

4. Training, regularization and augmentation

Learning rate and training schedules The common LR schedules, proposed by Ilg *et al.* [12], used to train deep optical flow networks, are S_{long} or S_{short} for the pretraining stage, and S_{ft} for the finetune phases. We used the shorter schedule, suggested by [11], of using S_{short} for pretraining, half of S_{short} for FThings finetuning, and S_{ft} for Sintel and KITTI datasets. We also examine the effect of retraining and over-training specific stages of the multi-phase training.

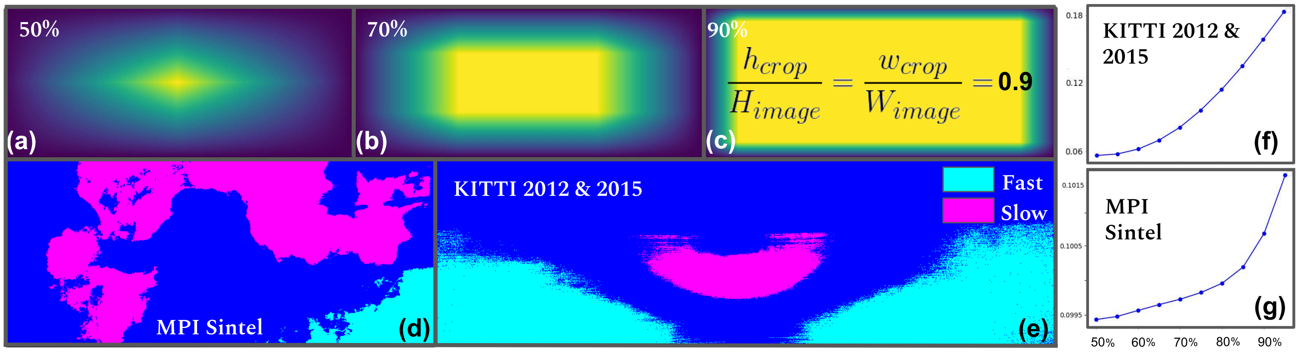

Figure 3. **Sampling bias caused by fixed size random crops.** (a),(b),(c): Pixel sampling probability maps for a fixed sized crop, with ratios of 50%, 70% and 90% respectively, for each axis. The probability to sample a marginal pixel shrinks drastically with the crop size. (d),(e): areas with strong prevalence for motion categories. High velocities tend to start from lower corners, while small ones tend to occur in the middle and upper part of the scene. (f),(g): graphs of the changing ratio of sampling probabilities between fast (> 40) and slow (< 10) pixels, for different crop and image axes ratios. clearly, fast pixels benefit more when increasing the crop size than slow pixels.

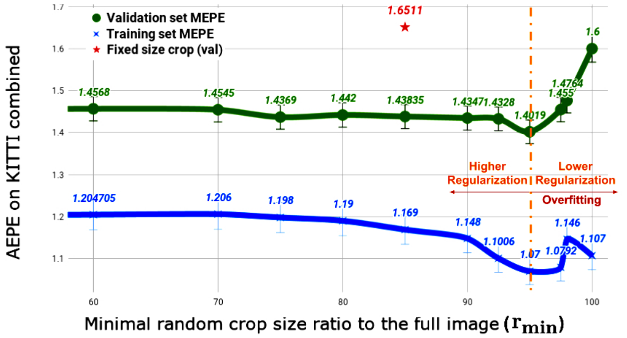

Figure 4. Accuracy of models trained with different ranges of random crop sizes, on the combined KITTI dataset. The maximal crop size is the full image. Validation AEPEs improve when increasing the minimal crop size ratio (r_{min} in Eq. 6) up to 95% of the full image axes. AEPE for a fixed sized crop based training: $*$.

Data augmentation The common practice in the current training protocol employs two types of data augmentation techniques: photometric and geometric. The details of these augmentations did not change much since FlowNet [4]. The photometric transformations include input image perturbation, such as color and gamma corrections. The geometric augmentations include a global or relative affine transformation, followed by random horizontal and vertical flipping. Due to the spatial symmetric nature of the translation, rotation and flipping parameters, we decided to focus on the effect of zooming changes, followed by our new cropping approaches.

Regularization The common protocol also includes weight decay and adding random Gaussian noise to the augmented image. In our experiments, we tested the effect of eliminating these sources of regularization at different stages of the multi-phase training.

5. Experiments

In this section, we describe the experiments and results for our research questions. Specifically, we tested (i) how can we change the current training pipeline in order to improve the final accuracy, and (ii) the effect of feeding the network with different scopes of the input during training, using different cropping and zooming changes.

All our experiments on KITTI used both KITTI2012 and KITTI2015, and for Sintel both the clean and final pass, for training and validation sets. We denote the combined datasets of Sintel and KITTI as Sintel combined and KITTI combined. We also tested the approach, suggested by Sun *et al*. [27], to first train on a combined Sintel dataset, followed by another finetune on the final pass.

All of our experiments employ the common End Point Error metric for flow evaluation, and F1 for occlusion evaluation. KITTI experiments also present outlier percentage.

5.1. Finetuning a pretrained model

Since the cost of pretraining is approximately ×7 than of the final finetune, we first present experiments done on the finetuning phase, in which we employ models pretrained on FChairs and FThings, published by the authors of IRR-PWC. These experiments are conducted on the Sintel dataset, since it has similar statistics of displacements to the FChairs dataset [19] used for pretraining. We tested different training protocol changes, and found that substantial gains could be achieved using the following changes: **1. Cropping strategies.** During the initial finetune, we tested the cropping approaches specified in Sec. 3.1 on Sintel. The results specified in Tab. 1 show that the range-based crop size randomization approach (Eq. 6) was comparable to taking the maximal valid crop (although much more efficient computationally), and both improved Sintel validation error of models trained with smaller fixed crop sizes.

RD+ RN	Zoom changes	Max crop	Random crop	Sintel train	Sintel val	KITTI train	KITTI val	KITTI val Out%
x	x	x	x	2.660	3.312	1.728	1.651	0.057
✓				2.623	3.224	1.654	1.644	0.056
✓	✓			2.453	3.108	1.580	1.649	0.056
✓		✓		2.428	3.053	1.182	1.594	0.059
✓			✓	2.537	3.081	**1.070**	**1.402**	**0.051**
✓	✓	✓		**2.320**	2.987	1.225	1.607	0.059
✓	✓		✓	2.349	**2.971**	1.094	1.434	**0.051**

Table 1. **Finetuning experiments.** Results were calculated with AEPE, except outlier percentage for KITTI validation. RD + RN is for removing random noise and weight decay. Zoom changes include an increased zoom out and a gradual reduction of the zoom in. Max crop is for using the maximal valid crop size for a batch. Random crop is for using Eq. 6 when sampling the crop size.

Model	Max zoom	WD+RN	VAL MEPE	Sintel MEPE
C*	1.5	✓	-	**3.138**
C1	1.5	x	1.622	3.264
C2	1.3	x	**1.597**	3.321

Table 2. **Removing regularization in pretraining.** Models trained 108 epochs, from initialized weights, without weight decay and random noise, for two maximal zoom values, on FChairsOcc.

Things Model	WD+ RN	Start From	Epochs	Val MEPE	Sintel MEPE
T2	✓	C*	109-159	1.843	2.613
T3	✓	C1	109-159	1.829	**2.544**
T4	x	T3	159-165	**1.817**	2.545

Table 3. The negative effect of over-training and reducing regularization on early stages. T3 and T4 were trained with larger scopes.

KITTI model	Start from	Val MEPE	Outliers
T2_K	T2	**1.474**	0.054
T3_K	T3	1.475	**0.053**

Table 4. Higher gains in early stages do not always translate to fine-tune gains. All models trained on KITTI combined.

Sintel model	WD+RN	Val MEPE	Val OCC F1
T3_SC1	✓	2.119	0.700
T3_SC2	x	**2.108**	**0.703**

Table 5. The positive effect of reducing regularization in finetune. All models trained on Sintel combined, from T3.

2. Zooming strategies. We found that applying a new random zooming range of $[0.8, 1.5]$ alone, which increases the zoom out, and gradually reducing the zoom in to 1.3, achieved considerable gains for Sintel in all evaluation parameters, with and without cropping strategy changes. Interestingly, increasing the zoom out range without any change to the crop size provided 50% of this gain. We suggest that this is additional evidence for the existing bias in small crop sizes, as explained in Sec. 3.

3. Removing artificial regularization. Removing the random noise and weight decay helped us to achieve extra 2%-3% of improvement during Sintel finetune, demonstrating the benefit of reducing augmentation in advanced stages.

Method	MEPE	Outlier %(EPE >3 px)
#1: FP (320,896)	1.651	0.057
#2: FF (370,1224)	1.594	0.059
#4: RR [0.75,0.9]	1.472	0.052
#3: FR {(0.73,0.69),(0.84,0.86),(1,1)}	1.466	0.053
#4: RR [0.9,1]	1.435	0.053
#4: RR [0.95,1]	**1.402**	**0.051**
Re-finetune		
#2: FF (370,1224) ->#4: RR [0.95,1]	1.421	0.052
#4: RR [0.95,1] ->#4: RR [0.95,1]	1.393	**0.051**
#4: RR [0.9,1] ->#4: RR [0.95,1]	**1.377**	**0.051**

Table 6. **Random cropping experiments.** Ranges are specified in [], sets in {}, and fixed sizes in (). FP is fixed partial, FF is fixed full, RR is random range (Eq. 6), FR is fixed range (Eq. 5a). Up: best results from each method described in Sec. 3.1 (the method number is on the left). Bottom: retraining experiments show that it is better to train a more regularized model in the first KITTI finetune, although it gets a lower MEPE on the first finetune.

Method	Clean	Final	Mean	Type
FlowNet2 [12]	0.377	0.348	0.362	fwd-bwd
MirrorFlow	0.39	0.348	0.369	CNN
S2DFlow	0.47	0.403	0.436	CNN
ContinualFlow [23]	-	-	0.48	CNN
SelFlow [17]	0.59	0.52	0.555	fwd-bwd
FlowDispOccBoundary [13]	0.703	0.654	0.678	CNN
IRR-PWC [11]	0.712	0.669	0.690	CNN
ScopeFlow (Ours)	**0.740**	**0.711**	**0.725**	CNN

Table 7. **Occlusion estimation comparison on Sintel.** Results were calculated with F1 score (higher is better).

5.2. Applying changes to the full training procedure

We then tested the changes from Sec. 5.1, along with all four cropping approaches described in Sec. 3.1, on the different stages of the common curriculum learning pipeline. Since we wanted to test our training changes and compare our results to other variants of the baseline architecture, we decided not to use any other dataset, other than the common pretraining or benchmarking datasets. For FChairs and FThings, all trained models were evaluated on Sintel training images, as an external test set.

FChairs pretraining For pretraining, we downloaded the newly released version of FChairs [11], which includes occlusions ground truth annotations. We trained two versions of the IRR-PWC model on FChairsOCC, for 108 epochs on 4 GPUs: (i) C1: removing weight decay and random noise (ii) C2: same as (i) with reduced zoom in. We then evaluated both models and the original model, trained by the authors with weight decay and original zoom in of 1.5, denoted by C*. Results are depicted in Tab. 2, showing

Method	Sintel – final pass							clean pass	KITTI – 2012		2015
	all	matched	unmatched	d0-10	d60-140	s0-10	s40+	all	Out-All	Avg-All	Fl-all
FlowNet2 [12]	5.739	2.752	30.108	4.818	1.735	0.959	35.538	3.959	8.8	1.8	10.41
MR-Flow[30]	5.376	2.818	26.235	5.109	1.755	0.908	32.221	**2.527**	-	-	12.19
DCFlow[31]	5.119	2.283	28.228	4.665	1.44	1.052	29.351	3.537	-	-	14.86
PWC-Net[27]*	5.042	2.445	26.221	4.636	1.475	2.986	31.07	4.386	8.1	1.7	9.6
ProFlow[18]	5.017	2.596	24.736	5.016	1.601	0.91	30.715	2.709	7.88	2.1	15.04
LiteFlowNet2[10]	4.903	2.346	25.769	4.142	1.546	0.797	31.039	3.187	6.16	1.4	7.62
PWC-Net+[26]*+	4.596	2.254	23.696	4.781	1.234	2.978	26.62	3.454	6.72	1.4	7.72
HD3-Flow[33]	4.666	2.174	24.994	**3.786**	1.647	0.657	30.579	4.788	**5.41**	1.4	6.55
IRR-PWC[11]*^	4.579	2.154	24.355	4.165	1.292	0.709	28.998	3.844	6.7	1.6	7.65
MFF[25]*+	4.566	2.216	23.732	4.664	1.222	0.893	26.81	3.423	7.87	1.7	7.17
ContinualFlow_ROB[23]*+	4.528	2.723	**19.248**	5.05	1.713	0.872	26.063	3.341	-	-	10.03
VCN[32]	4.52	2.195	23.478	4.423	1.357	0.934	26.434	2.891	-	-	**6.3**
SelFlow[17]*	4.262	2.04	22.369	4.083	1.287	**0.582**	27.154	3.745	6.19	1.5	8.42
ScopeFlow*, regularization	4.503	2.16	23.607	4.124	1.292	0.706	27.831	3.86	-	-	-
ScopeFlow*, zooming	4.317	2.086	22.511	4.018	1.311	0.739	26.218	3.696	-	-	-
ScopeFlow* (Ours)	**4.098**	**1.999**	21.214	4.028	**1.18**	0.725	**24.477**	3.592	5.66	**1.3**	6.82

Table 8. **Public benchmarks results.** Models with comparable architecture (PWC-Net) are marked with *. Models using extra data in finetune are marked with +. Our baseline model is marked with ^. We get the best EPE results in both Sintel and KITTI2012 benchmarks, surpassing all other comparable variants of our baseline model on KITTI2015, with a considerable improvement to our baseline.

that regularization is important in early stages, since removing either weight decay and random noise, or reducing the zoom-in hurt the performance.

FThings finetune We then trained three versions of IRR-PWC on FThings, for 50 epochs: (i) T2: resuming C* training with batch size of 2, with the original crop size of $[384, 768]$, (ii) T3: resuming C1 with the maximal crop size, and (iii) T4: resuming T3 without weight decay and random noise. We can infer from the results in Tab. 3: (i) increasing the scope during FChairs training leads to better accuracy on the Sintel test set, and (ii) over-training without weight decay and random noise did not improve the results on the external test set (but did on the validation).

KITTI finetune We trained two different versions, both with the same protocol, for 550 epochs on KITTI combined: (i) resuming T2 and (ii) resuming T3. Although T3 got better performance in the evaluation, after finetuning, both results were similar on KITTI validation, as shown in Tab. 4.

Sintel finetune Two different versions were trained with the same protocol, for 290 epochs on Sintel combined, both from T3: (i) with weight decay and random noise and (ii) without weight decay and random noise. The results, presented in Tab. 5, show that reducing regularization in Sintel finetune produced an extra gain.

Dynamic scene scoping Since the scoping approaches were already tested on Sintel during the initial finetune, we further tested the four different approaches for dynamic scene scoping, detailed in Sec. 3.1, on the combined KITTI dataset. The results are depicted in Tab. 6. For KITTI, cropping the maximal valid crop per batch shows noticeable improvement from using a fixed sized crop. However, for KITTI datasets, strategy S4 (Eq. 6) shows much better performance than using the maximal valid crop size. In order to find the optimal range of crop size ratios (Eq. 6), we trained different models with different ranges of crop size to image ratios $[r_{min}, r_{max}]$. All models used an upper crop size ratio limit r_{max} equal to 1 (*i.e.* the maximal valid crop for the batch), and different lower limit r_{min}, ranging from 0.5 to 1 and representing random crop sizes with different aspect ratios, which are larger than a quarter of the image.

Fig. 4 shows the training and validation accuracy as a function of the lower ratio of the range of randomized crop sizes. Specifically, the best results obtained with r_{min} equals the 0.95, as also demonstrated in Fig. 4. The validation accuracy improves consistently when increasing r_{min} from 0.5 to 0.95 and then starting to deteriorate until r_{min} is reaching the maximal valid crop size. As can be seen, when enlarging the crop size expectation, we also reduce the regularization provided by the larger number of scopes (as analyzed in Eq. 4). This observation can be considered as additional evidence of a regularization-accuracy trade-off in the training process. It also emphasizes the power of Eq. 6, in improving the training outcome, while keeping the regularization provided by partial cropping.

Re-finetune with dynamic scene scoping In order to further understand the effect of this regularization-accuracy trade-off, we re-trained three models with the best random approach ($[r_{min}, r_{max}] = [0.95, 1]$) on the KITTI combined set, using the same finetuning protocol. We took three different models, finetuned with $r_{min} \in \{0.9, 0.95, 1\}$, as the checkpoint for this second finetune.

As described in the lower part of Tab. 6, finetuning again on the KITTI dataset improved the validation accuracy for all starting points (compared to their accuracy after

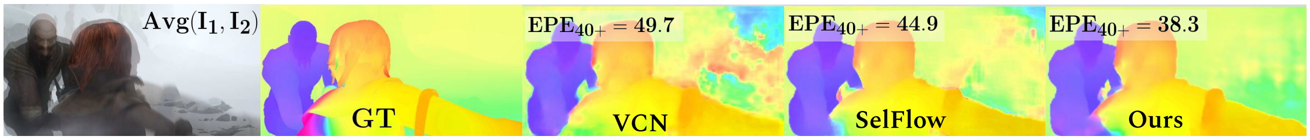

Figure 5. **Improving estimation for fast moving pixels.** A qualitative comparison with the other two leading methods on the Sintel benchmark. Images were downloaded from MPI Sintel website, evaluated online on a test image, for the category of fast pixels (40+). Left to right: averaged first and second image and flow visualization for each method. EPE_{40+} is the end point error calculated on fast pixels.

the first finetune). Surprisingly, in the second finetune, repeating the best approach (of randomizing using Eq. 6 with $([r_{min}, r_{max}] = [0.95, 1])$) did not provide the best result. The best approach was to finetune for the second time from a model with a larger range $([r_{min}, r_{max}] = [0.9, 1])$, thus stronger regularization, but lower EPE in the first finetune. We propose to consider this as additional evidence for the notion that gradually reducing regularization in optical-flow training, helps to achieve a better final local minima.

Full training insights Concluding Sec. 5.2 experiments, we suggest the following: (i) larger scopes can improve optical flow training as long as the regularization provided by small crops is not needed, (ii) range based crop size randomization (Eq. 6) is a good strategy when regularization is needed, (iii) strong regularization is required on early stages, and should be relaxed when possible, and (iv) gains on early stages do not always improve the final accuracy.

5.3. Occlusion estimation

We evaluated the occlusion estimation of our trained models, using the F1 score, during all stages of the full training. As demonstrated in Tab. 5, it appears that gains in optical flow estimation are highly correlated with improvements in occlusion estimation. This might be due to the need for a network to identify non-matchable pixels and to infer the flow from the context. Tab. 7 shows a comparison of our F1 score to other reported results, on the MPI Sintel dataset. Our updated training protocol improves the best reported occlusion result by more than 5%.

5.4. Official Results

Evaluating our best models on the MPI Sintel and KITTI benchmarks shows a clear improvement over the IRR-PWC baseline, and an advantage over all other PWC-Net variants.

MPI Sintel We uploaded results for three different versions: (i) with reduced regularization, (ii) with improved zooming schedule and (iii) with the best dynamic scoping approach. As Tab. 8 shows, there is a consistent improvements on the test set. This is congruent with the results in Tab. 1, obtained on the validation set.

At the time of submission, our method ranks first place on the MPI Sintel benchmark, improving two-frame methods by more than 10%, surpassing other competitive methods trained on multiple datasets, with multiple frames and all other PWCNet variants, using an equal or larger size of trainable parameters. On the clean pass, we improve the IRR-PWC result by 20 ranks and 7%. Interestingly, analyzing Sintel categories in Tab. 8, our model is leading in the categories of fast pixels ($S40_+$) and non-occluded pixels, while also producing the best estimation for occluded pixels among two frame methods. This is consistent with our insights on these challenging categories from Sec. 3. Fig. 5 shows a comparison of our method in the category of fast pixels, with the other two leading methods on Sintel.

KITTI On KITTI 2012 and KITTI 2015, we saw a consistent improvement from the baseline model results, of more than 19.7% and 12% respectively, surpassing all other published methods of the popular PWC-Net architecture, and achieving state-of-the art EPE results among two frame methods. Since our training protocol can be readily applied to other methods, we plan, as future work, to test it on other leading architectures.

6. Conclusions

While a lot of effort is dedicated for finding effective network architectures, much less attention is provided to the training protocol. However, when performing complex multi-phase training, there are many choices to be made, and it is important to understand, for example, the proper way to schedule the multiple forms of network regularization. In addition, the method used for sampling as part of the augmentation process can bias the training protocol toward specific types of samples.

In this work, we show that the conventional scope sampling method leads to the neglect of many challenging samples, which hurts performance. We advocate for the use of larger scopes (crops and zoom-out) when possible, and a careful crops positioning when needed. We further show how regularization and augmentation should be relaxed as training progresses. The new protocol developed has a dramatic effect on the performance of our trained models and leads to state of the art results in a very competitive domain.

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974).

References

[1] Mathieu Aubry, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[2] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *CVPR*, pages 2710–2719, 07 2017. 2

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 2

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, Dec 2015. 2, 5

[5] David J. Fleet and Y. Weiss. Optical flow estimation. In *Handbook of Mathematical Models in Computer Vision*, 2006. 2

[6] David Gadot and Lior Wolf. Patchbatch: a batch augmented loss for optical flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[7] 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. 2

[8] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb 2008. 2

[9] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[10] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn - revisiting data fidelity and regularization. *arXiv preprint arXiv:1903.07414*, 2019. 2, 7

[11] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019. 1, 2, 3, 4, 6, 7

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4, 6, 7

[13] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 6

[14] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016. 2

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989. 1

[16] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. DDFlow: Learning Optical Flow with Unlabeled Data Distillation. In *AAAI*, 2019. 2

[17] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *CVPR*, 2019. 2, 3, 6, 7

[18] Daniel Maurer and Andrés Bruhn. Proflow: Learning to predict optical flow. In *BMVC*, 2018. 7

[19] 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*, 126(9):942–960, Apr 2018. 2, 5

[20] 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 *CVPR*, pages 4040–4048, 06 2016. 2

[21] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGC-Net: Dense geometric correspondence network. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 3

[22] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[23] Michal Neoral, Jan Šochman, and Jiří Matas. Continual occlusions and optical flow estimation. In *14th Asian Conference on Computer Vision (ACCV)*, Dec. 2018. 2, 3, 6, 7

[24] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[25] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik B Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2, 3, 7

[26] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 2, 7

[27] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2, 3, 5, 7

[28] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018. 2

[29] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *2013 IEEE International Conference on Computer Vision*, pages 1385–1392, Dec 2013. 2

[30] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7

[31] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, 2017. 2, 7

[32] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 2, 7

[33] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR*, 2019. 7

[34] Shay Zweig and Lior Wolf. Interponet, a brain inspired neural network for optical flow dense interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2