Layered Optical Flow Estimation Using a Deep Neural Network with a Soft Mask

Xi Zhang, Di Ma, Xu Ouyang, Shanshan Jiang, Lin Gan, Gady Agam
Illinois Institute of Technology
Chicago, IL 60616
{xzhang22, dma2, xouyang3, sjiang20}@hawk.iit.edu, {lgan, agam}@iit.edu

Abstract

Using a layered representation for motion estimation has the advantage of being able to cope with discontinuities and occlusions. In this paper, we learn to estimate optical flow by combining a layered motion representation with deep learning. Instead of pre-segmenting the image to layers, the proposed approach automatically generates a layered representation of optical flow using the proposed soft-mask module. The essential components of the soft-mask module are maxout and fuse operations, which enable a disjoint layered representation of optical flow and more accurate flow estimation. We show that by using masks the motion estimate results in a quadratic function of input features in the output layer. The proposed soft-mask module can be added to any existing optical flow estimation networks by replacing their flow output layer. In this work, we use FlowNet as the base network to which we add the soft-mask module. The resulting network is tested on three well-known benchmarks with both supervised and unsupervised flow estimation tasks. Evaluation results show that the proposed network achieve better results compared with the original FlowNet.

1 Introduction

Optical flow estimation is a crucial and challenging problem with numerous applications in computer vision. Traditional differential methods for estimating optical flow include variational methods [Horn and Schunck, 1981], which uses a regularization term and provide a global solution for optical flow. Various improvements of these initial formulations have been proposed over many years.

Layered models of optical flow offer an easy performance boost for optical flow estimation. Disjointly splitting the optical flow into layers enables easier modeling of optical flow in each layer. Such a representation is especially helpful for small object motion estimation, as many optical flow estimation techniques are biased towards motion in large areas. Layered representation also improves flow computation on flow field boundaries by handling the smoothness constraint separately in each layer.

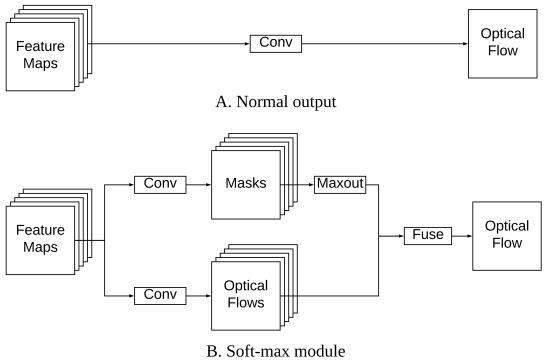

Figure 1: Illustration of the structure of the proposed soft-mask module compared with traditional linear optical flow network.

FlowNet [Dosovitskiy *et al.*, 2015] was the first to use a deep neural network for end-to-end optical flow estimation. FlowNet is fundamentally different from established differential approaches. As traditional differential optical flow estimation techniques perform well and are well established, several deep learning based approaches try to bridge the gap between traditional approaches and deep learning based approaches by using the best of both sides. For example, Ranjan et al. [Ranjan and Black, 2017] use a pyramid representation of flow and residual flows to address large flow displacement estimation. Several approaches [Ren *et al.*, 2017][Ahmadi and Patras, 2016][Yu *et al.*, 2016] investigated the basic principles of flow estimation and proposed unsupervised network training.

Our work combines the idea of using a layered optical flow representation with a deep neural network structure. Unlike previous approaches [Yang and Li, 2015], where the layered representation is generated separately, the layered representation in the proposed approach is inferred internally and automatically when training the neural network. We achieve this by designing a soft-mask module. The soft-mask module is a network structure which splits optical flow into layers using disjoint real-valued masks. As the masks are not binary, we use the term 'soft' to refer to them. The soft-mask module offers a more accurate flow estimation due to two unique characteristics. The first is its ability to represent estimated flow using disjoint layers, which results in a more focused

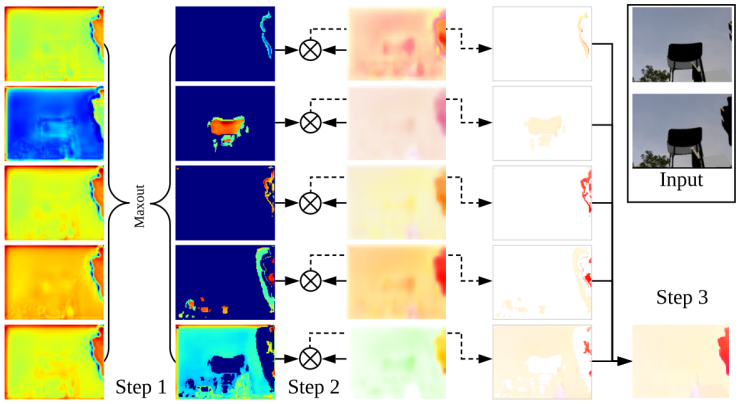

Figure 2: Pipeline of the soft-mask module. Step 1: Setting maxout and non-max pixels in masks to zero. Step 2: Pixel-wise multiplication with generated intermediate flows. Step 3: Generating final flow by summing up flows from every mask.

and simpler flow estimation for each layer. Second, compared with the linear flow output in FlowNet, the flow estimated using the soft-mask module is quadratic in terms of input features. This allows the soft-mask module to better fit more complicated optical flow patterns. The idea of using the soft-mask module is similar to the maxout networks proposed by Goodfellow [Goodfellow *et al.*, 2013], where the output of a neuron is the max of a set of inputs. The proposed soft-mask module extends the maxout operation to 2D. In addition, instead of keeping max values only, we zero-out non-max values and use them when fusing layered optical flows.

In this work, the soft-mask module is added to FlowNet by replacing the output layer of the network with the soft-mask module. While more generally, the soft-mask module could be used in other per-pixel prediction tasks such as semantic segmentation and single image depth estimation, we focus in this paper on its application to optical flow estimation.

We show that by using the soft-mask module, we boost the performance of FlowNet when tested on several public datasets such as the Flying Chairs [Dosovitskiy *et al.*, 2015], Sintel [Butler *et al.*, 2012], and KITTI [Geiger *et al.*, 2012]. We further show that both supervised and unsupervised flow estimation methods benefit from using the soft-mask module.

1.1 Related Work

Our work effectively combines ideas from using layered representation in classical optical flow approaches with recent deep learning approaches.

Layered approaches. Using layered approaches in motion estimation are commonly used to overcome discontinuities and occlusions. A layered approach has been proposed [Darrell and Pentland, 1991] where a Bayesian model for segmentation and robust statistics is incorporated. Recent work [Sun *et al.*, 2010] use affine motion to regularize the flow in each layer, while Jepson and Black [Jepson and Black, 1993] formalize the problem using probabilistic mixture models. Yang [Yang and Li, 2015] fit a piecewise adaptive flow field using piecewise parametric models while maintaining a global inter-piece flow continuity constraint. Exploiting recent advances in semantic scene segmentation, [Sevilla-Lara *et al.*, 2016] use different flow types for segmented objects in different layers. Hur and Roth [Hur and Roth, 2016] treat semantic segmentation and flow estimation as a joint problem.

Deep learning approaches. Deep neural networks have been shown to be successful in many computer vision tasks including object recognition and dense prediction problems [Long *et al.*, 2015]. FlowNet attempts to solve optical flow estimation using a deep neural network. FlowNet provides an end-to-end optical flow learning framework which serves as a base model for many later works. Two notable existing works include [Zhou *et al.*, 2016] and [Flynn *et al.*, 2016]. Masks generated by these approaches are normalized and then used as weight maps and multiplied with features before a final prediction is made. The masks used in our proposed work is different from masks in prior works. Instead of normalizing mask values across different channels, the proposed soft-mask module applies a maxout operation among channels. Because of the maxout operation, the soft-mask module could segment objects better with different flows. In Section 3.5, we compare the results of different ways of using masks in optical flow prediction and verify that the proposed soft-mask module over performs other existing methods.

1.2 Novel Contribution

In this work, we extend FlowNet and improve its performance in several ways. First, we propose combining a traditional layered approach for optical flow estimation with deep learning. The proposed approach does not require pre-segmentation of images. Instead, the separation of layers is done automatically when training the network. Second, a soft-mask module is proposed. This soft-mask module implements a channel-wise maxout operation among masks. As a result, the estimated optical flow is separated into layers, each of which contains optical flow that is estimated using a quadratic function. Third, we extend FlowNet by adding the proposed soft-mask module in the output layers. The resulting network is trained and compared with both supervised and unsupervised optical flow estimation approaches using neu-

ral networks. Experimental results show that the proposed network structure achieves lower error in each experimental group.

2 Methodology

2.1 Soft-mask Module

FlowNet was the first work to use a deep convolutional neural network for optical flow estimation. The network architecture used by FlowNet is very similar to the structure of a classical auto-encoder, where optical flows are generated using deconvolution at each scale level of the image pyramid. To refine flow estimations, shortcuts are built to connect layers of corresponding levels in the encoder and decoder layers. Consider a single computation of convolution, and for simplicity assume that f represents both horizontal and vertical components of the output flow. Given $X \in \mathbb{R}^{s\times s\times c}$, representing an input feature volume, where s is the kernel size, and c is the number of channels, FlowNet employs a linear activation to compute optical flow:

$$f = X^T W + b \tag{1}$$

Given that actual optical flow fields are nonlinear and piecewise smooth, using a linear function to fit the flow field shifts the non-linearity to the convolutional layers making the learning there more difficult. Using the soft-mask module proposed in this paper to replace the linear output of optical flow estimation, we can separate the optical flow field into multiple layers. The flow estimation in each layer is smooth and is easier to estimate compared with the original model. This results in a more accurate and flexible optical flow estimation.

An illustration of the soft-mask module is shown in Figure 1. The essential part of the soft-mask module is its dual-branch structure which contains a mask branch and an optical flow branch. The input feature maps represented as a set of volume feature vectors, $X \in \mathbb{R}^{s\times s\times c}$ are fed to both branches. The most significant contribution of this work is the separation of the optical flow field to multiple layers. For a separation into k layers, k masks will be generated in the mask branch as illustrated in Figure 1. This requires k convolutional filters $\{W_n^m, b_n^m\}_{n=1}^k$ in the mask branch. Correspondingly, the same number of filters are used in the optical flow branch $\{W_n^f, b_n^f\}_{n=1}^k$. The mask and intermediate optical flow are then computed as follows:

$$\begin{aligned} m_n &= X^T W_n^m + b_n^m \qquad \text{for } n = 1\dots k \\ f_n &= X^T W_n^f + b_n^f \qquad \text{for } n = 1\dots k \end{aligned} \tag{2}$$

Thus, given k filters, we obtain k corresponding pairs of mask and intermediate optical flow. By using k filters in the optical flow branch and generating k intermediate optical flow fields, we assume that each filter works independently and model a single type or a few types of object motions. Correspondingly, filters in the mask branch are expected to mask out parts with consistent motions by being high in certain regions and low in others. This leads us to use a maxout operation to extract mask entries with maximal activation along the channel axis. After the maxout operation, for each mask $m_n(n = 1\dots k)$, all entries will be zero-out except for entries whose activation values are maximal in some regions among all masks. We denote the masks after maxout using $\{m'_n\}_{n=1}^k$. Following the maxout operation, there is no intersection among masks, and the union of all $m'_n, n = 1\dots k$ has activation in the full region. The maxout is given by:

$$m'_n = \begin{cases} m_n, & \text{if } m_n = \max\limits_{p=1\dots k}(m_p) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $n = 1\dots k$. Note that the masks produced by maxout are not converted to binary values, thus resulting in soft-masks. By using soft masks, we can mask out irrelevant parts and prioritize values in each layer. In the proposed approach, masks generated by the maxout operation are applied to corresponding intermediate optical flow field by element-wise multiplication as shown below:

$$f'_n = \begin{cases} m'_n \times f_n, & \text{if } m'_n \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $n = 1\dots k$ The result of the above computation is a set of disjoint optical flow layers, each of which represents a certain type of motion. An illustration of the soft-mask module works is shown in Figure 2 and results of generated masks are shown in Figure 4.

2.2 Quadratic Fitting of Optical Flow

Objects moving in different ways, result in different types of motion. The underlying optical flows are non-linear and locally piecewise smooth.

There are two advantages in the proposed soft-mask module that make the estimation of optical flow easier. The first advantage is due to using maxout in the mask generation. By keeping only the maximal value among all masks, the optical flow is separated into multiple disjoint layers. The qualitative results as shown in Figure 4 demonstrate that the soft-mask module allows the resulting masks to separate the flow field into pieces according to detected motion types. In addition, the masks detect the boundary of each motion piece, thus allowing the estimation of optical flow on boundaries to be more accurate. The second advantage of using the proposed soft-mask module is that the output is quadratic in terms of feature maps X fed to the module. To see this, consider the computation of masks and intermediate optical flow shown in Equation 2. The computation of non-zero f'_n could be written as:

$$\begin{aligned} f'_n &= m'_n \times f_n \\ &= (X^T W_n^m + b_n^m) \times (X^T W_n^f + b_n^f) \\ &= W_n^{mT} X X^T W_n^f + X^T (b_n^f W_n^m + b_n^m W_n^f) + b_n^m b_n^f \end{aligned} \tag{5}$$

As can be observed in the above equation, the representation of f'_n is quadratic in terms of the variable X.

To better illustrate the advantage in using the soft-mask module with respect to linear output. Consider the 1D example shown in Figure 3. In this example, function values

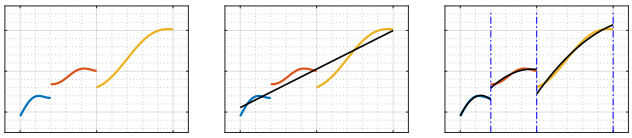

Figure 3: A: Given data. B: Fitting using a linear function. C: Fitting using a piecewise quadratic function.

are smooth in three separate domains. The improvement of fitting data using a piecewise quadratic function is shown in Figure 3 B and C.

2.3 Regularization for Unsupervised Training

Training an unsupervised neural network for optical flow estimation is possible by using a network similar to FlowNet for base optical flow inference followed by a spatial transform network (STN). To show that proposed soft-mask module can improve flow estimation using the same framework, we add the soft-mask module to FlowNet and use it as a base optical flow inference network in an unsupervised training framework.

The smoothness term which is used by all above unsupervised approaches plays a significant role in regularizing the local consistency of optical flow. We use the bending energy regularization [Rohlfing *et al.*, 2003]:

$$\varphi(\mathbf{u},\mathbf{v}) = \sum((\frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2})^2 + (\frac{\partial^2 \mathbf{u}}{\partial \mathbf{y}^2})^2 + 2(\frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}\partial \mathbf{y}})^2) + \sum((\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}^2})^2 + (\frac{\partial^2 \mathbf{v}}{\partial \mathbf{y}^2})^2 + 2(\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}\partial \mathbf{y}})^2)$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{H \times W}$ are estimated horizontal and vertical components of the optical flow field.

3 Empirical Evaluation

3.1 Benchmark

We evaluatethe our performance of the proposed approach on three standard optical flow benchmarks: Flying Chairs [Dosovitskiy *et al.*, 2015], Sintel [Butler *et al.*, 2012], and KITTI [Geiger *et al.*, 2012]. We compare the performance of the proposed approach to both supervised methods such as: FlowNet(S/C) [Dosovitskiy *et al.*, 2015][Ilg *et al.*, 2017], SPyNet [Ranjan and Black, 2017], as well as DeepFlow [Weinzaepfel *et al.*, 2013], and EpicFlow [Revaud *et al.*, 2015]. We compare the proposed approach to methods including: DSTFlow [Ren *et al.*, 2017], USCNN [Ahmadi and Patras, 2016], and back-to-basic unsupervised FlowNet (bb-FlowNet) [Yu *et al.*, 2016].

Recently, FlowNet 2.0, a follow-up work of FlowNet, achieved state of the art results on most datasets. The architecture of FlowNet 2.0 [Ilg *et al.*, 2017] uses several FlowNets and contains cascade training of the FlowNets in different phases. Since the focus of this paper is on using the soft-mask module to boost performance of a single network, we do not include FlowNet 2.0 in our evaluation. Note that the proposed soft-mask module can be incorporated into FlowNet 2.0.

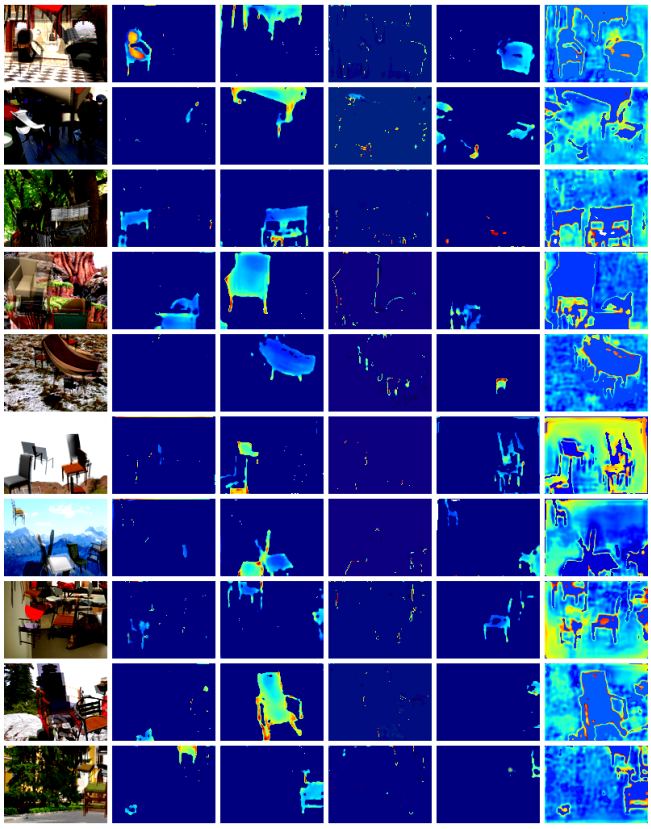

Figure 4: Examples of masks generated by the proposed soft-mask module. Five masks are generated for each input image pair which is shown as overlaid. Colors are according to normalized values in each image separately.

3.2 Network Structure

The goal of this paper is to show how the performance of existing optical flow networks can be improved by replacing the normal optical flow output layer with the proposed soft-mask module. We choose FlowNetS and FlowNetC as the base networks and replace their optical flow output layers with a soft-mask module. Using the layered optical flow estimation (LOFE) proposed in this paper, we term the resulting modified networks: FlowNetS+LOFE and FlowNetC+LOFE, respectively. To make our evaluation more complete, we also replaced the output layer of SPyNet with the soft-mask module. The resulting model is labeled as SPyNet+LOFE.

3.3 Training Details

Training of Soft-mask Module. Both FlowNetS+LOFE and FlowNetC+LOFE could be built by simply replacing the output layer with a soft-mask module. Data scheduling has been shown very useful when training FlowNetS and FlowNetC in [Ilg *et al.*, 2017]. For supervised networks, we used pre-trained weights of two networks from FlowNet 2.0 [Ilg *et al.*, 2017] both trained with and without fine-tuning. Then, for each dataset in our experiment, we trained the soft-mask module by fixing the pre-trained weights. We compared our proposed method to FlowNetS and FlowNetC

Method	Flying Chairs	Sintel Clean		Sintel Final		KITTI	Time (s)
	Test	Train	Test	Train	Test	Train	
EpicFlow	2.94	2.40	4.12	3.7	6.29	3.47	16
DeepFlow	3.53	3.31	5.38	4.56	7.21	4.58	17
FlowNetS+schd	2.69	4.42	6.86	5.25	7.46	8.64	0.12
FlowNetC+schd	2.35	4.24	6.82	5.07	8.34	8.85	0.23
FlowNetS+schd+ft	2.49	4.08	6.98	4.75	7.52	8.26	0.12
FlowNetC+schd+ft	2.17	3.79	6.83	4.59	7.99	8.35	0.23
SPyNet	2.63	4.23	6.82	5.67	8.49	9.12	0.11
SPyNet+LOFE	2.33	3.99	6.52	5.30	8.49	9.12	0.31
FlowNetS+LOFE+schd	2.49	4.20	6.80	4.88	**7.36**	8.03	0.36
FlowNetC+LOFE+schd	2.17	3.96	6.78	**4.54**	7.59	8.14	0.47
FlowNetS+LOFE+schd+ft	2.37	3.81	6.44	4.62	7.45	**7.98**	0.35
FlowNetC+LOFE+schd+ft	**2.02**	**3.49**	**6.21**	4.56	**7.51**	8.01	0.46

Table 1: Average end point errors (EPE) of the proposed networks compared to several existing methods on Flying Chairs and Sintel Clean datasets. EpicFlow and DeepFlow are traditional methods which do not use neural networks. All other methods in the table are trained with supervised data. Bold font indicates the most accurate results among the network-based methods.

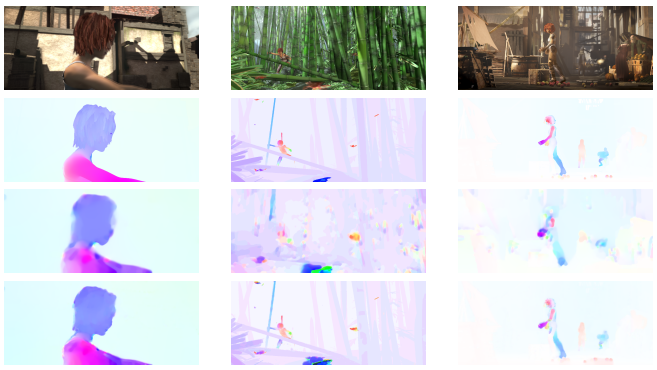

Figure 5: Examples of predicted flows compared with results from FlowNetC. First row: input image pair (overlaid). Second row: ground truth optical flow. Third row: flow results of FlowNetC. Fourth row: flow results of FlowNetC+LOFE.

trained using data scheduling. We are aware that it is unfair by training soft-mask module more. Therefore, when compared to baseline models, we trained their output layers and fixed other parts in the same way.

Data augment. Various types of data augmentation are used during training. We applied rotation at random within $[-17°, 17°]$. A random translation within $[-50, 50]$ pixels was applied to both horizontal and vertical directions. In addition, following [Ranjan and Black, 2017] we included additive white Gaussian noise sampled uniformly from $\mathcal{N}(0, 0.1)$. We also applied color jitter with additive brightness, contrast and saturation sampled from a Gaussian, $\mathcal{N}(0, 0.4)$. All data augmentations were done using GPU during training.

3.4 Results

Evaluation was done with compared methods in two groups according to whether the training of the method is unsupervised or supervised. Table 1 shows the endpoint error (EPE) of the proposed network and several well-known methods. The endpoint error measures the distance in pixels between known optical flow vectors and estimated ones. Except for EpicFlow and DeepFlow, all other methods were trained in a supervised manner. We compare results of unsupervised methods in Table 2.

Supervised methods. The proposed FlowNetS+LOFE and FlowNetC+LOFE tested in this group use $k = 10$ for the number of layers. As can be seen in Table 1, FlowNetS+LOFE and FlowNetC+LOFE achieve better performance compared with methods not using the module. We observe that the performance of SPyNet is also boosted by replacing the optical flow output layer with the soft-mask module. Considering the computation time we observe a small time increment when using the soft-mask module, and which is in an acceptable range.

Qualitative results are shown in Figure 5. It could be observed that due to the soft-mask module in FlowNetC+LOFE, generally the network has a better prediction on flow boundaries over FlowNetC.

Unsupervised methods. Training optical flow estimation networks without supervision is straight forward. The results are shown in Table 2. As can be observed, the proposed networks achieve the best performance except for the KITTI dataset where the proposed approach achieved 2nd place.

3.5 Evaluation of the Soft-mask Module

Since we replace the simple linear output layer in FlowNet(S/C) with a more complex soft-mask module, we would like to verify whether the improved results are obtained due to the way the soft-mask module works and not simply due to having a model with more coefficients. To better investigate the operation of the soft-mask module, we compared the FlowNetC+LOFE with three other networks in which we slightly changed the structures of the soft-mask module.

In the first network, given the proposed structure as FlowNetC+LOFE, we removed the maxout opera-

Method	Flying Chairs	Sintel Clean		Sintel Final		KITTI	
		Train	Test	Train	Test	Train	Test
DSTFlow	5.11	6.93	10.40	7.82	11.11	**10.43**	-
USCNN	-	-	-	8.88	-	-	-
BB-FlowNet	5.36	-	-	-	-	11.32	**9.93**
FlowNetS+LOFE	**4.81**	**6.56**	10.10	**7.62**	10.98	10.78	10.82
FlowNetC+LOFE	4.92	6.78	**9.98**	7.77	**10.19**	11.01	11.25

Table 2: EPE errors of methods that are trained without supervision. The results of compared methods are taken directly from the corresponding paper. The notation 'ft' means fine-tuning.

	Chairs	**Sintel**
FNetC+schd	2.35	4.24
FNetC+LOFE/no-maxout+schd	2.29	4.12
FNetC+LOFE/normalize+schd	2.32	4.03
FNetC+LOFE/no-masks+schd	2.62	4.35
FNetC+LOFE+schd	**2.17**	**3.96**

Table 3: Comparison of the proposed FlowNetC+LOFE and its three variants. The notation 'schd' represents that networks are trained using data scheduling.

tion from the soft-mask module and kept the remaining configuration the same. We denote the resulting work FlowNetC+LOFE/no-maxout. In this case, FlowNetC+LOFE/no-maxout will have the exact same number of coefficients as FlowNetC+LOFE. In [Zhou *et al.*, 2016][Flynn *et al.*, 2016], intermediate generated masks are also combined with extracted image features. Instead of adopting a max-out operation, masks are normalized in their works. Therefore, for the second network, we first copied the structure of FlowNetC+LOFE/no-maxout and employed the same normalization in the second network. We denote the resulting network as FlowNetC+LOFE/normalize. For the third network, we removed the mask branch from the soft-mask module and left the intermediate optical flow only. The third network is denoted as FlowNetC+LOFE/no-masks.

For all four networks, we used $k = 10$ in the soft-mask module. We used FlowNetC trained by data scheduling without fine-tuning as a baseline in the evaluation. To obtain an unbiased evaluation result, we trained and tested each of these networks on both Flying Chairs and Sintel dataset [Butler *et al.*, 2012] three times. The average EPE is reported in Table 3.

As we can see from Table 3, the proposed FlowNetC+LOFE performed better than its three variants. This comparison leads to three conclusions. First, the better performance obtained by adding the soft-mask module to FlowNetC is not because using a larger model. Since both no-maxout and normalize versions of the proposed network have the identical complexity to the proposed network. Thus we conclude that the maxout operation makes optical flow estimation a more manageable task by separating optical flows into multiple layers. Second, with the same structure, FlowNetC+LOFE is better than no-maxout and normalize versions of the model. This result is caused by the maxout operation in the proposed network which can separate flows to layers to better generate flows in local regions. Third, the performance of FlowNetC/no-maxout and FlowNetC/normalize are both better than FlowNetC/no-masks version. While a possible hypothesis is that the model of no-maxout is larger than the model of no-masks. However, since FlowNetC, the smallest model in this comparison, achieved a better performance compared with the no-masks model. We thus conclude that the actual reason is a fact that both the FlowNetC/no-maxout and the FlowNetC/normalize models are using a quadratic function to fit optical flow instead of the linear function which is used in the no-masks FlowNetC model.

k value	5	10	20	30	40
EPE	2.192	2.173	2.176	2.237	2.249

Table 4: EPE as a function of k, the number of masks generated intermediately.

We investigate the relationship between k the number of masks and flow layers used in the soft-mask module, and network performance in terms of EPE. Experiments were done using the Flying Chairs dataset. We set $k = 5\mathrm{x}$, where $\mathrm{x} = 1, \ldots, 8$. As can be observed in Table 4, there is an immediate benefit to using the soft-mask module with respect to FlowNetC, where $k = 5$ will efficiently boost performance. We see a convergence of EPE after $k = 10$ and a slightly increase when $k > 20$. This may be due to slight overfitting when separating the optical flow to too many layers.

4 Conclusion

We describe a new approach for optical flow estimation by combining a traditional layered flow representation with a deep learning method. Rather than pre-segmenting images to layers, the proposed approach automatically learns a layered representation of optical flow using the proposed soft-mask module. The soft-mask module has the advantage of splitting flow to layers in which the computation of the flow is quadratic in terms of input features. For evaluation, we use FlowNet as our base net to add the soft-mask module. The resulting networks are tested on three well-known benchmarks with both supervised and unsupervised flow estimation tasks. Experimental results show that the proposed network achieves better results with respect to the original FlowNet.

References

[Ahmadi and Patras, 2016] Aria Ahmadi and Ioannis Patras. Unsupervised convolutional neural networks for motion

estimation. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1629–1633. IEEE, 2016.

[Butler *et al.*, 2012] 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, October 2012.

[Darrell and Pentland, 1991] Trevor Darrell and Alexander Pentland. Robust estimation of a multi-layered motion representation. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 173–178. IEEE, 1991.

[Dosovitskiy *et al.*, 2015] A. Dosovitskiy, P. Fischery, 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.

[Flynn *et al.*, 2016] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[Goodfellow *et al.*, 2013] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1319–III–1327. JMLR.org, 2013.

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

[Hur and Roth, 2016] Junhwa Hur and Stefan Roth. *Joint Optical Flow and Temporally Consistent Semantic Segmentation*, pages 163–177. Springer International Publishing, Cham, 2016.

[Ilg *et al.*, 2017] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Jepson and Black, 1993] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, Jun 1993.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[Ranjan and Black, 2017] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Ren *et al.*, 2017] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Artificial Intelligence (AAAI-17), Proceedings of the Thirty-First AAAI Conference on*, pages 1495–1501, 2017.

[Revaud *et al.*, 2015] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[Rohlfing *et al.*, 2003] Torsten Rohlfing, Calvin R Maurer, David A Bluemke, and Michael A Jacobs. Volume-preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint. *IEEE transactions on medical imaging*, 22(6):730–741, 2003.

[Sevilla-Lara *et al.*, 2016] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. Optical flow with semantic segmentation and localized layers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Sun *et al.*, 2010] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2010.

[Weinzaepfel *et al.*, 2013] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[Yang and Li, 2015] Jiaolong Yang and Hongdong Li. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1027, 2015.

[Yu *et al.*, 2016] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *CoRR*, abs/1608.05842, 2016.

[Zhou *et al.*, 2016] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.