Machine Perception 2021 - Project report

Optical Flow Estimation

Maxime Raafat
raafatm@student.ethz.ch

Nicolas Muntwlyer
municola@student.ethz.ch

ABSTRACT

Human optical flow plays an important role in the analysis of human action. We present a deep learning based approach to predict the optical flow in multi-human scenes. Our method is based on PWC-Net, a general optical flow estimation network. We use the Multi-human optical flow dataset to fine-tune the before-mentioned model, and introducing an iterative refinement procedure and a cyclic loss to achieve a significant improvement in the flow computation. We furthermore introduce a novel architecture which runs in harmony with PWC-Net and makes additional use of pre-computed segmentation masks.

1 INTRODUCTION

Predicting an accurate optical flow plays a core part in many computer vision applications such as autonomous driving, video compression and video editing. For the past years classical methods have strongly dominated the field, while most methods implement an energy minimization pioneered by Horn and Schunck [3]. However recent advances in machine learning and deep learning have led to innovative learning-based approaches[8, 10] that not only outperform classical methods, but also make flow computation significantly faster and enable real-time applications.

Predicting an accurate optical flow is a challenging task to solve : since flow estimation requires per-pixel localization, a deep learning model not only needs to learn relevant features from a scene frame, but also requires to match those features between two input images. A further challenge is the increasing demand for real-time performance for applications on mobile device.

While PWC-Net [10] achieves impressive generalizability to predict optical flow, many applications care specifically for human flow estimation. Ranjan et al. [9] therefore generated a *Multi-human optical flow* (MHOF) dataset, which we use to fine-tune PWC-Net on human scenes.

In this work we fine-tune a PWC-Net-based architecture on the MHOF dataset and present an iterative refinement procedure (similar to IRR-NET [5]) together with a newly introduced cycle-loss to further increase the flow prediction accuracy. Lastly we explore an approach combining the intermediate flow-predictions of PWC-Net with a novel network, Seg-Net, inspired from LiteFlowNet [4] and which makes additional use of the segmentation masks of the two input images.

2 RELATED WORK

In many areas of computer vision, powerful deep learning models are progressively replacing classical methods. For optical flow estimation it started with the pioneering work of Dosovitskiy et al. [1]. Although they could not achieve state-of-the-art results, they showed that CNN's have high potential for flow prediction tasks.

Figure 1: Example of an optical flow prediction (middle) between Image 1 (left) and and Image 2 (right)

By stacking multiple FlowNet modules on top of each other, Ilg et al. [6] developed FlowNet2, competing on par with state-of-the-art methods. Although Flownet2 achieves astonishing results, its size is large and does not yet outperform classical approaches. Hui et al. [4] came up with a network based on FlowNet2 while being 30 times smaller in model size. By introducing a cascaded flow inference relying on sub-pixel refinement (which will be the foundation for our novel architecture Seg-Net), LiteFlowNet not only outperforms FlowNet2 in size, but is also 1.36 times faster in running speed.

With PWC-Net, Sun et al. [10] presented a new simplistic model with far less parameters than Flownet2 [6] and significantly outperforming it. PWC-Net is based on the well-established principles of pyramidal processing, warping and the use of a cost volume. By reducing PWC-Net to only one pyramid layer, which is repeatedly called to iteratively update the final flow through the addition of the individual residual flows, IRR-Net [5] further reduces the number of parameters while not losing in prediction accuracy.

Creating datasets for supervised optical flow learning is a challenging task which is simplified if data is synthesized such that ground truth is known. While large scale datasets like FlyingChairs focus on general optical flow prediction, MHOF [9] is first to introduce a dataset tailored for human motion prediction. Other approaches for self-supervised learning [7, 11] have also been developed in order to remove the need for ground truth optical flow generation procedures, however our work builds on PWC-Net, which requires ground truth samples.

3 METHOD

3.1 Problem Statement

We solve the following task: given two input images I_1 and I_2, predict the optical flow $\mathbf{w}$ between the two images. As additional input we get a body segmentation mask for both images and we are provided with the PWC-Net model pre-trained on the FlyingChairs dataset. The images contain multi-human scenes with arbitrary backgrounds. We measure the model accuracy with the average End Point Error (EPE) over all image pixels.

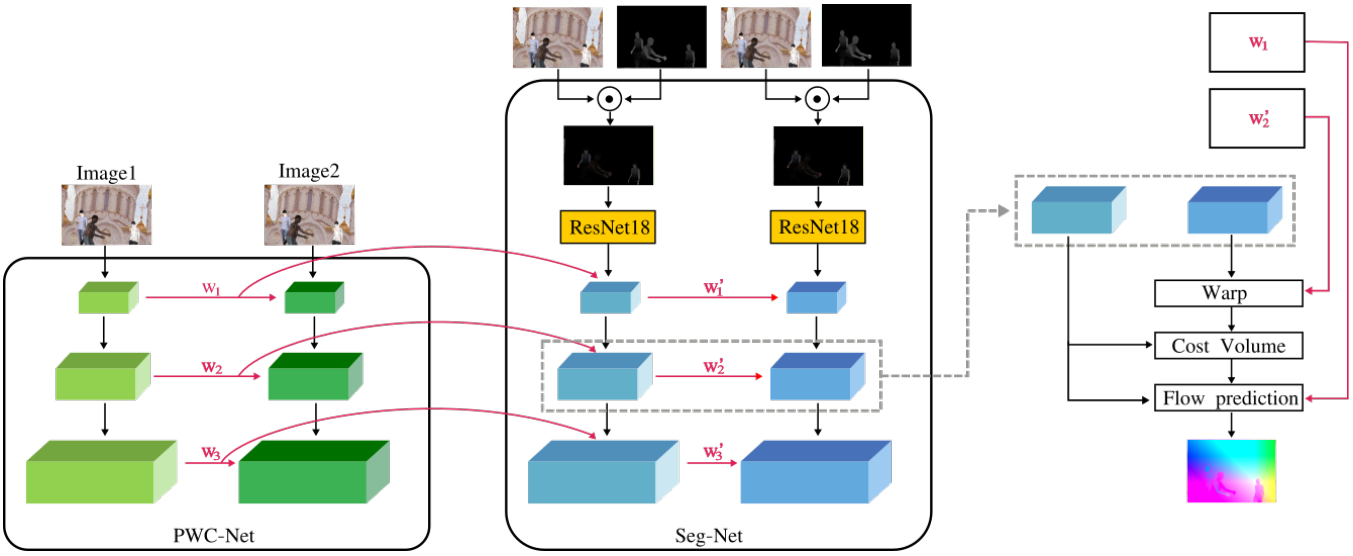

Figure 2: Visualization of our whole architecture. We propagate the by PWC-Net predicted optical flow at different levels to Seg-Net. Seg-Net first builds an hierarchical feature pyramid with ResNet18 [2] from the segmentation masks. Together with the from PWC-Net propagated flow we then predict the flow for the respective layer. After going through all levels of the pyramid we compute the final flow.

3.2 Network Architecture

Figure 2 gives an overview of the network architecture. Our method is based on three core ideas: Iterative Residual Refinement (IRR), cyclic consistency and the use of the segmentation masks. We explain each concept in the next three subsections.

3.3 Iterative Residual Refinement

Figure 3 depicts this whole iterative residual refinement (IRR) process. Given reference image I_1 and target image I_2 we predict the optical flow w_{base} between I_1 and I_2. We then apply the optical flow to I_1 and get image I_2' which should be close to I_2. The idea is to now again predict the flow but this time between I_2' and I_2, hence we estimate the residual flow, which we call w_{res}^1. We can repeat this process to iteratively refine the estimated flow. The final flow is then the sum of the base flow prediction and all the residual flow refinements:

$$w_{final} = w_{base} + \sum_{i=1}^{n} w_{res}^i$$

There exists a trade-off between accuracy and runtime. More iterations tend to result in better flow estimation, but increases the runtime in a linear fashion. To this end we use n = 3.

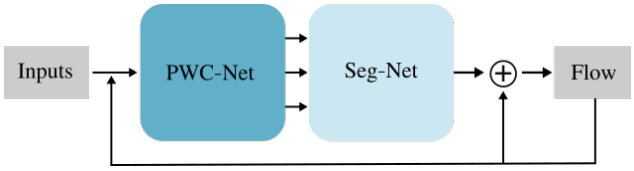

Figure 3: Iterative Residual Refinement. We iteratively refine the final flow by adding the residual flow from the previous flow estimation.

3.4 Loss function

The authors of PWC-Net suggest to use a multi-scale training loss. Let w_{I_1,I_2}^l denote the predicted flow field between I_1 and I_2 at the l-th pyramid level, and w_{GT}^l be the ground truth flow between I_1 and I_2 at that level.

$$\mathcal{L}_{forward}(w_{I_1,I_2}, w_{GT}) = \sum_{l=l_0}^{L} \alpha_l \sum_x (|w_{I_1,I_2}^l(x) - w_{GT}^l(x)| + \epsilon)^q$$

Inspired by [CycleGAN] we perform a cyclic flow estimation. After we have predicted the flow w_{I_1,I_2} from I_1 to I_2 we apply w_{I_1,I_2} to I_1 to get I_2', which should be close to I_2. We then use our network to predict the multi-scale flow w_{I_2',I_1}^l from I_2' to I_1. This flow can be seen as the inverse flow and ideally $w_{I_2',I_1}^l = -w_{GT}^l$, and we yield the below cyclic loss, also depicted in Figure 4.

$$\mathcal{L}_{cycle}(w_{I_2',I_1}, w_{GT}) = \sum_{l=l_0}^{L} \alpha_l \sum_x (|w_{I_2',I_1}^l(x) + w_{GT}^l(x)| + \epsilon)^q$$

Lastly we regularize the weights with an L2-norm where Θ denotes the network parameters.

$$\mathcal{L}_{reg}(\Theta) = \|\Theta\|_2$$

We conclude on the final loss-function :

$$\begin{aligned}\mathcal{L}(I_1, I_2, w_{I_1,I_2}, w_{I_2',I_1}, w_{GT}, \Theta) = \mathcal{L}_{flow}(w_{I_1,I_2}, w_{GT}) \\ + \mathcal{L}_{cycle}(w_{I_2',I_1}, -w_{GT}) \\ + \mathcal{L}_{reg}(\Theta)\end{aligned} \quad (1)$$

Note that we start training by only using the forward flow loss and only add the cycle-consistency loss after we get an accurate

forward loss. For both forward-loss and cycle-loss we chose an $\epsilon = 0.01$ and $q = 0.4$.

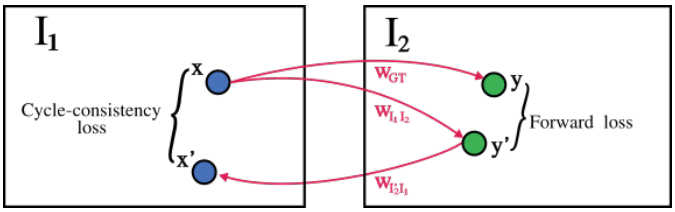

Figure 4: Visualization of the cyclic consistency. After we have predicted the forward flow we take the predicted image and estimate the inverse flow. Ideally we would be back to our original point

3.5 SegNet architecture

The original PWC-Net is less accurate than traditional approaches on the clean pass of the MPI Sintel dataset, as many classical methods use image edges to refine motion boundaries which are perfectly aligned in the clean pass. We therefore want to address this shortcoming by utilising the segmentation maps of each image, which have clear boundaries. Additionally they should help our network to better differentiate between back- and foreground movement.
PWC-Net takes as an input two images on which it performs convolutional operations in order to extract image features. Seg-Net proceeds in a similar fashion, but does some pre-processing on the images first. We first multiply each input image with its respective segmentation mask, which results in the same image as before but with a black background instead of an arbitrary one. This will then be fed into ResNet18 which yields useful image features. Those frozen features (due to frozen ResNet18 weights) are then progressively convolved into different tensors of increasing dimensions, which serve as an input for the cascaded flow inference (inspired by LiteFlowNet [4]). Each sub-pixel refinement layer (layer of the cascading flow inference method) is constructed in the same way : we warp the augmented features of masked image 2 (extracted from ResNet18 [2]) with the upscaled flow generated in the previous layer and compute the correlation between this warping and the features for masked image 1. We then concatenate the correlation with the matching flows generated in PWC-Net and convolve this in order to again obtain a flow. Since our first layer doesn't have a previous layer, we feed it with the smallest flow generated in PWC-Net. Figure 2 depicts the complete model architecture.

3.6 Data Augmentation

A large difficulty was to control over-fitting due to the comparably small training set of 8343 pairs of training and 1647 pairs of validation images. We therefore applied several data augmentation transformations. We randomly translate, crop and rotate the images and additionally add Salt and Pepper noise and random Gaussian blur to further diversify our data.

4 EVALUATION

Our main results are reported in Table 1. The official PWC-Net serves as a baseline. By adding some dropout layers and further data augmentation we then fine-tune the PWC-Net on the MHOF

Method	Average EPE
(1) PWC	1.724
(2) PWC-ft ($1e^{-4}$)	0.948
(3) PWC + IRR ($1e^{-5}$)	0.789
(4) PWC + IRR + Cycle ($1e^{-5}$)	0.646
(5) PWC + IRR + Cycle ($1e^{-6}$)	0.546
(6) PWC* + SegNet + IRR ($1e^{-4}$)	0.558
(7) PWC* + SegNet + IRR + Cycle ($1e^{-4}$)	0.558

Table 1: Comparison of End Point Error (EPE) between the different network architectures on the Multi-Human Optical Flow (MHOF) dataset

[9] dataset for 400 Epochs with a learning rate of $1e^{-4}$, which gave us a 45% error reduction with an average EPE of 0.948. By adding the iterative residual refinement (IRR) we further decrease the error by 10% and the cyclic loss gave us another 8% improvement. Interestingly the network performs better with a smaller learning rate (see run (5) in Table 1). Our final run with Seg-Net makes uses of (frozen) pre-trained weights from run (5) (see Table 1), which we call PWC*. Training Seg-Net with the provided pre-trained PWC-Net model on the *on Flying Chairs* dataset yields a quite high loss which struggles to predict accurate flows (validation EPE converging towards around 1.5).
Note that run (5) iteratively refined the flow, but only the flow after one iteration gets propagated to the Seg-Net layers. By propagating an refined flow, Seg-Net would receive a strongly accurate flow already, and might produce much preciser flows. This was unfortunately not implemented due to time constraints.
Although achieving satisfying results already, we believe that a larger data sample would considerably improve generalizability, since Ranjan et al. [9] obtained bluffing results with 3 times as much data as provided for this project.

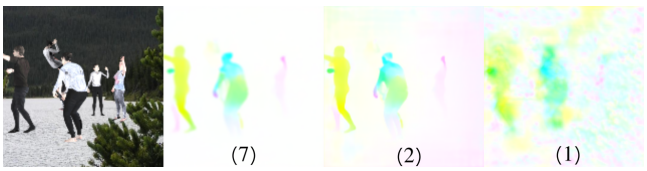

Figure 5: Comparison of the final flow prediction between method (1),(2) and (7). We see that method (7) has clearer boundaries and less artifacts

5 CONCLUSION

We presented an iterative refinement procedure and a cycle-consistency loss, which significantly improve the final flow prediction. We further introduce a novel network architecture that makes use of the pre-computed body segmentation masks and runs concurrent to PWC-Net. Our final submission combines all mentioned ideas and achieves an End Point Error reduction of 67%. We believe that with more training data and careful loss manipulation we can even further improve our flow estimation.

REFERENCES

[1] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 2758–2766.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[3] Berthold KP Horn and Brian G Schunck. 1981. Determining optical flow. *Artificial intelligence* 17, 1-3 (1981), 185–203.

[4] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8981–8989.

[5] Junhwa Hur and Stefan Roth. 2019. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5754–5763.

[6] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2462–2470.

[7] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. 2019. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4571–4580.

[8] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4161–4170.

[9] Anurag Ranjan, David T Hoffmann, Dimitrios Tzionas, Siyu Tang, Javier Romero, and Michael J Black. 2020. Learning multi-human optical flow. *International Journal of Computer Vision* 128, 4 (2020), 873–890.

[10] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8934–8943.

[11] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2018. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898* (2018).