Deep Optical Flow Estimation Via Multi-Scale Correspondence Structure Learning

Shanshan Zhao[1], Xi Li[1,2,*], Omar El Farouk Bourahla[1]

[1] Zhejiang University, Hangzhou, China

[2] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China

{zsszju,xilizju,obourahla}@zju.edu.cn

Abstract

As an important and challenging problem in computer vision, learning based optical flow estimation aims to discover the intrinsic correspondence structure between two adjacent video frames through statistical learning. Therefore, a key issue to solve in this area is how to effectively model the multi-scale correspondence structure properties in an adaptive end-to-end learning fashion. Motivated by this observation, we propose an end-to-end multi-scale correspondence structure learning (MSCSL) approach for optical flow estimation. In principle, the proposed MSCSL approach is capable of effectively capturing the multi-scale inter-image-correlation correspondence structures within a multi-level feature space from deep learning. Moreover, the proposed MSCSL approach builds a spatial Conv-GRU neural network model to adaptively model the intrinsic dependency relationships among these multi-scale correspondence structures. Finally, the above procedures for correspondence structure learning and multi-scale dependency modeling are implemented in a unified end-to-end deep learning framework. Experimental results on several benchmark datasets demonstrate the effectiveness of the proposed approach.

1 Introduction

Optical flow estimation seeks for perceiving the motion information across consecutive video frames, and has a wide range of vision applications such as human action recognition and abnormal event detection. Despite the significant progress in the literature, optical flow estimation is still confronted with a number of difficulties in discriminative feature representation, correspondence structure modeling, computational flexibility, etc. In this paper, we focus on how to set up an effective learning pipeline that is capable of performing multi-scale correspondence structure modeling with discriminative feature representation in a flexible end-to-end deep learning framework.

---

*Corresponding author

Due to the effectiveness in statistical modeling, learning based approaches emerge as an effective tool of optical flow estimation [Dosovitskiy *et al.*, 2015; Jason *et al.*, 2016; Ahmadi and Patras, 2016; Zhu *et al.*, 2017]. Usually, these approaches either just take image matching at a single scale into account, or take a divide-and-conquer strategy that copes with image matching at multiple scales layer by layer. Under the circumstances of complicated situations (e.g., large inter-image displacement or complex motion), they are often incapable of effectively capturing the interaction or dependency relationships among the multi-scale inter-image correspondence structures, which play an important role in robust optical flow estimation. Furthermore, their matching strategies are often carried out in the following two aspects. 1) Set a fixed range of correspondence at a single scale in the learning process [Dosovitskiy *et al.*, 2015; Jason *et al.*, 2016; Zhu *et al.*, 2017]; and 2) update the matching range dynamically with a coarse-to-fine scheme [Ahmadi and Patras, 2016; Ranjan and Black, 2016]. In practice, since videos have time-varying dynamic properties, selecting an appropriate fixed range for matching is difficult for adapting to various complicated situations. Besides, the coarse-to-fine scheme may cause matching error propagations or accumulations from coarse scales to fine scales. Therefore, for the sake of robust optical flow estimation, correspondence structure modeling ought to be performed in an adaptive multi-scale collaborative way. Moreover, it is crucial to effectively capture the cross-scale dependency information while preserving spatial self-correlations for each individual scale in a totally data-driven fashion.

Motivated by the above observations, we propose a novel unified end-to-end optical flow estimation approach called **Multi-Scale Correspondence Structure Learning (MSCSL)** (as shown in Fig. 1), which jointly models the dependency of multi-scale correspondence structures by a Spatial Conv-GRU neural network model based on multi-level deep learning features. To summarize, the contributions of this work are twofold:

- We propose a multi-scale correspondence structure learning approach, which captures the multi-scale inter-image-correlation correspondence structures based on the multi-level deep learning features. As a result, the task of optical flow estimation is accomplished by jointly learning the inter-image correspondence structures

at multiple scales within an end-to-end deep learning framework. Such a multi-scale correspondence structure learning approach is innovative in optical flow estimation to the best of our knowledge.

- We design a Spatial Conv-GRU neural network model to model the cross-scale dependency relationships among the multi-scale correspondence structures while preserving spatial self-correlations for each individual scale in a totally data-driven manner. As a result, adaptive multi-scale matching information fusion is enabled to make optical flow estimation adapt to various complicated situations, resulting in robust estimation results.

2 Our Approach

2.1 Problem Formulation

Let $\{(\mathbf{X}_i, \mathbf{O}_i)\}_{i=1}^N$ be a set of N training samples, where $\mathbf{X}_i = (\mathbf{X}_i^A \in \mathbb{R}^{H \times W \times 3}, \mathbf{X}_i^B \in \mathbb{R}^{H \times W \times 3})$ and $\mathbf{O}_i \in \mathbb{R}^{H \times W \times 2}$ represent an RGB image pair and the corresponding optical flow respectively. In this paper, our objective is to learn a model $f(\mathbf{X}_i; \theta)$ parameterized by θ to predict the dense motion of the first image $\mathbf{X}_i^A$. For the sake of expression, we ignore the left subscript i in the remaining parts.

In this paper, we focus on two factors, (1) computing the correlation maps between image representations at different scales and adaptively setting up the correspondence structure in a data-driven way, (2) encoding the correspondence maps into high-level feature representation for regressing the optical flow.

2.2 Multi-Scale Correspondence Structure Modelling

Multi-Scale Image Representations. To represent the input image at multiple scales, we firstly employ convolution neural networks (CNNs) to extract the deep features at a single scale parameterized by θ_1 to represent the image $\mathbf{I}$, as illustrated in Fig. 1:

$$\mathbf{Y} = f_{CNN1}(\mathbf{I}; \theta_1) \tag{1}$$

and then model the multi-level feature representations parameterized by $\{\theta_{2,l}\}$ with $\mathbf{Y}$ as the input, as depicted in Fig. 1:

$$\mathbf{F}^l = f_{CNN2}(\mathbf{Y}; \theta_{2,l}, l = 1, 2, \dots) \tag{2}$$

where $\mathbf{F}^l$ represents the l-th level, and the size of $\mathbf{F}^{l+1}$ is larger than that of $\mathbf{F}^l$. From top to bottom (or coarse to fine), the feature representations at small scales[1] tend to learn the sematic components, which contribute to find the correspondence of semantic parts with large displacements; Furthermore, the large scale feature maps tend to learn the local representation, which can distinguish the patches with small displacements. In this paper, we use $\{\mathbf{F}^{A,l}\}$ and $\{\mathbf{F}^{B,l}\}$ to denote the multi-scale representations of $\mathbf{X}^A$ and $\mathbf{X}^B$ respectively.

Correspondence Structure Modelling. Given an image pair $(\mathbf{X}^A, \mathbf{X}^B)$ from a video sequence, we firstly extract their

[1]In this paper, the small scale means small size; the large scale means large size

multi-level feature representations $\{\mathbf{F}^{A,l}\}$ and $\{\mathbf{F}^{B,l}\}$ using Eq. 1 and Eq. 2. In order to learn the correspondence structures between the image pair, we calculate the similarity between the corresponding feature representations instead. Firstly, we discuss the correlation computation proposed in [Dosovitskiy *et al.*, 2015]:

$$Corr(\mathbf{F}_{i,j}^A, \mathbf{F}_{S(i,j;d)}^B) =$$

$$Concat\{ \sum_{o_x=-k}^{k} \sum_{o_y=-k}^{k} \langle \mathbf{F}_{i+o_x, j+o_y}^A, \mathbf{F}_{p+o_x, q+o_y}^B \rangle,$$

$$(p, q) \in [i - d, i + d] \times [j - d, j + d]\} \tag{3}$$

where $\mathbf{F}_{i,j}^A$ and $\mathbf{F}_{i,j}^B$ denote the feature vector at the (i, j)-th location of $\mathbf{F}^A$ and $\mathbf{F}^B$ respectively, and $Concat\{\cdot\}$ denotes concatenating the elements in the set $\{\cdot\}$ to a vector, $S(i, j; d)$ denotes the $(2d + 1) \times (2d + 1)$ neighborhood of location (i, j). The meaning of Eq. 3 is that given a maximum displacement d, the correlations between the location (i, j) in $\mathbf{F}^A$ and $S(i, j; d)$ in $\mathbf{F}^B$ can be obtained by computing the similarities between the square patch of size $(2k + 1) \times (2k + 1)$ centered at location (i, j) in $\mathbf{F}^A$ and square patches of the same size centered at all locations of S in $\mathbf{F}^B$.

To model the correspondence between the (i, j)-th location in $\mathbf{F}^A$ and its corresponding location $(\hat{i}, \hat{j})$ in $\mathbf{F}^B$, we can (1) calculate $Corr(\mathbf{F}_{i,j}^A, \mathbf{F}_{S(i,j;d)}^B)$ in a small neighbourhood S of the (i, j)-th location in $\mathbf{F}^B$, or (2) calculate $Corr(\mathbf{F}_{i,j}^A, \mathbf{F}_{S(i,j;d)}^B)$ in a large enough neighbourhood S of the (i, j)-th location in $\mathbf{F}^B$, or even in the whole feature map $\mathbf{F}^B$. But the former can not guarantee the computation of similarity between the (i, j)-th location and the corresponding $(\hat{i}, \hat{j})$-th location, while the latter leads to low computational efficiency, because the complexity $\mathcal{O}(d^2 k^2)$ of Eq. 3 exhibits quadratic growth when the value of d increases. To address that problem, we adopt correlation computation at each scale of multi-scale feature representations $\{\mathbf{F}^{A,l}\}$ and $\{\mathbf{F}^{B,l}\}$:

$$\mathbf{M}_{i,j}^l = Corr(\mathbf{F}_{i,j}^{A,l}, \mathbf{F}_{S(i,j;d_l)}^{B,l}) \tag{4}$$

where the maximum displacement d_l varies from bottom to top.

In order to give the network more flexibility in how to deal with the correspondence maps, we add three convolutional layers to the outputs of the $Correlation$ operation, which is the same as that proposed in [Dosovitskiy *et al.*, 2015], to extract the high-level representations parameterized by $\{\theta_{3,l}\}$, as described in Fig. 1:

$$\mathbf{Q}^l = f_{CNN3}(\mathbf{M}^l; \theta_{3,l}, l = 1, 2, \dots) \tag{5}$$

2.3 Correspondence Maps Encoding Using Spatial Conv-GRU

Cross-Scale Dependency Relationships Modelling. For the sake of combining the correlation representations $\{\mathbf{Q}^l\}$ and preserving the spatial structure to estimate dense optical flow, we consider the representations as a feature map sequence,
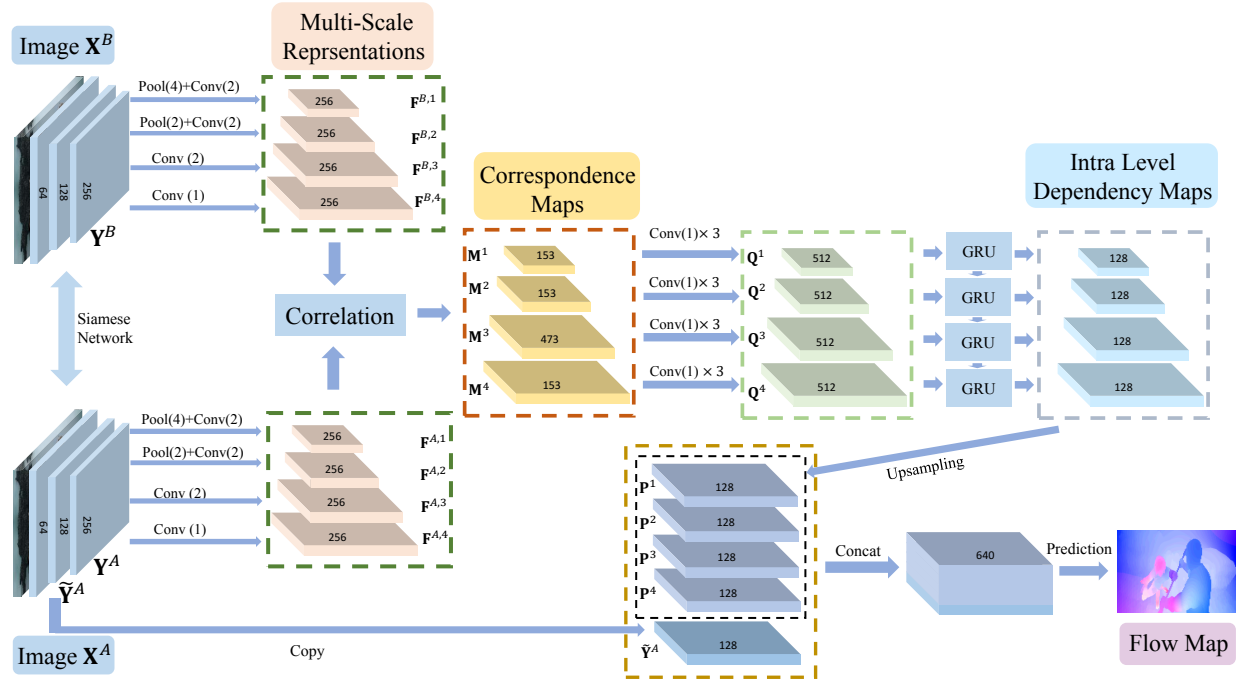
Figure 1: The proposed CNN framework of Multi-Scale Correspondence Structure Learning (MSCSL). The n in Pool(n) and Conv(n) denotes the $stride$ of corresponding operation, and $\times 3$ denotes three consecutive operations. The network consists of three parts: (1) Multi-Scale Correspondence Structure Modelling, this part uses a Siamese Network to extract robust multi-level deep features for the two images, and then constructs the correspondence structures between the feature maps at different scales, (2) Correspondence Maps Encoding, this part employs a Spatial Conv-GRU presented in this work to encode the correspondence maps at different scales, (3) Prediction, we use the encoded feature representation to predict the optical flow map.

and then apply Convolutional Gated-Recurrent-Unit Recurrent Networks(Conv-GRUs) to model the cross-scale dependency relationships among the multi-scale correspondence structures. Conv-GRUs have been used to model the temporal dependencies between frames of the video sequence [Ballas *et al.*, 2015; Siam *et al.*, 2016]. A key advantage of Conv-GRUs is that they can not only model the dependencies among a sequence, but also preserve the spatial location of each feature vector. One of significant differences between a Conv-GRU and a traditional GRU is that innerproduct operations are replaced by convolution operations.

However, because of the employed scheme similar to coarse-to-fine, the size of the $(t + 1)$-th input in the sequence is larger than that of the t-th input. We cannot apply the standard Conv-GRU on our problem, so instead we propose a Spatial Conv-GRU in which each layer's output is upsampled as the input of the next layer. For the input sequence $\{\mathbf{Q}^l\}$, the formulation of the Spatial Conv-GRU is:

$$\mathbf{Z}^l = \sigma(\mathbf{W}_z * \mathbf{Q}^l + \mathbf{U}_z * \mathbf{H}^{l-1,\uparrow}) \quad (6)$$

$$\mathbf{R}^l = \sigma(\mathbf{W}_r * \mathbf{Q}^l + \mathbf{U}_r * \mathbf{H}^{l-1,\uparrow}) \quad (7)$$

$$\widetilde{\mathbf{H}}^l = tanh(\mathbf{W} * \mathbf{Q}^l + \mathbf{U} * (\mathbf{R}^l \odot \mathbf{H}^{l-1,\uparrow})) \quad (8)$$

$$\mathbf{H}^l = (\mathbf{1} - \mathbf{Z}^l) \odot \mathbf{H}^{l-1,\uparrow} + \mathbf{Z}^l \odot \widetilde{\mathbf{H}}^l \quad (9)$$

$$\mathbf{H}^{l,\uparrow} = \mathbf{W}^\uparrow \circledast \mathbf{H}^l \quad (10)$$

where $*$ and $\odot$ denote a convolution operation and an element-wise multiplication respectively, and σ is an activa-

tion function, e.g., $sigmoid$., $\circledast$ denotes the transposed convolution. The Spatial Conv-GRU can model the transition from coarse to fine and recover the spatial topology, outputting intra-level dependency maps $\{\mathbf{H}^l\}$.

Intra-Level Dependency Maps Combination. After getting the hidden outputs $\{\mathbf{H}^l\}$, we upsample them to the same size, written as $\mathbf{P}^l$:

$$\mathbf{P}^l = f_{CNN4}(\mathbf{Q}^l; \theta_4) \quad (11)$$

where $\theta_4 := \{\mathbf{W}_z, \mathbf{U}_z, \mathbf{W}_r, \mathbf{U}_r, \mathbf{W}, \mathbf{U}, \mathbf{W}^\uparrow\}$ are the parameters needed to be optimized. Furthermore, we concatenate the hidden outputs $\{\mathbf{P}^l\}$ with the 2nd convolutional output $\widetilde{\mathbf{Y}}^A$ of $\mathbf{X}^A$ to get the final encoded feature representation for optical flow estimation, as depicted in Fig. 1:

$$\mathbf{E} = Concat\{\widetilde{\mathbf{Y}}^A, \mathbf{P}^l, l = 1, 2, \dots\} \quad (12)$$

where $Concat$ represents the concatenation operation.

Finally, the proposed framework learns a function parameterized by θ_5 to predict the optical flow:

$$\begin{aligned}
\hat{\mathbf{O}} &= f_{CNN5}(\mathbf{E}; \theta_5) \\
&= f(\mathbf{X}^A, \mathbf{X}^B; \theta_1, \theta_{2,l}, \theta_{3,l}, \theta_4, \theta_5, l = 1, 2, \dots)
\end{aligned} \quad (13)$$

2.4 Unified End-to-End Optimization

As the image representation, correspondence structure learning and correspondence maps encoding are highly related, we construct a unified end-to-end framework to optimize the

Table 1: Comparison of average endpoint errors (EPE) to the state-of-the-art. The times with right superscript $*$ indicate that the methods run on CPU, while the rest run on GPU. The numbers in parentheses are the results of the networks on dataset they were fine-tuned on. And the methods with +ft represent that the models were fine-tuned on MPI Sintel training dataset (two versions together) after trained on Flying Chairs training dataset.

Methods	Sintel clean		Sintel final		KITTI 2012		Middlebury	Flying Chairs	Time (sec)
	train	test	train	test	train	test	train	test	
EpicFlow	2.40	4.12	3.70	6.29	3.47	3.80	0.31	2.94	16^*
DeepFlow	3.31	5.38	4.56	7.21	4.58	5.80	0.21	3.53	17^*
FlowFields	1.86	3.75	3.06	5.81	3.33	3.50	8.33	0.27	22^*
EPPM	−	6.49	−	8.38	−	9.20	−	−	0.2
DenseFlow	−	4.39	−	7.42	−	2.90	−	−	265^*
LDOF	4.64	7.56	5.96	9.12	10.94	12.40	0.44	3.47	65^*
FlowNetS	4.50	7.42	5.45	**8.43**	8.26	−	1.09	2.71	0.08
FlowNetC	4.31	7.28	5.87	8.81	9.35	−	1.15	2.19	0.15
SPyNet	4.12	**6.69**	5.57	**8.43**	9.12	−	**0.33**	2.63	0.07
MSCSL/wosr	3.63	−	4.93	−	5.98	−	0.87	2.14	**0.05**
MSCSL/wor	**3.37**	−	4.72	−	**5.80**	−	0.92	2.11	0.06
MSCSL	3.39	−	**4.70**	−	5.87	−	0.90	**2.08**	0.06
FlowNetS+ft	(3.66)	6.97	(4.44)	7.76	7.52	9.10	0.98	3.04	0.08
FlowNetC+ft	(3.78)	6.85	(5.28)	8.51	8.79	−	0.93	2.27	0.15
SPyNet+ft	(3.17)	6.64	(4.32)	8.36	8.25	10.10	**0.33**	3.07	0.07
MSCSL/wosr+ft	(3.18)	**5.68**	(4.21)	7.49	5.89	6.90	0.81	2.51	**0.05**
MSCSL/wor+ft	(**3.07**)	5.79	(4.16)	**7.42**	5.87	**6.80**	0.87	2.28	0.06
MSCSL+ft	(**3.07**)	5.78	(**4.15**)	**7.42**	**5.77**	7.10	0.86	**2.25**	0.06

three parts jointly. The loss function used in the optimization framework consists of two parts, namely, a supervised loss and an unsupervised loss (or reconstruction loss). The former is the endpoint error (EPE), which measures the Euclidean distance between the predicted flow $\hat{\mathbf{O}}$ and the ground truth $\mathbf{O}$, while the latter is based on the brightness constancy assumption, which measures the Euclidean distance between the first image $\mathbf{X}^A$ and the warped second image $\mathbf{X}^{B_{warp}}$.

$$\mathcal{L}(\mathbf{O}, \hat{\mathbf{O}}; \mathbf{X}^A, \mathbf{X}^B) = \mathcal{L}_s(\mathbf{O}, \hat{\mathbf{O}}) + \lambda \mathcal{L}_{us}(\hat{\mathbf{O}}; \mathbf{X}^A, \mathbf{X}^B) \tag{14}$$

$$\mathcal{L}_s(\mathbf{O}, \hat{\mathbf{O}}) = \sum_{i,j} \sqrt{(\mathbf{O}_{i,j}^u - \hat{\mathbf{O}}_{i,j}^u)^2 + (\mathbf{O}_{i,j}^v - \hat{\mathbf{O}}_{i,j}^v)^2} \tag{15}$$

$$\mathcal{L}_{us}(\hat{\mathbf{O}}; \mathbf{X}^A, \mathbf{X}^B) = \sum_{i,j} \sqrt{(\mathbf{X}_{i,j}^A - \mathbf{X}_{i,j}^{B_{warp}})^2} \tag{16}$$

where $\hat{\mathbf{O}}^u$ and $\hat{\mathbf{O}}^v$ denote the displacement in horizontal and vertical respectively, and λ is the balance parameter. $\mathbf{X}^{B_{warp}}$ can be calculated via bilinear sampling according to $\hat{\mathbf{O}}$, as proposed in Spatial Transform Networks[Jaderberg *et al.*, 2015]:

$$\mathbf{X}_{i,j}^{B_{warp}} = \sum_n^H \sum_m^W \mathbf{X}_{m,n}^B max(0, 1 - \|i + \hat{\mathbf{O}}_{i,j}^u - m\|)$$
$$max(0, 1 - \|j + \hat{\mathbf{O}}_{i,j}^v - n\|) \tag{17}$$

Because the raw data $\mathbf{X}^A$ and $\mathbf{X}^B$ contain noise and illumination changes and are less discriminative, in some cases the brightness constancy assumption is not satisfied; Furthermore, in highly saturated or very dark regions, the assumption also suffers difficulties [Jason *et al.*, 2016]. Therefore, applying Eq. 16 on the raw data directly will make the network more difficult when training. To address that issue, we apply the brightness constancy assumption on the 2nd convolutional outputs $\widetilde{\mathbf{Y}}^A$ and $\widetilde{\mathbf{Y}}^B$ of $\mathbf{X}^A$ and $\mathbf{X}^B$ instead of $\mathbf{X}^A$ and $\mathbf{X}^B$. The training and test stages are shown in Alg. 1.
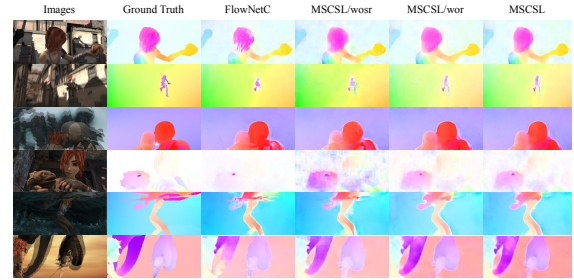

Figure 2: Examples of optical flow estimation using FlowNetC, M-SCSL/wosr, MSCSL/wor and MSCSL on the MPI Sintel dataset (Clean version). Note that our proposed methods perform well in both small displacement and large displacement.

3 Experiments

3.1 Datasets

Flying Chairs [Dosovitskiy *et al.*, 2015] is a synthetic dataset created by applying affine transformations to a real image dataset and a rendered set of 3D chair models. This dataset contains $22,872$ image pairs, and is split into $22,232$ training and 640 test pairs.

MPI Sintel [Butler *et al.*, 2012] is created from an animated movie and contains many large displacements and provides dense ground truth. It consists of two versions: the Final version and the Clean version. The former contains motion blurs and atmospheric effects, while the latter does not include these effects. There are $1,041$ training image pairs for each version.

KITTI 2012 [Geiger *et al.*, 2012] is created from real world scenes by using a camera and a 3D laser scanner. It consists of 194 training image pairs with sparse optical flow ground truth.

Middlebury [Baker *et al.*, 2011] is a very small dataset, containing only 8 image pairs for training. And the displacements are typically limited to 10 pixels.

Table 2: Comparison of FlowNet, SPyNet and our proposed methods on MPI Sintel test datasets for different velocities (s_*) and displacement (d_*).

Methods	Sintel Final					
	d_{0-10}	d_{10-60}	d_{60-140}	s_{0-10}	s_{10-40}	s_{40+}
FlowNetS+ft	7.25	4.61	2.99	1.87	5.83	43.24
FlowNetC+ft	7.19	4.62	3.30	2.30	6.17	**40.78**
SPyNet+ft	6.69	4.37	3.29	**1.39**	5.53	49.71
MSCSL/wosr+ft	6.27	3.77	2.96	1.96	4.97	40.98
MSCSL/wor+ft	6.08	**3.57**	**2.79**	1.76	**4.81**	41.74
MSCSL+ft	**6.06**	3.58	2.81	1.73	4.83	41.87
Methods	Sintel Clean					
	d_{0-10}	d_{10-60}	d_{60-140}	s_{0-10}	s_{10-40}	s_{40+}
FlowNetS+ft	5.99	3.56	2.19	1.42	3.81	40.10
FlowNetC+ft	5.57	3.18	1.99	1.62	3.97	**33.37**
SPyNet+ft	5.50	3.12	1.71	**0.83**	3.34	43.44
MSCSL/wosr+ft	4.84	2.39	1.64	1.27	3.26	33.40
MSCSL/wor+ft	4.80	2.34	1.61	1.26	**3.07**	34.90
MSCSL+ft	**4.79**	**2.33**	**1.58**	1.24	3.08	34.83

3.2 Implementation Details

Network Architecture

In this part, we introduce the network architecture briefly. We use 7×7 convolutional kernel for the first convolutional layer and 5×5 for the second and third convolutional layers. Then we use max-pooling and convolutional operations to obtain multi-scale representations, as illustrated in Fig. 1. The correlation layer is the same as that proposed in [Dosovitskiy *et al.*, 2015], and the d_l are set to 5, 5, 10, 10 from top to bottom (or from coarse to fine). And then we employ 3×3 kernel and 4×4 kernel for the other convolutional layers and deconvolutional layers respectively.

Data Augmentation

To avoid overfitting and improve the generalization of network, we employ the data augmentation strategy for the training by performing random online transformations, including scaling, rotation, translation, as well as additive Gaussian noise, contrast, multiplicative color changes to the RGB channels per image, gamma and additive brightness.

Training Details

We implement our architecture using Caffe [Jia *et al.*, 2014] and use an NVIDIA TITAN X GPU to train the network. To verify our proposed framework, we conduct three comparison experiments, (1) MSCSL/wosr, this experiment does not contain both the proposed Spatial Conv-GRU and reconstruction loss, and use the refinement network proposed in [Dosovitskiy *et al.*, 2015] to predict dense optical flow, (2) MSCSL/wor, this experiment employs the Spatial Conv-GRU, which can be implemented by unfolding the recurrent model in the prototxt file, to encode the correspondence maps for dense optical flow estimation and demonstrates the effectiveness of the Spatial Conv-GRU in comparison to MSCL/wosr, (3) MSCSL, this experiment contains all parts (Spatial Conv-GRU and reconstruction loss) aforementioned.

In the MSCSL/wosr and MSCSL/wor, we train the networks on Flying Chairs training dataset using Adam optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. To tackle the gradients explosion, we adopt the same strategy as proposed in [Dosovitskiy *et al.*, 2015]. Specifically, we firstly use a learning rate of $1e-6$ for the first 10k iterations with a batch size of 8 pairs. After that, we increase the learning rate to $1e-4$ for the following 300k iterations, and then divide it by

2 every 100k iterations. We terminate the training after 600k iterations (about 116 hours).

In the MSCSL, we firstly train the MSCSL/wor for 500k iterations using the training strategy above. After that, we add the reconstruction loss with the balance parameter $\lambda = 0.005$. And then we fine-tune the network for 100k iterations with a fixed learning of $1.25e-5$.

After training the three networks on Flying Chairs training dataset respectively, we fine-tune the networks on the MPI Sintel training dataset for tens of thousands of iterations with a fixed learning rate of $1e-6$ until the networks converge. Specifically, we fine-tune the networks on the Clean version and Final version together with 1,816 for training and 266 for validation. Since the KITTI 2012 dataset and Middlebury dataset are small and only contain sparse ground truth, we do not conduct fine-tuning on these two datasets.

3.3 Comparison to State-of-the-Art

In this section, we compare our proposed methods to recent state-of-the-art approaches, including traditional methods, such as EpicFlow [Revaud *et al.*, 2015], DeepFlow [Weinzaepfel *et al.*, 2013], FlowFields [Bailer *et al.*, 2015], EPPM [Bao *et al.*, 2014], LDOF [Brox and Malik, 2011], DenseFlow [Yang and Li, 2015], and deep learning based methods, such as FlowNetS [Dosovitskiy *et al.*, 2015], FlowNetC [Dosovitskiy *et al.*, 2015], SPyNet [Ranjan and Black, 2016]. Table 1 shows the performance comparison between our proposed methods and the state-of-the-art using average endpoint errors (EPE). We mainly focus on the deep learning based methods, so we only compare our proposed methods with the learning-based frameworks such as FlowNet and SpyNet.

Flying Chairs. For all three comparison experiments, We train our networks on this dataset firstly, and employ MPI Sintel dataset to fine-tune them further. Table 1 shows that MSCSL outperforms the other comparison experiments, MSCSL/wosr and MSCSL/wor. Furthermore, our proposed methods achieve better performance comparable with the state-of-the-art methods. After fine-tuning, in most cases most learning based methods suffer from performance decay, this is mostly because of the disparity between Flying Chairs and MPI Sintel dataset. Some visual estimation results on this dataset are shown in Fig. 3.

MPI Sintel. After the training on Flying Chairs firstly, we fine-tune the trained models on this dataset. The models

Algorithm 1: Deep Optical Flow Estimation Via MSCSL

Input: A set of N training samples $\{((\mathbf{X}_i^A, \mathbf{X}_i^B), \mathbf{O}_i)\}_{i=1}^N$
Output: The deep model parameterized by θ: $f(\mathbf{X}^A, \mathbf{X}^B; \theta)$

```
  /* The training stage                    */
1 repeat
    /* For the K batches, do              */
2   for k = 1, ..., K do
      /* Process the k-th training
         mini-batches Bk                  */
3     for n ∈ Bk do
        /* Process the n-th image pair in
           Bk                             */
```

4   Extract the image representation $\mathbf{Y}_n^A$ and $\mathbf{Y}_n^B$ using Eq. 1;

5   Model the multi-scale feature representation $\{\mathbf{F}_n^{A,l}\}$ and $\{\mathbf{F}_n^{B,l}\}$ using Eq. 2;

6   Compute the correlation between feature representations $\{\mathbf{M}_n^l\}$ using Eq. 3 and Eq. 4;

7   Extract the high-level representations $\{\mathbf{Q}_n^l\}$ of $\{\mathbf{M}_n^l\}$ using Eq. 5;

8   Encode the correspondence representations $\{\mathbf{Q}_n^l\}$ to get $\{\mathbf{P}_n^l\}$ using Eq. 6;

9   Concatenate $\{\mathbf{P}_n^l\}$ with the 2nd convolutional outputs of $\mathbf{X}_n^A$ to obtain $\mathbf{E}_n$ using Eq. 12;

10   Regress the optical flow estimation $\hat{\mathbf{O}}_n$ using Eq. 13;

11   Minimize the objective function Eq. 14;

12   **end**

```
      /* Update network parameters        */
```

13   Update parameters $\theta = \{\theta_1, \theta_{2,l}, \theta_{3,l}, \theta_4, \theta_5, l = 1, 2, \dots\}$ using Adam;

14   **end**

15    $iter \leftarrow iter + 1$

16 **until** $iter < max\_iter$;

17 **return**;

```
  /* The test stage                        */
```

Input: Given an image pair $(\mathbf{X}^{A_t}, \mathbf{X}^{B_t})$ and the trained deep model $f(\mathbf{X}^A, \mathbf{X}^B; \theta)$
Output: The predicted optical flow $\hat{\mathbf{O}}^t = f(\mathbf{X}^{A_t}, \mathbf{X}^{B_t}; \theta_1, \theta_{2,l}, \theta_{3,l}, \theta_4, \theta_5, l = 1, 2, \dots)$
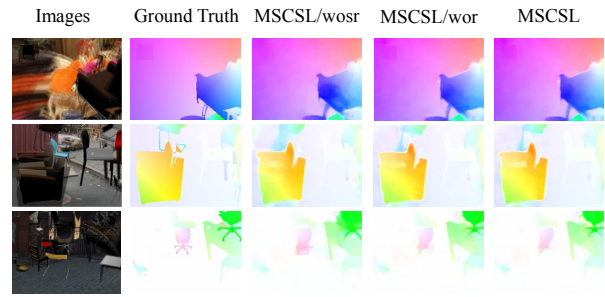
18 **return**;

Figure 3: Examples of optical flow prediction on the Flying Chairs dataset. Comparison to MSCSL/wosr, the results of MSCSL/wor and MSCSL are more smooth and finer.

KITTI 2012 test dataset and Middlebury training dataset respectively. Table 1 shows that our proposed methods outperform other deep learning based approaches remarkably on the KITTI 2012 dataset (including training set and test set). However, in most cases, on Middlebury training dataset, mainly containing small displacements, our proposed methods do not perform well, comparison to SPyNet.

Analysis. The results of our framework are more smooth and fine-grained. Specifically, our framework is capable of capturing the motion information of fine-grained object parts, as well as preserving edge information. Meanwhile, our Spatial Conv-GRU can suppress the noises in the results of model without it. All these insights can be observed in Fig. 3 and Fig. 2. However, our proposed frameworks are incapable of effectively capturing the correspondence structure and unstable in regions where the texture is uniform (e.g., on Middlebury dataset).

Timings. In Table 1, we show the per-frame runtimes of different approaches. Traditional methods are often implemented on a single CPU, while deep learning based methods tend to run on GPU. Therefore, we only compare the runtimes with FlowNetS, FlowNetC and SPyNet. The results in Table 1 demonstrate that our proposed methods (run on NVIDIA TITAN X GPU) improve the accuracy with a comparable speed against the state-of-the-art.

4 Conclusion

In this paper, we propose a novel end-to-end multi-scale correspondence structure learning based on deep learning for optical flow estimation. The proposed MSCSL learns the correspondence structure and models the multi-scale dependency in a unified end-to-end deep learning framework. Our model outperforms the state-of-the-art approaches based on deep learning by a considerable computing efficiency. The experimental results on several datasets demonstrate the effectiveness of our proposed framework.

Acknowledgements

trained on Flying Chairs are evaluated on the training dataset. The results shown in Table 1 demonstrate MSCSL's and MSCSL/sor's better ability to generalize than MSCSL/wosr's and other learning based approaches'. To further verify our proposed methods, we compare our methods with FlowNetS, FlownetC and SPyNet on MPI Sintel test dataset for different velocities and distances from motion boundaries, as described in Table 2. As shown in Table 1 and Table 2, our proposed methods perform better than other deep learning based methods. However, in the regions with velocities larger than 40 pixels (smaller than 10 pixels), the proposed methods are less accurate than FlowNetC (SpyNet). Some visual results are shown in Fig. 2.

KITTI 2012 and Middlebury. These two datasets are too small, so we do not fine-tune the models on these datasets. We evaluate the trained models on KITTI 2010 training dataset,

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.

[Bailer *et al.*, 2015] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4015–4023, 2015.

[Baker *et al.*, 2011] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[Ballas *et al.*, 2015] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

[Bao *et al.*, 2014] Linchao Bao, Qingxiong Yang, and Hailin Jin. Fast edge-preserving patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3534–3541, 2014.

[Brox and Malik, 2011] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[Butler *et al.*, 2012] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[Dosovitskiy *et al.*, 2015] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, et al. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766. IEEE, 2015.

[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.

[Jaderberg *et al.*, 2015] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[Jason *et al.*, 2016] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *Computer Vision–ECCV 2016 Workshops*, pages 3–10. Springer, 2016.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[Ranjan and Black, 2016] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. *arXiv preprint arXiv:1611.00850*, 2016.

[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.

[Siam *et al.*, 2016] Mennatullah Siam, Sepehr Valipour, Martin Jagersand, and Nilanjan Ray. Convolutional gated recurrent networks for video segmentation. *arXiv preprint arXiv:1611.05435*, 2016.

[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.

[Zhu *et al.*, 2017] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Guided optical flow learning. *arXiv preprint arXiv:1702.02295*, 2017.