Unsupervised Learning of Optical Flow With CNN-based Non-Local Filtering

Long Tian, *Member, IEEE,* Zhigang Tu, *Member, IEEE,* Dejun Zhang, *Member, IEEE,* Jun Liu, Baoxin Li, *Senior Member, IEEE,* and Junsong Yuan, *Senior Member, IEEE*

***Abstract*—Estimating optical flow from successive video frames is one of the fundamental problems in computer vision and image processing. In the era of deep learning, many methods have been proposed to use convolutional neural networks (CNNs) for optical flow estimation in an unsupervised manner. However, the performance of unsupervised optical flow approaches is still unsatisfactory and often lagging far behind their supervised counterparts, primarily due to over-smoothing across motion boundaries and occlusion. To address these issues, in this paper, we propose a novel method with a new post-processing term and an effective loss function to estimate optical flow in an unsupervised, end-to-end learning manner. Specifically, we first exploit a CNN-based non-local term to refine the estimated optical flow by removing noise and decreasing blur around motion boundaries. This is implemented via automatically learning weights of dependencies over a large spatial neighborhood. Because of its learning ability, the method is effective for various complicated image sequences. Secondly, to reduce the influence of occlusion, a symmetrical energy formulation is introduced to detect the occlusion map from refined bi-directional optical flows. Then the occlusion map is integrated to the loss function. Extensive experiments are conducted on challenging datasets, i.e. FlyingChairs, MPI-Sintel and KITTI to evaluate the performance of the proposed method. The state-of-the-art results demonstrate the effectiveness of our proposed method.**

***Index Terms*—Optical Flow, Unsupervised Learning, Non-local term, Loss function, Occlusion Map.**

I. Introduction

Optical flow estimation, which aims at computing a pixel-wise motion field between video frames, is one of the most fundamental problems in computer vision. It has attracted much attention in both the academic community and the industry due to its wide range of applications, e.g., action recognition [1], [2], segmentation [3], etc.

Recently, the optical flow estimation methods [4], [5], [6], [7] based on deep learning have overcome the typical problems (such as low speed performance) in traditional models [8], [9], by using convolutional neural networks (CNNs). However, one main drawback of these methods is that they are supervised and need large amount of annotated data with ground-truth optical flow for training. Unfortunately, obtaining per-pixel ground-truth optical flow for real videos is difficult [3], [10].

One promising research direction is to use unsupervised learning method. Ahmadi *et al*. [11] proposed to train a CNN to compute optical flow by designing a loss function based on the classical brightness constant assumption (BAC). However, occluded pixels will affect the estimation accuracy since the classical BAC-based loss function prefers to compensate the occluded regions with other arbitrary pixels. Recently, some more advanced unsupervised methods were presented to calculate optical flow [12], [13], [14], [15], [16] by taking into account occlusion. Still, there is a big performance gap between the unsupervised methods and their supervised counterparts (as will be shown in Section V). Mostly, these models are unable to handle outliers well and often produce over-smoothed flows.

Sun *et al*. [8] conducted massive experiments and found that the use of filters (such as mean filter, non-local filter, bilateral filter, etc.) can effectively improve the accuracy of traditional models. Among them, the non-local filter appears to be one of the best. The non-local term minimizes the L1 distance between the central value and all flow values in its neighborhood except itself, which can prevent smoothing across motion boundaries [17], [8]. However, the traditional non-local approach has two drawbacks: (1) the low speed performance, and (2) the weights are human-designed according to the measure of the Euclidean distance or else [8].

In this paper, we propose a new end-to-end unsupervised method for optical flow estimation. Firstly, to solve the problem of over-smoothing across motion boundaries, we design a CNN-based non-local term. Our non-local term is embedded in the deep neural network so that its weights can be automatically learned, making it is suitable for denoising various images and can preserve the details of motion boundaries. Besides, our CNN-based non-local term consists of just five convolutional layers and is only conducted at the last pyramid level, and hence saving significant computational time.

Secondly, to tackle problems due to occluded regions that usually produce misleading information [14], following pioneer works [15] [18], we enable our network to extract occlusion maps and integrate them to form a loss function. Specifically, we estimate the forward and backward optical flow to generate bi-directional occlusion maps (see Fig. 1). We use their relationship and allow them to leverage each other via the estimated bi-directional flow and occlusion maps.

Zhigang Tu and Long Tian contributed equally, Zhigang Tu and Long Tian are co-first authors. *Corresponding author: Zhigang Tu.*

Zhigang Tu is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 430079 Wuhan, China, E-Mail: (tuzhigang@whu.edu.cn).

Dejun Zhang is with the School of Information Engineering, China University of Geosciences, 30074 Wuhan, China, E-Mail: (zhangdejun@cug.edu.cn).

Jun Liu is with the Shenzhen Infinova Ltd. Company, 518100 Shenzhen.

Baoxin Li is with School of Computing, Informatics, Decision System Engineering, Arizona State University, AZ, 85287, USA, E-Mail: (Baoxin.Li@asu.edu).

Junsong Yuan is with the Computer Science and Engineering department, State University of New York at Buffalo, NY, 14260-2500, USA, E-Mail: (jsyuan@buffalo.edu).

Manuscript received May 17, 2019.

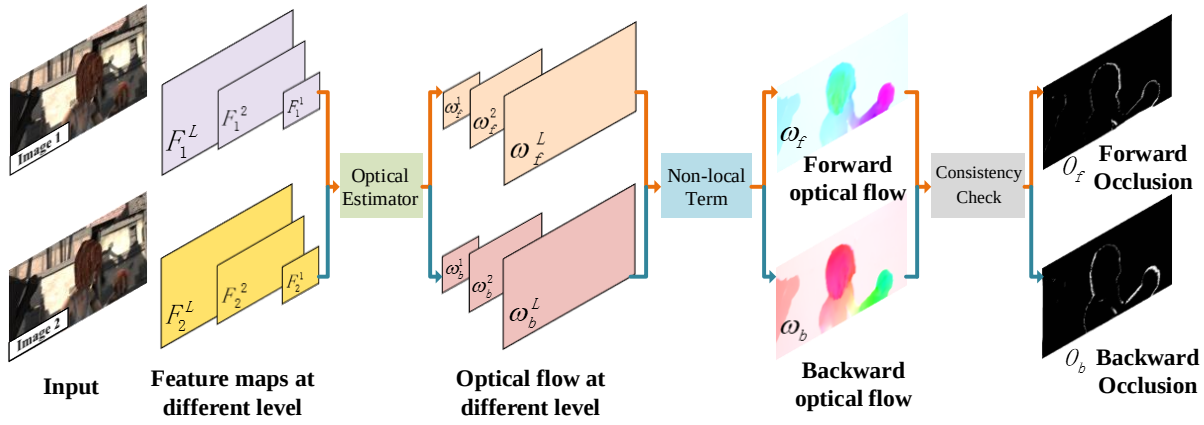


Fig. 1. Illustration of the framework. The entire network is based on the coarse-to-fine strategy. Specifically, we build a feature pyramid on the input two consecutive images. Due to the limited space, we only show the flow estimation modules at the top two levels and the end level. The optical flow estimator has different parameters at different levels. At the end level, the full resolution optical flow is obtained.

More concretely, in the training phase, the input is two consecutive frames. Exchanging the order of the images, we can estimate both the forward and the backward optical flow. Subsequently, the CNN-based non-local term is used to refine the computed optical flow. After that, we utilize the refined bi-directional optical flow to get bi-directional occlusion maps by forward and backward consistency check. Finally, we build a loss function according to the estimated bi-directional optical flow and occlusion maps. We evaluate our approach on the FlyingChairs, MPI-Sintel, KITTI 2012 and KITTI 2015 benchmarks. The approach yields the highest accuracy among all unsupervised learning-based optical flow methods. The comprehensive ablation study validates the effectiveness of our unsupervised optical flow technique.

We summarized the contributions of this work as follows:

- We propose a novel end-to-end neural network to estimate optical flow in an unsupervised manner to handle the problems of occlusion and lacking large-scale datasets with ground-truth.
- We introduce a CNN-based non-local term, which can learn the dependencies over a large spatial neighborhood, so that the flow estimates can be robustly integrated in the optimization process to prevent over-smoothing across motion boundaries.
- The CNN-based non-local term can be widely used as it can be easily combined with other CNN-based optical flow models.
- With high robustness and efficiency, the proposed model achieves high accuracy on the MPI-Sintel, KITTI 2012/2015 datasets.

II. Related Work

A large number of studies have been carried out to improve the performance of optical flow estimation for a long time [19], especially on handling outliers and occlusions [3]. In this section, we consider three classes of related optical flow techniques, i.e., variational methods, supervised deep learning methods, and unsupervised deep learning methods. In addition, we will show different ways of refining optical flow in section II-D.

A. Variational Methods

Horn and Schunck (HS) [19] creatively combined the two-dimensional velocity field with the change in brightness to construct a basic data term based on the BAC, and further designed a smoothness term according to the overall smoothing constraint. With these two terms, the theoretical structure of variational optical flow model was established. Because the variational optical flow approach has many advantages, it became the mainstream in optical flow calculation. Subsequent variational optical flow algorithms were almost all derived and extended from the HS algorithm [3]. To deal with the large displacement problem, Black *et al*. [20] proposed a coarse-to-fine variational method, and introduced a robust framework to deal with outlier, i.e., brightness inconstancy and spatial discontinuities. The coarse-to-fine strategy is a heuristic work to handle large displacements which has a huge impact on subsequent optical flow methods [21] [22] [23]. To handle outliers and reduce over-smoothing at motion boundaries, Sun *et al*. [8] introduced a non-local term which robustly integrates flow estimates over large spatial neighborhoods, [24] took into account the symmetry across the images as well as possible occlusions in the flow field. Tu *et al*. [9] proposed a method, which uses edge detection to extract flow edges and piecewise occlusion detection to extract occlusions, to handle both occlusion and over-smoothing. A weighted median filter, a bilateral filter and a fast median filter were together utilized to post-process the detected edges and occlusions. However, these methods require a lot of computation, making them less likely to work in real-time.

B. Supervised Methods in Deep Learning

In recent years, using CNN to build a model to learn optical flow in an end-to-end way has achieved great success [3]. Dosovitskiy *et al*. [4] pioneered to use CNN to calculate optical flow. They proposed a network called FlowNet, which

contains two CNN networks – FlowNetS and FlowNetC. They are implemented in trainable encoder-decoder networks end-to-end. The networks were pre-trained on a large synthetic dataset (FlyingChairs), and fine-tuned on other datasets. The FlowNet is very fast, but its flow field contains a lot of errors in smooth regions and thus variational refinement is required. Inspired by the FlowNet, Ilg *et al*. [5] proposed a network called FlowNet 2.0, which extends FlowNet by stacking multiple encoder–decoder networks. It obtained much more accurate result than FlowNet, but the parameters and computational complexity were increased significantly. Ranjan *et al*. [25] explored a compact network termed SpyNet, which was derived from the spatial pyramid. Sun *et al*. [6] further improved the performance of SpyNet, and presented an effective model, which combined different good practices from optical flow and stereo matching, by training a shallow Siamese network and constructing a cost volume at different scales. Hui *et al*. [7] proposed a model named LiteFlownet, which conducted flow inference at each pyramid level via a lightweight cascaded network, and used a feature-driven local convolution to deal with the issue of outliers and blurry flow boundaries. However, these supervised models share a big limitation: they need a large amount of data with ground-truth optical flow. Unfortunately ground-truth flow is difficult to obtain for real videos.

C. Unsupervised Methods in Deep Learning

To address the problem of lacking ground-truth flow, one promising way is to exploit unsupervised learning. Jason *et al*. [12] presented a method to calculate optical flow via brightness constancy and motion smoothness in an unsupervised way. Similarly, Ren *et al*. [13] suggested an unsupervised method based on FlowNet to sidestep the limitation of synthetic annotated datasets. They replaced the original supervised loss with the classic photometric loss. But the accuracy was unsatisfactory, and one reason is that when the pixels are occluded, the photometric loss would provide misleading information. Wang *et al*. [14] first inferred occlusion maps and then utilized them in computing the photometric difference. Hur *et al*. [26] proposed to learn optical flow and occlusion jointly, and showed how to utilize occlusion as an important cue for estimating optical flow. Meister *et al*. [15] proposed a model named UnFlow, which designed an unsupervised loss based on occlusion-aware bidirectional flow estimation to avoid the need of ground-truth flow. Neoral *et al*. [27] presented a ContinualFlow method which estimated optical flow by using a multi-frame formulation with temporal consistency. It utilized more data with advanced occlusion reasoning for getting better accuracy, but it lacks the capability to learn optical flow in occluded regions. Additionally, Lai *et al*. [18] proposed an unsupervised network to jointly learn spatiotemporal correspondence for stereo matching and optical flow estimation. It also includes occlusion extraction, where the stereo matching can help the model understanding the depth information in the scene. This strategy is effective to improve the accuracy of optical flow estimation but increases the complexity of the model. In this paper, we only use two consecutive images to learn the bi-directional optical flow and the occlusion map. Moreover, we apply a CNN-based non-local term to alleviate the boundary smoothness.

D. Refinement

Traditional optical flow methods often use contextual information to post-process the flow field. Median filtering and bilateral filtering are common approaches used to prevent smoothing across motion boundaries [9]. Buades *et al*. [17] proposed a non-local algorithm for image denoising, which can capture long-range dependencies. It alleviates boundary over-smoothing during the denoising process. Sun *et al*. [8] showed how to incorporate a weighted version of the non-local term into the variational model to refine optical flow. They suggested that if we know a pair of pixels belonging to the same surface, a higher weight should be given to them. In this way, boundary blur is reduced, and weights are approximated according to their spatial distance, color-value distance and occlusion state. However, it is not only difficult to calculate weights, but also difficult to incorporate with current deep networks. Inspired by the classical non-local technique, Wang *et al*. [28] constructed a non-local term to compute the response at a position as a weighted sum of the deep features at all positions within a deep neural network. This simple operation shows effectiveness in the application of video classification. Motivated by the above mentioned works, we design a CNN-based non-local term to refine the extracted optical flow, which can be easily integrated with deep models to overcome over-smoothing.

III. The Proposed Method

The schematic structure of our framework is depicted in Fig. 1. Based on the coarse-to-fine strategy [7], [6], [25], we propose an end-to-end method to estimate (forward) optical flow in an unsupervised manner. The input are two successive images, and the output is the (forward) estimated optical flow.

During the training process, to detect occlusion, we also need inverse (backward) optical flow [18], [25]. Optical flow and occlusion maps are integrated into our loss function to constrain each other. Note that, for the sake of brevity, some explanations only discuss on the forward optical flow. Exchanging the order of the input images, we can obtain the backward optical flow, and the reasoning is implemented in the same way.

Following the previous work [6] [7], we first build a feature pyramid from the two input images. At the *top* level, we use the features of the first image and the features of the second image to construct forward cost volume. The forward cost volume and features of the first image are fed into the optical flow estimator to compute the forward flow at the top level. After that, the estimated forward flow is upsampled and rescaled to the *second* level.

At the *second* level, the way to compute optical flow is a little different from the top level. We warp the feature map of the second image toward the first using the upsampled forward flow from the top level to obtain a forward warped feature. The features of the first image and the forward warped features are

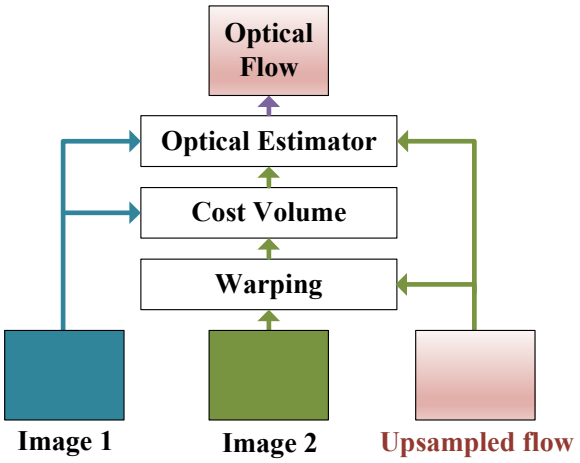


Fig. 2. Illustration the extraction process of the forward optical flow at l-layer: (2) Getting cost volume based on the future of image 1 and the warped image 2. (3) Estimating optical flow based on the future of image 1, the cost volume and the upsampled flow.

used to construct the forward cost volume. Next, the features of the first image, the forward cost volume, and the upsampled forward flow from the top level are fed to the optical flow estimator to obtain the forward flow at the second level. The process repeats until reaches to the desired level. The details of the process are shown in Fig. 2.

We utilize the CNN-based non-local term to refine the estimated optical flow at the *end* level, which can learn long-range dependencies directly. We will show the implementation details of the non-local term in section III-E. In this way, the motion boundary can be effectively prevented from being over-smoothing. Finally, the refined flow is used to generate the occlusion map. We integrate the flow and the occlusion map into the energy formulation, which can better estimate both forward and backward flow through iterative optimization.

We will describe the framework in detail from the following aspects: In Section III-A, we will define our notations. In Section III-B, the method of feature extraction will be introduced. Feature warping will be explained in Section III-C. How to estimate optical flow and occlusion will be respectively illustrated in Sections III-D and III-F. Most importantly, the CNN-based non-local term and the new loss function will be described in Sections III-E and III-G separately.

A. Notation

The two input RGB images are I_1, $I_2 \in R^{H\times W\times 3}$, where H and W respectively represent the height and width. The forward and backward optical flow are denoted by w_f and w_b. Specifically, the forward optical flow w_f is computed from I_1 to I_2, while the backward optical flow w_b is estimated from I_2 to I_1, where w_f, $w_b \in R^{H\times W\times 2}$. We use O_f and O_b to indicate the forward and backward occlusion maps, where O_f, $O_b \in R^{H\times W\times 1}$. Taking the forward occlusion map as an example, the white area denotes the area in the first image does not have a correspondence in the second image, see Fig. 1.

B. Feature Extraction

First of all, for the two input images I_1 and I_2, we extract L multi-scale feature representations, where L denotes the number of pyramid layers and the bottom level being the input images. We use F_1^l and F_2^l to represent the extracted CNN feature maps of I_1 and I_2 at the l-th pyramid level respectively. Inspired by [6], we use layers of convolutional filters to downsample the feature at the l+1 pyramid level to generate feature representation at the l-th layer. For our model, there are six layers in the pyramid. From the first to the sixth pyramid levels, we set the number of feature channels at each convolutional layer fixed to 16, 32, 64, 96, 128 and 192 respectively [6], [7], [25].

C. Feature Warping

If the two images have large displacements, it will be difficult for the network to get high quality optical flow. To handle this problem, we add a feature warping layer [6] [7]. At the l-th level, we warp F_2^l towards F_1^l via the estimated flow to obtain $F_{f\omega}^l$, the implementation is expressed as:

$$F_{f\omega}^l = F_2^l\big(x + up\omega_f^l(x)\big), \tag{1}$$

where x is the pixel index, and $up\omega_f^l$ is upsampled flow from the (l-1)-th level. Notably, at the top level, there is no upsampled flow, we simply set $up\omega_f^1$ and $up\omega_b^1$ to zero, i.e., $F_{f\omega}^1$=F_2^1 and $F_{b\omega}^1$=F_1^1.

The bilinear interpolation [3] is applied to implement the warping operation. For non-translational motions, warping can compensate some geometric distortions and put image patches at the right scale [6].

D. Optical Flow Estimation

To store the matching cost for associating a pixel with its corresponding pixels at the next frame, we use deep features to construct a partial cost volume [6], [7], [29] at multiple pyramid levels. The matching cost is defined as the correlation [30] between features of the two images. The cost volume uses feature of a pixel in the first image with corresponding feature in the second image, and only needs to calculate a limited range of d pixels, i.e., $|x_1 - x_2|_\infty \leq d$. At the top level, the spatial resolution is small, a small search range is used. In the remaining levels, to reduce the search range, we use the warped feature map and the feature map of the first image to construct the cost volume, defined as:

$$cv_f^l(x_1, x_2) = \frac{1}{N}(F_1^l(x_1))^T F_{f\omega}^l(x_2). \tag{2}$$

The dimension of the cost volume is $d^2 \times H^l \times W^l$, where H^l and W^l denote the height and width of the l-th pyramid level, respectively.

The optical flow estimator is built on a multi-layer CNN. To be specific, at the l-level, the inputs are the deep feature map of the first image F_1^l, the upsampled forward optical flow $up\omega_f^l$ and the cost volume cv_f^l, and the output is the calculated forward optical flow ω_f^l, see Fig. 2.

The optical flow estimators at different levels have their own parameters instead of sharing the same parameters. Importantly, since the forward flow is inverse to the backward flow, we share the weights of forward and backward flow estimators to reduce computational cost, see Fig. 1.

E. CNN-based Non-local Term

For the traditional non-local term [8], [28], it not only compares the gray value of a single point, but also compares the geometry of the entire neighborhood. Specifically, if a pair of pixels have similar gray values but the structures around the pixels are not similar, small weights are assigned to them. Consequently, this denoising strategy ensures to retain the detailed information of the image and reduces over-smoothing at image boundaries. However, it is costly to run and difficult to combine with the deep learning approach.

Inspired by [8], [28], we build a non-local term with deep learning to refine the optical flow. Our non-local term can automatically learn the relationship of pixels in the flow field, and effectively alleviate over-smoothing crossing motion boundaries. Because our non-local term is based on deep learning, it can be easily integrated into deep convolutional networks, with parameters updated according to the tasks in an end-to-end manner.

Specifically, our non-local term is defined as:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j), \tag{3}$$

where i is the index of an output position whose response is to be computed, and j is the index that enumerates all possible positions. $C(x)$ denotes a normalizing factor. In Eq. (3), we consider not only local neighborhood but also all pixels over the flow field.

$f(x_i, x_j)$ is a pairwise function which computes a scalar, and represents the relationship affinity between x_i and all x_j. If x_i and x_j are on the same surface and have similar peripheral structure information there should be a higher weight, otherwise the weight should be small. Accordingly, we define the $f(x_i, x_j)$ as:

$$f(x_i, x_j) = e^{(W_\theta x_i)^T W_\psi x_j}, \tag{4}$$

where $C(x) = \sum_{\forall j} f(x_i, x_j)$, $W_\theta x_i$ and $W_\psi x_j$ are two embeddings.

$g(x_j)$ is a unary function computes a representation of the input signal at the position j, and we consider it in the form of a linear embedding, which is defined as:

$$g(x_j) = W_g x_j, \tag{5}$$

where W_g is a weight matrix to be learned.

Our non-local term consists of only five convolutional layers and is performed at the last pyramid level, which means that it is relatively light to compute. What's more, unlike the traditional way where the weights are manually designed, our CNN-based non-local term can automatically learn the weights. It will assign a weight value $f(x_i, x_j)$ based on the two pixels x_i and x_j whether they are located on the same surface, whether they have similar color-space characteristic, and whether they have similar structural information. Due to this power, our CNN-based non-local term is effective to prevent over-smoothing across motion boundaries and can be applied to complex scenes. In Section V-E, we will test the effectiveness of the proposed CNN-based non-local term by conducting various experiments.

F. Occlusion Estimation

Occlusion is a consequence of depth and motion. Accurate detection of occluded regions is crucial for reliable optical flow estimation as they are beneficial for preventing non-occluded areas being adversely affected by occluded pixels [26]. Consequently, estimating accurate optical flow is required to localize occlusions reliably. In this work, we extract occlusion masks as key clues for computing optical flow. Generally, there are three different ways to handle occlusion:

- Treating occlusion as outliers and predicting target pixels in the occluded regions as a constant value or through interpolation [31], [32].
- Building sophisticated framework to reason occlusion [33].
- Dealing with occlusion by exploiting the symmetric property of optical flow and ignoring the loss penalty on predicted occluded regions [3], [26], [34].

Our method is similar to the last one, where the occlusion detection is performed according to the forward-backward consistency checking [34]. We consider pixels as occluded when the mismatch between the forward and backward flow is too large or the flow is out of image boundary Ω. We follow the technique of [15], which defines the occlusion flag $O_f(x)$ to be 1 when the constraint Eq.(6) is violated and 0 otherwise.

$$\begin{aligned} &\left|\omega_f(x) + \omega_b\big(x + \omega_f(x)\big)\right|^2 \leq \\ &\qquad \alpha_1\Big(\left|\omega_f(x)\right|^2 + \left|\omega_b(x + \omega_f(x)\right|^2)\Big) + \alpha_2, \end{aligned} \tag{6}$$

where we set α_1=0.01 and α_2=0.05 for all experiments. The backward occlusion maps are computed in the same way, just with ω_f and ω_b exchanged. In the Section V-F, we will prove the effectiveness of occlusion maps.

G. Loss Function

In this paper, we add the loss function to the end of the network. We jointly integrate the forward-backward flow consistency checking and occluded/non-occluded symmetry operation into the loss formulation. The occlusion and non-occlusion information can be used to improve the performance of both backward and forward optical flow estimation. For non-occluded pixels, the forward flow should be inverse to the backward flow at the corresponding pixel position in the second frame, which means that corresponding pixels should be similar after mapping by the flow. Accordingly, we add a forward-backward consistency penalty [15], [18] as following:

$$\begin{aligned} L_C(\omega_f, \omega_b, O_f, O_b) = &\\ &\sum_{x \in P} (1 - O_f) \cdot \sigma(\omega_f(x) + \omega_b(x + \omega_f(x)) \\ &+ (1 - O_b) \cdot \sigma(\omega_b(x) + \omega_f(x + \omega_b(x)), \end{aligned} \tag{7}$$

where σ is a robust generalized Charbonnier penalty function [35] and defined as:

$$\sigma(x) = (x^2 + \varepsilon^2)^\tau, \tag{8}$$

we set ε=0.01 and τ=0.4 for all of our experiments.

We design an occlusion-aware data loss as:

$$\begin{aligned} L_D(\omega_f, \omega_b, O_f, O_b) = \\ \sum_{x \in p} (1 - O_f) \cdot \sigma(f_D(I_1 - I_2(x + \omega_f))) \\ + (1 - O_b) \cdot \sigma(f_D(I_2 - I_1(x + \omega_b))), \end{aligned} \tag{9}$$

where f_D function is used for measuring the photometric difference.

The final loss function is expressed as:

$$L_{(I_1, I_2, \omega_f, \omega_b, O_f, O_b)} = L_C + L_D. \tag{10}$$

The brightness constancy is not invariant to illumination changes in real world [36]. To increase the robustness, we use ternary census transform [37], which can compensate additive and multiplicative illumination changes, and provide a more reliable constancy for realistic imagery.

To sum up, we take full advantage of the relationship and symmetry between optical flow and occlusion, and use them as important clues to help promote each other. This contribution significantly improves the accuracy of optical flow estimation in the unsupervised-learning manner. We will show the influence of occlusion in section V-F.

IV. Training Procedure

Algorithm 1 shows the pseudo-code for training. The input is two consecutive frames I_1 and I_2, and the output is the forward optical flow ω_f. In the inner loop of Algorithm 1, at the L-th pyramid level, we extract the feature information F_1^l and F_2^l from I_1^l and I_2^l respectively. Simultaneously, we upsample $\omega_f^{(l-1)}$ to $up\omega_f^l$. We use Eq. (1) to warp F_2^l towards F_1^l via upsampled flow $up\omega_f^l$ and obtain the warped feature $F_{f\omega}^l$. We combine F_1^l and $F_{f\omega}^l$, then use Eq. (2) to calculate the cost volume cv_f^l. Finally, cv_f^l, F_1^l and $up\omega_f^l$ are input to the optical flow estimator to compute the l-level optical flow ω_f^l. Notably, at the top pyramid level, we set optical flow to zero. After the inner loop, we obtain a full resolution optical flow, and Eq. (3) is used to regularize the optical flow and refine optical flow ω_f. Occlusion maps are computed by forward-backward consistency checking.

Training is conducted via the stochastic gradient descent method over shuffled mini-batch. Following the prior works [12], [13], we first pre-train the model on a large synthetic dataset FlyingChairs with 300k iterations to ensure our model has good generalization ability. Then, we fine-tune the pre-trained model on MPI-Sintel and KITTI datasets respectively. Parameters are updated by using back-propagation. We utilize Adam [38] as the optimizer with initial learning rate of 1e-4, which decays half every 50k iterations. The model is trained by using TensorFlow. To benefit from the efficiency of the parallel computation of the tensors, all simulation studies are conducted with three NVIDIA Pascal TitanX GPU on an Ubuntu PC. Testing is implemented with one single GPU.

Algorithm 1: Pseudo-code for training

Input: Two consecutive frames I_1, I_2
Output: Optical Flow ω_f
for i *in* [1, *epoch*] **do**
  for l *in* [1, L] **do**
    if l==1 **then**
      Set $up\omega_f^1$, $up\omega_b^1$ to zero;
    else
      $F_1^l, F_2^l \leftarrow I_1^l, I_2^l$; *// Feature extraction.*
      $up\omega_f^l$, $up\omega_b^l \leftarrow \omega_f^{(l-1)}$, $\omega_b^{(l-1)}$; *// Upsample.*
      $F_{f\omega}^l$, $F_{b\omega}^l \leftarrow (F_2^l, up\omega_f^l)$, $(F_1^l, up\omega_b^l)$; *// Warping.*
      cv_f^l, $cv_b^l \leftarrow (F_1^l, F_{f\omega}^l)$, $(F_2^l, F_{b\omega}^l)$; *// Obtaining cost volume.*
      ω_f^l, $\omega_b^l \leftarrow (cv_f^l, F_1^l, up\omega_f^l)$,$(cv_b^l, F_2^l, up\omega_b^l)$; *// Optical flow extraction.*
  ω_f, $\omega_b \leftarrow \omega_f^L$, ω_b^L; *// Non-local refine.*
  O_f, $O_b \leftarrow \omega_f$, ω_b; *// Occlusion reasoning.*
  Minimize Eq.(10) with the Adam update rule [38].
End for

V. Experiments

We evaluate our method on standard optical flow benchmark datasets including FlyingChairs [4], MPI-Sintel [10], KITTI 2012 [39] and KITTI 2015 [40], and compare our results with other state-of-the-art deep learning based supervised and unsupervised optical flow approaches. We will detail the benchmark datasets in Section V-A. In Section V-B, the evaluation metrics used in this paper will be introduced. The experimental results are shown and analyzed in Section V-C. Specifically, the role of occlusion reasoning and the CNN-based non-local regularization will be analyzed in detail in Sections V-F and V-E respectively.

A. Datasets

1) FlyingChairs: The FlyingChairs is a large synthetic dataset with optical flow ground-truth. It consists of segmented background images from Flickr [4], and contains 22872 image pairs as well as their corresponding flow fields. Motions of both the chairs and the background are purely planar. We crop a 448×384 image patch of every frame and use a batch size of 4.

2) MPI-Sintel: The Sintel benchmark [10] is created by using the open source graphics movie "Sintel" with two passes, i.e., Clean and Final. The original resolution of Sintel images is 1024×436. The dataset contains flow fields, motion boundaries, unmatched regions and image sequence. MPI-Sintel provides ground-truth for training, and also supplies pixel-wise occlusion masks. We crop a 768×384 image patch of every frame and use a batch size of 4.

3) KITTI: The KITTI dataset provides real road scenes which were captured from a mobile platform. A laser scanner provides accurate but sparse ground-truth optical flow for a small number of images. KITTI contains two classes, i.e. KITTI 2012 [39] and KITTI 2015 [40]. KITTI 2012 consists of

194 training image pairs and 195 test pairs, while KITTI 2015 consists of 200 training scenes and 200 test scenes. The size of the first 156 sequences of KITTI 2015 is 375×1242, but the last 44 sequences with different resolutions, e.g., 370×1224, 374×1238 and 376×1241. We fine-tune our model on these two classes separately, and crop a 896×320 image patch on each frame. The batch size is set to 4.

B. Evaluation Metrics

Two standard metrics are selected for evaluation.

- **Average End Point Error (EPE)** is defined as the average Euclidean distance between the estimated flow and the ground-truth flow, which is expressed as follows:

$$EPE = \frac{1}{N}\sum_{N}\sqrt{(v_x - v_x^{gt})^2 + (u_x - u_x^{gt})^2}, \quad (11)$$

  where x denotes a pixel point. v_x and u_x represent the optical flow predicted in the horizontal and vertical directions respectively, u_x^{gt} and u_x^{gt} represent the ground-truth flow in the horizontal and vertical directions respectively.
- **Percentage of Erroneous** for KITTI Fl-all and Fl-noc are reported. Fl-all denotes the ratio of pixels where the estimated flow is wrong, and Fl-noc represents over non-occluded pixels only.

TABLE I
ACCURACY (EPE) COMPARISON WITH THE **SUPERVISED** METHODS ON THE MPI-SINTEL DATASET. THE FIRST 16 ROWS ARE THE RESULTS OF THE SUPERVISED METHODS. (-) DENOTES THAT THE ORIGINAL PAPER DOES NOT GIVE A CORRESPONDING REPORT. OUR (CHAIRS) MEANS THE MODEL IS PRE-TRAINED ON FLYINGCHAIRS DATASET. OUR+FT-SINTEL MEANS FINE-TUNE (FT) THE PRE-TRAINED MODEL ON THE MPI-SINTEL DATASET.

	Method	Chairs test	Sintel train Clean	Sintel test Clean	Sintel train Final	Sintel test Final
Supervised	DDF [41]	-	-	-	-	5.73
	PatchBatch [42]	-	-	-	-	5.36
	FlowFieldCNN [43]	-	-	-	-	-
	LDOF [21]	3.47	4.46	7.56	5.96	9.12
	DeepFlow [22]		2.66	5.38	3.57	7.21
	Classic+NLP [35]		4.49	6.73	7.46	8.29
	PCA-Layers [44]		3.22	5.73	4.52	7.89
	EpicFlow [23]		2.27	4.12	3.56	6.29
	SpyNet [25]	2.63	4.12	6.69	5.57	8.43
	SPyNet+FT [25]	-	3.17	6.64	4.32	8.36
	FlowNet2 [5]	-	2.02	3.96	3.14	6.02
	FlowNet2+FT [5]	-	1.45	4.16	2.01	5.74
	PWC-Net [6]	2.00	3.33	-	4.59	-
	PWC-Net+FT [6]	-	1.70	3.86	2.21	5.13
	LiteFlowNet [7]	-	2.48	-	4.04	-
	LiteFlowNet+FT [7]	-	1.34	4.54	1.78	5.38
	Our(Chairs)	2.98	3.68	-	4.12	-
	Our+FT-Sintel	3.51	2.58	7.12	3.85	8.51

C. Results on benchmarks

We compare our model with state-of-the-art supervised and unsupervised models on the MPI dataset in Table I and Table II, respectively. Besides, we also compare our method with the well-known supervised and unsupervised methods on the KITTI dataset in Table VI and Table VII.

TABLE II
ACCURACY (EPE) COMPARISON WITH THE **UNSUPERVISED** METHODS ON THE MPI-SINTEL DATASET. THE FIRST 9 ROWS INDICATE THE RESULTS OF THE UNSUPERVISED METHODS, WHILE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

	Method	Chairs test	Sintel train Clean	Sintel test Clean	Sintel train Final	Sintel test Final
Unsupervised	BackToBasic+FT [12]	5.30	-	-	-	-
	DSTFlow+FT [13]	5.11	6.16	10.41	6.81	11.27
	UnFlow-CSS [15]	-	-	-	7.91	10.22
	OccAwareFlow [14]	3.30	5.23	8.02	6.34	9.08
	OccAwareFlow+FT+Sintel [14]	3.76	4.03	7.95	5.95	9.15
	OccAwareFlow+FT+KITTI [14]	-	7.41	-	7.92	-
	MultiFrameOccFlow-Hard [16]	-	5.38	8.35	6.01	9.38
	MultiFrameOccFlow-Hard+FT [16]	-	6.05	-	7.09	-
	MultiFrameOccFlow-Soft+FT [16]	-	3.89	7.23	5.52	8.81
	Our(Chairs)	**2.98**	3.68	-	4.12	-
	Our+FT-Sintel	3.51	**2.58**	**7.12**	**3.85**	**8.51**

1) Pre-training on FlyingChairs: We use **"Our(Chairs)"** to indicate the model that was only pre-trained on the FlyingChairs dataset. As shown in Table II, on the testing set of FlyingChairs, the EPE of "Our(Chairs)" is 2.98, which is at least 10.7% more accurate than the previous methods (2.98 (Our) vs 3.30 [14]). Furthermore, as shown in Table I, the result of "Our(Chairs)" approaches to the best supervised-learning optical flow method. This is due to our occlusion processing strategy, and the contribution of the CNN-based non-local term which can remove noise and decrease over-smoothing at motion boundaries. The results prove that our model has a strong generalization ability.

2) Fine-tuning on MPI-Sintel: We utilize **"Our+FT-Sintel"** to represent the model which has been fine-tuned through the MPI-Sintel dataset. Notably, when comparing on the Sintel Clean set, we use the training set of Sintel Clean for fine-tuning. When comparing on the Sintel Final set, we use the training set of Sintel Final for fine-tuning. Table II shows that "Our+FT-Sintel" gets EPE=2.58 and EPE=3.85 on the training set of Sintel Clean and Final respectively. Besides, it obtains EPE=7.12 on the testing set of Sintel Clean, which surpasses the previous best result obtained by MultiFrameOccFlow-Soft+FT [16] (7.12 vs 7.23). "Our+FT-Sintel" gets EPE=8.51 on the Sintel Final testing set, which outperforms the best prior result of [16] by 3.4% (8.51 vs 8.81).

Fig. 3 shows the visualization results of "Our+FT-Sintel" on the Sintel Clean pass and the Sintel Final pass, and also gives the comparison of our estimated optical flow and occlusion masks with the ground-truth. It can be clearly observed from Fig. 3 that the optical flow boundary obtained by our model is quite clear and can distinguish different surfaces well. Even if the original image has blurred boundaries (such as Sintel Final), our model can still estimate optical flow with clear boundary. The experimental results prove that our designed CNN-based non-local term can effectively reduce the noise of the optical flow while retaining details.

In Section V-E, we show the effect of the proposed non-local on improving the performance of estimating optical flow, and visualize the results when the model does not add the non-local term. Besides, we analyze the impact of the non-local term on the computing time.

TABLE III
RESULTS ON THE SINTEL BENCHMARK FOR DIFFERENT REGIONS. "EPE MATCHED" DENOTES END POINT ERROR (EPE) OVER REGIONS THAT REMAIN VISIBLE IN ADJACENT FRAMES, WHILE "EPE UNMATCHED" DENOTES EPE OVER REGIONS THAT ARE VISIBLE ONLY IN ONE OF TWO ADJACENT FRAME. "D0-10" DENOTES EPE OVER BETWEEN 0 AND 10 PIXELS APART FROM THE NEAREST OCCLUSION BOUNDARY. SIMILARLY "D10-60" AND "D60-140" ARE JUST DIFFERENT IN REGIONS. "S0-10" DENOTES EPE OVER REGIONS WITH VELOCITIES BETWEEN 0 AND 10 PIXELS PER FRAME. SIMILARLY "S10-40" AND "S40+" ARE JUST DIFFERENT IN REGIONS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

MPI Final	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	d0-10	d10-40	s40+
Our	**8.51**	**4.18**	43.87	**6.34**	**3.91**	**2.94**	1.82	**4.99**	**50.23**
MultiFrameOccFlow [16]	8.81	5.03	**39.65**	7.15	4.88	3.90	1.75	5.96	50.73
UnFlow [15]	10.22	6.06	44.11	8.41	5.83	4.67	**1.74**	6.69	60.77
MPI Clean	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	d0-10	d10-40	s40+
Our	**7.17**	**2.99**	41.26	**5.36**	**2.70**	**1.85**	1.26	**3.41**	45.61
MultiFrameOccFlow [16]	7.23	3.60	**36.78**	6.10	3.49	2.46	**1.22**	4.03	**44.84**
UnFlow [15]	9.38	5.37	42.11	7.85	5.15	3.89	1.31	5.11	59.71

TABLE IV
IMPACT OF USING GROUND-TRUTH DATA ON RESULTS. THE FIRST LINE INDICATES THE PERCENTAGE OF DIFFERENT GROUND-TRUTHS ADDED. THE SECOND LINE REPRESENTS THE RESULTS ON THE FLYINGCHAIRS TESTING SET. THE MEASUREMENT IS EPE.

Added Ground-truth	10%	25%	50%	100%
Results	2.61	2.35	2.01	1.85

TABLE V
RESULTS OF OUR MODEL AND UNFLOW ON DIFFERENT DATASETS WITH USING THE SAME PRE-TRAINING DATASET (SYNTHIA). THE MEASUREMENT IS EPE.

Datasets	Sintel Final (training set)	KITTI 2015 (training set)
UnFlow [15]	7.91	8.10
Our	5.22	7.15
Improvement	51.5%	13.3%

Moreover, Table III reports the EPE results of different regions on the MPI-Sintel dataset. It can be found that our model performs relatively better than other methods in regions with large motion and away from the motion boundaries. This is likely because we employ the coarse-to-fine strategy and use data with large motions to train the network.

3) Fine-tuning on KITTI: We termed **"Our+FT-KITTI"** to denote the model which has been fine-tuned through the KITTI dataset. Notably, when comparing with other models on the KITTI 2012 dataset, we use the training set of KITTI 2012 for fine-tuning. Similarly, when comparing on the KITTI 2015 dataset, we use the training set of KITTI 2015 for fine-tuning.

As shown in Table VII: (1) on the training set of KITTI 2012, "Our+FT-KITTI" gets an EPE=3.02, which is more accurate than the best existing counterpart UnFlow-CSS [15] (EPE=3.29), i.e. the performance is improved by 8.94% (3.02 vs 3.29). On the testing set of KITTI 2012, "Our+FT-KITTI" gains EPE=4.5 and Fl-noc=5.86%. Specifically, our EPE is not as good as OccAwareFlow+FT+KITTI [14] (4.5 vs 4.2 [14]), but OccAwareFlow+FT+KITTI has more complex loss functions and more hyperparameters, while our loss function is more concise and has fewer hyperparameters. (2) On the training set of KITTI 2015, "Our+FT-KITTI" obtains EPE=6.05 and Fl-al=11.2%, which outperforms [16] by 8.93% (6.05 vs 6.59) on the EPE measure. On the testing set, the Fl-al of us is 22.75%, and it is more accurate than the previous best result of MultiFrameOccFlow (22.94%). Notably, MultiFrameOccFlow utilizes multiple frames, while our method use only two consecutive frames.

Fig. 4 shows the visuals results of our fine-tuned model on KITTI. We can find that the captured occlusion map of our model is quite good, and the accurate occlusion map is beneficial for improving the performance of optical flow estimation. Table VIII reports the evaluation results on the KITTI 2015 benchmark testing set. "Our+FT-KITTI" has lower percentage of flow outliers in both "Non-occluded pixels" and "All pixels" than the most recently unsupervised method MultiFrameOccFlow [16].

In Section V-F, we compare the accuracy of the model with/without occlusion extraction to show the effect of occlusion on optical flow estimation. In addition, we analyze the impact of occlusion extraction on the calculation time.

4) Comparison with consistent pre-train strategy: For a comprehensive comparison with other models, we conduct another experiment. Specifically, we use the SYNTHIA dataset [45] to pre-train our model, which is consistent with UnFlow [15]. Then we fine-tune the pre-trained model on Sintel and KITTI datasets. In this case, the EPE of us on Sintel Final (training set) and KITTI 2015 (training set) are 5.22 and 7.15 respectively, while the corresponding results of UnFlow on these datasets are 7.91 and 8.10 separately. The results demonstrated that our model is better than Unflow, whose pre-train strategy is consistent with us (see Table V).

D. Ground-truth boost results

Originally, we used the entire FlyingChairs dataset to pre-train the model in an unsupervised manner. Now we change the way to pre-train the model, that is, add different proportions of ground truth.

Firstly, we use 50% of FlyingChairs data (no ground-truth) to initialize the model, where the loss function is consistent with Eq. 10. Next, we add 10%, 25%, 50% of the remaining data (with ground truth) to further train the model, and use EPE as the loss function, which is a common loss function in supervised models. Finally, we test the mixed training model on the FlyingChairs test set, the results are shown in Table IV.

Note that we can also transform our model into a supervised way by using 100% FlyingChairs data (with ground-truth) to

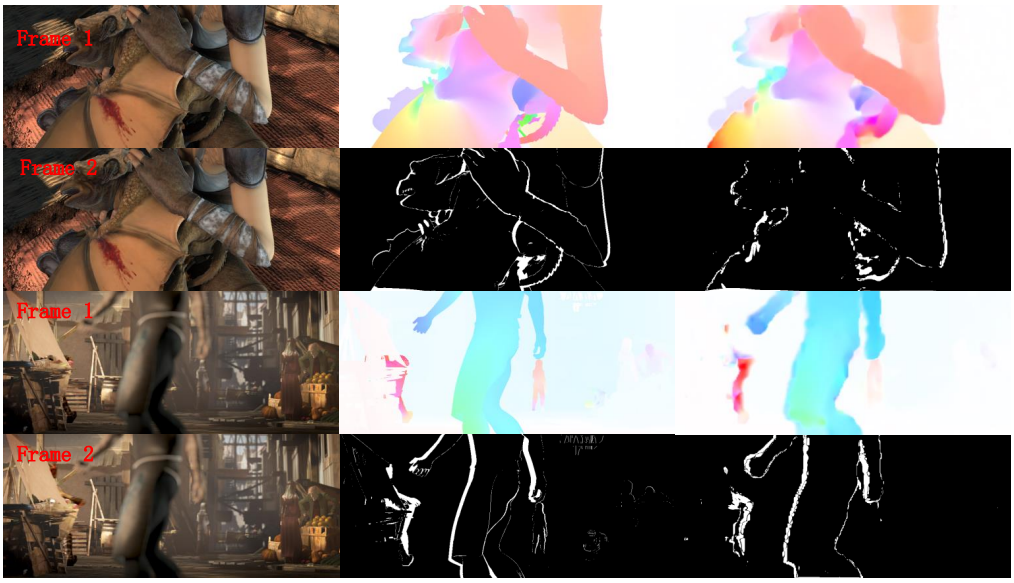


Fig. 3. Illustration of qualitative examples on MPI Clean and MPI Final. Every two rows are a group. At the first group, the first column indicates the input images (from MPI Clean), the top of the second column represents ground-truth flow while the bottom represents ground-truth occlusion map, and the top of the third column denotes the estimated optical flow while the bottom denotes the estimated occlusion map. The second group has the same structure as the first group, but the images are from MPI Final, which means there are more boundary blurs.

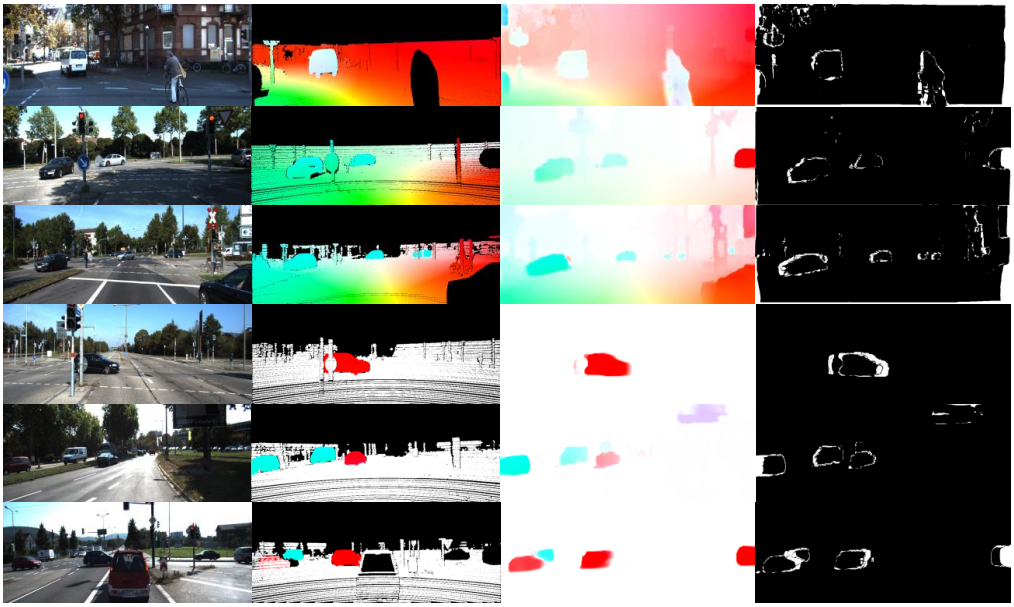


Fig. 4. Results on the KITTI dataset. The first column is the input image, the second column denotes the ground-truth flow with occlusion, the third column is the estimated optical flow, and the last column denotes the extracted occlusion map.

pre-train the model. In this case, the pre-training strategy is the same as FlowNet2 [5], PWC-Net [6], LiteFlowNet [7]. Not surprisingly, the more ground-truth data is used, the better performance our model can get. As shown in the last column of Table IV, if we use a supervised approach to pre-train the model, the results are slightly better than the state-of-the-art supervised model PWC-Net. We obtain EPE=1.85 on the FlyingChairs testing set, while PWC-Net gets EPE=2.0 (see Table I). This may be due to the effect of our CNN-based non-local term, which can effectively mitigate over-smoothing across motion boundaries.

E. Non-local Regularization

To verify the effectiveness of our CNN-based non-local regularization, we conduct following experiments.

As shown in Table XII, when adding the CNN-based non-local term to refine the estimated optical flow, higher accuracy can be obtained. In particular, the EPE has increased by at least 27.70% (4.08 vs 5.21) on the FlyingChairs, 23.79% (4.75 vs 5.88) on the Sintel Clean, and 30.18% (4.87 vs 6.34) on the Sintel Final. Integrating the CNN-based non-local term to the baseline model leads to a larger improvement on the

TABLE VI
PERFORMANCE (EPE, FL-NOC AND FL-AL) COMPARISON WITH THE **SUPERVISED** MODELS ON KITTI DATASET. FL-NOC: PERCENTAGE OF ERRONEOUS PIXELS IN NON-OCCLUDED AREAS. F1-AL: PERCENTAGE OF ERRONEOUS PIXELS IN TOTAL. THE FIRST 16 ROWS INDICATE THE RESULTS OF THE SUPERVISED METHODS. (-) DENOTES THAT THE ORIGINAL PAPER DID NOT GIVE A CORRESPONDING REPORT. OUR (CHAIRS) MEANS THE MODEL IS PRE-TRAINED ON FLYINGCHAIRS DATASET. OUR+FT-KITTI MEANS FINE-TUNE (FT) THE PRE-TRAINED MODEL ON THE KITTI DATASET.

	Method	KITTI 2012			KITTI 2015		
		EPE		Fl-noc	EPE	Fl-al	
		train	test	test	train	train	test
Supervised	DDF [41]	-	3.4	-	-		21.17%
	PatchBatch [42]	-	3.3	5.29%	-	-	21.07%
	FlowFieldCNN [43]	-	3.0	-	-	-	19.80%
	LDOF [21]	10.94	12.4	-	18.19	-	-
	DeepFlow [22]	4.48	5.8	-	10.63	-	29.18%
	Classic+NLP [35]	-	7.2	-	-	-	-
	PCA-Layers [44]	5.99	5.2	-	12.74	-	-
	EpicFlow [23]	3.09	3.8	-	9.27	-	27.10%
	SpyNet [25]	9.12	-	-	-	-	-
	SPyNet+FT [25]	8.25	4.7	12.31%	-	-	35.07%
	FlowNet2 [5]	4.09	-	-	10.6	30.37%	-
	FlowNet2+FT [5]	1.28	1.8	4.82%	2.3	8.61%	11.48%
	PWC-Net [6]	4.57	-	-	10.35	33.67%	-
	PWC-Net+FT [6]	1.45	1.7	4.22%	2.16	9.80%	9.60%
	LiteFlowNet [7]	4.00	-	-	10.39	28.50%	-
	LiteFlowNet+FT [7]	1.05	1.6	-	1.62	5.58%	-
	Our(Chairs)	8.51	-	-	17.25	-	-
	Our+FT-KITTI	3.02	4.5	5.86%	6.05	11.2%	22.75%

TABLE VII
PERFORMANCE (EPE, FL-NOC AND FL-AL) COMPARISON WITH THE **UNSUPERVISED** MODELS ON THE KITTI DATASET. THE FIRST 9 ROWS INDICATE THE RESULTS OF THE UNSUPERVISED METHODS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD. (-) DENOTES THAT THE ORIGINAL PAPER DID NOT GIVE A CORRESPONDING REPORT.

	Method	KITTI 2012			KITTI 2015		
		EPE		Fl-noc	EPE	Fl-al	
		train	test	test	train	train	test
Unsupervised	BackToBasic+FT [12]	11.3	9.9	-	-	-	-
	DSTFlow+FT [13]	10.43	12.4	-	16.79	-	39%
	UnFlow-CSS [15]	3.29	-	-	8.10	-	23.30%
	OccAwareFlow [14]	12.95	-	-	21.30	-	-
	OccAwareFlow+FT+Sintel [14]	12.9	-	-	22.60	-	-
	OccAwareFlow+FT+KITTI [14]	3.55	**4.2**	-	8.88	-	31.2%
	MultiFrameOccFlow-Hard [16]	-	-	-	15.63	41.65%	49.93%
	MultiFrameOccFlow-Hard+FT [16]	-	-	-	11.58	27.29%	-
	MultiFrameOccFlow-Soft+FT [16]	-	-	-	6.59	19.11%	22.94%
	BridgeDepthFlow(Flow) [18]	4.29	-	-	9.70	32.77%	-
	BridgeDepthFlow(Flow+Stereo) [18]	**2.64**	-	-	7.47	28.54%	-
	Our(Chairs)	8.51	-	-	17.25	-	-
	Our+FT-KITTI	3.02	4.5	**5.86**%	**6.05**	**11.2**%	**22.75**%

TABLE VIII
EVALUATION RESULTS ON THE KITTI 2015 BENCHMARK TESTING SET. *bg* DENOTES THE PERCENTAGE OF OUTLIERS AVERAGED ONLY OVER BACKGROUND REGIONS; *fg* REPRESENTS THE PERCENTAGE OF OUTLIERS AVERAGED ONLY OVER FOREGROUND REGIONS; *all* DENOTES PERCENTAGE OF OUTLIERS AVERAGED OVER ALL GROUND-TRUTH PIXEL. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	Non-occluded pixels			All pixels		
	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
MultiFrameOccFlow	12.49%	20.00%	13.85%	22.67%	24.27%	22.94%
Our	**11.01**%	**19.46**%	**12.55**%	**22.50**%	**23.99**%	**22.75**%

Sintel Final than on the Sintel Clean, because the Sintel Final contains more noise than the Sintel Clean.

On the other side, our model runs at about 27 fps on the Sintel dataset (resolution is 1024×436). When we remove the non-local term, the speed becomes to 28 fps under the same data. It reveals that the computational time is just increasing about 3.5% with the application of our non-local term. Its impact on the calculation time is acceptable, as it contains only five convolutional layers and just conducts at the end level of the pyramid.

Fig. 5 shows the visual results of the CNN-based non-local term. We can find that when the non-local term is not used, the estimated optical flow is blurred at the boundary. In contrast, when the non-local term is employed, motion blur at the boundary is effectively alleviated, and the final accuracy is significantly enhanced.

Furthermore, we compare the effects of the CNN-based non-local term at different stages of the deep learning process. In this paper, the baseline is adding the non-local term to the end layer of the pyramid. For comparison, we incorporate the non-local term to every layer of the pyramid as done by the traditional way [35]. Table IX shows that the accuracy is boosted by about 3%, but the computation cost is increased by 100%. According to the experimental results, we can find that using non-local term to denoise the flow at each level of

Fig. 5. Visually illustration the effect of our proposed non-local term. The left column shows the results of adding the non-local term to baseline model, while the right column shows the results of without the non-local term.

the pyramid does increase the accuracy, but results in much more computational burden. Adding the non-local term only at the end level of the pyramid enables the model to balance between the efficiency and accuracy, and hence more proper for real-time applications.

TABLE IX
THE EFFECT OF ADDING THE NON-LOCAL TERM AT DIFFERENT STAGES. WHERE "EACH" MEANS THAT NON-LOCAL TERM IS ADDED TO EACH LAYER OF THE PYRAMID, AND "END" DENOTES THE NON-LOCAL TERM IS ONLY ADDED AT THE END OF THE PYRAMID. THE MEASUREMENT IS EPE AND THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

stage	Sintel Clean train	Sintel Final train	KITTI 2012 train	KITTI 2015 train
Each	**2.43**	**3.76**	**2.95**	**5.89**
End	2.58	3.85	3.02	6.05

TABLE X
THE EFFECT OF DIFFERENT NUMBER OF LAYERS OF THE NON-LOCAL TERM. THE FIRST ROW INDICATES THE NUMBER OF LAYERS. THE SECOND ROW SHOWS THE RESULTS ON THE FLYINGCHAIRS TESTING SET, WHERE THE MEASUREMENT IS EPE. THE THIRD ROW IS THE COMPUTATION TIME RATIO COMPARED TO THE BASELINE (5 LAYERS).

Number of layers	3 layers	5 layers	7 layers	9 layers
EPE	3.21	2.98	2.88	2.81
Improvement	70%	100%	130%	170%

Additionally, we compared the effect of the number of layers on both accuracy and efficiency. As shown in Table X, when the number of non-local layers increases, the accuracy improves a little but its computational time increases significantly. E.g., compared to 5 layers, when the number of layers is 9, its accuracy increases by 0.17 while the calculating time increases by 70%. Consequently, we chose 5 layers to balance the accuracy and the efficiency.

Summary: Our CNN-based non-local term works particularly well when incorporating it into the deep architecture to estimate optical flow, as it can learn to find meaningful relational clues regardless of the distance, and also it can robustly integrate the flow estimate over large spatial neighborhood. Accordingly, our non-local term is able to remove noise and recover sharp motion boundaries.

F. Occlusion Reasoning

We evaluate our occlusion estimation on both MPI-Sintel and KITTI datasets. We compare our method quantitatively to MultiFrameOccFlow [16], OccAwareFlow [14], S2D [46] and MODOF [47] by calculating the maximum F-measure introduced in [46], as shown in Table XI. Specifically, S2D used a binary classification, and employed ground-truth occlusion maps to train their model in a supervised manner. MODOF [47] utilized discrete-continuous optimization to minimize an energy function.

On the MPI-Sintel dataset, it is difficult to learn occlusion maps in an unsupervised way, since occlusions often occur in untextured regions with limited guidance by the photometric loss [16]. However, we use forward-backward flow and occlusion symmetry relationship to allow them to leverage each other, which not only couples optical flow with occlusion, but also allows to exploit the geometric and temporal information. With this contribution, our model has better results than previous unsupervised methods, and obtains F-measures of 0.55 and 0.49 respectively on the Clean and Final pass. On the Sintel Clean, we have improved at least 1.85% compared to the past best result of OccAwareFlow [14] (0.55 vs 0.54). On the Sintel Final, our result is also the most accurate and is even close to the best result of the supervised model S2D (0.49 vs 0.57 [46]). The results illustrate that the occlusion map is an important clue to estimate optical flow.

On the KITTI dataset, the occlusion masks only contain pixels moving out of the image [16]. The F-measure of ours is 0.98 and 0.93 on the KITTI 2012 and the KITTI 2015 respectively, the improvement reaches to 3.16% (0.98 vs 0.95 [14]) and 2.20% (0.93 vs 0.91 [16]) compared to the state-of-the-arts. It can be seen that our model can accurately extract occlusion information and provide accurate occlusion clues when estimating optical flow, which is beneficial for improving the performance of the CNN-based optical flow model when meets occlusion.

Ablation analysis is further conducted on the FlyingChairs and MPI-Sintel datasets to show the effectiveness of the occlusion reasoning of our method, see Table XII. If we only utilize photometric as the loss function, which means setting forward and backward occlusion maps to zero and without using the non-local term. We get an EPE=5.21 on the FlyingChairs testing set, an EPE=5.88 on the Sintel Clean training set, and an EPE=6.34 on the Sintel Final training set. Remarkably, if we add the occlusion reasoning to the baseline network, the performance is significantly boosted, where the accuracy improvement reaches to 26.46% (4.12 vs 5.21) on the FlyingChairs testing set, 17.13% (5.02 vs 5.88) on the Sintel Clean training set, and 21.46% (5.22 vs 6.34) on the Sintel Final training set respectively. It can be seen that the occlusion mask is helpful for optical flow estimation, and integrating occlusion reasoning is effective to enhance the ability of the model to deal with occlusion. Besides, using occlusion reasoning will be more helpful when there is more ambiguity in the dataset, e.g., Sintel Final. When the occlusion reasoning is not integrated into the model, the model runs at about 27.5 fps on the Sintel dataset, where the computational

TABLE XI
OCCLUSION ESTIMATION EVALUATION. THE RESULTS ARE THE MAXIMUM F-MEASURE. (-) DENOTES THAT THE ORIGINAL PAPER DOES NOT GIVE A CORRESPONDING REPORT. ALL RESULTS ARE FROM THE TRAINING SET OF THE CORRESPONDING DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Methods	Sintel Clean	Sintel Final	KITTI 2012	KITTI 2015
S2D [46]	-	**0.57**	-	-
MODOF [47]	-	0.48	-	-
OccAwareFlow [14]	0.54	0.48	0.95	0.88
MultiFrameOccFlow [16]	0.49	0.44	-	0.91
Our	**0.55**	0.49	**0.98**	**0.93**

TABLE XII
THE EFFECT OF NON-LOCAL AND OCCLUSION ON RESULTS, $\surd$ DENOTES THE COMPONENT THAT WE ADDED, WHILE $\times$ MEANS NOT ADDED. THE MEASUREMENT IS EPE AND THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Occlusion Reasoning	Non-local Term	Chairs test	Sintel Clean train	Sintel Final train
$\times$	$\times$	5.21	5.88	6.34
$\surd$	$\times$	4.12	5.02	5.22
$\times$	$\surd$	4.08	4.75	4.87
$\surd$	$\surd$	**2.98**	**3.68**	**4.12**

time is reduced just by 1.8% (27.5 fps vs 28 fps, see the result in subsection *E*). This is because the consistency check equation does not contain any trainable parameters, therefore the time requirement for occlusion handling is quite small.

VI. CONCLUSION

In this paper, we proposed an end-to-end unsupervised method to learn optical flow from unlabeled images. Our model can automatically learn forward and backward optical flow as well as occlusion maps without using any human annotations. With the captured occlusion information, we design a new loss function to handle the issue of occlusion. Furthermore, we exploit a CNN-based non-local term to refine optical flow during the learning process, which is able to alleviate the boundary over-smoothing effectively. Since our non-local term is based on deep learning, the weights can be learned automatically, making it adaptable to various complex image sequences. The state-of-the-art experimental results demonstrate the effectiveness of the CNN-based non-local term and the loss function.

ACKNOWLEDGMENT

The work was supported by the National Key Research and Development Program of China (No. 2018YFB2100603), the Wuhan University-Infinova project (No. 2019010019), and the Natural Science Fund of Hubei Province (No. 2017CFB598).

REFERENCES

[1] Z. Tu, W. Xie, Q. Qin, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream cnn: Learning representations based on human-related regions for action recognition," *Pattern Recognition*, vol. 79, pp. 32–43, 2018.

[2] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, and J. Yuan, "Action-stage emphasized spatio-temporal vlad for video action recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2799–2812, 2019.

[3] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "A survey of variational and CNN-based optical flow techniques," *Signal Processing: Image Communication*, vol. 72, pp. 9–24, 2019.

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[5] 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*, 2017, pp. 2462–2470.

[6] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[7] T.-W. Hui, X. Tang, and C. Change Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8981–8989.

[8] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2432–2439.

[9] Z. Tu, N. Van Der Aa, C. Van Gemeren, and R. C. Veltkamp, "A combined post-filtering method to improve accuracy of variational optical flow estimation," *Pattern Recognition*, vol. 47, no. 5, pp. 1926–1940, 2014.

[10] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*, 2012, pp. 611–625.

[11] A. Ahmadi and I. Patras, "Unsupervised convolutional neural networks for motion estimation," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1629–1633.

[12] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *European Conference on Computer Vision*, 2016, pp. 3–10.

[13] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4884–4893.

[15] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *European Conference on Computer Vision*, 2018, pp. 690–706.

[17] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 60–65.

[18] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1890–1899.

[19] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[20] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer vision and image understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[21] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[22] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *IEEE International Conference on Computer Vision*, 2013, pp. 1385–1392.

[23] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1164–1172.

[24] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical dense optical flow estimation with occlusions detection," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 371–385, 2007.

[25] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170.

[26] J. Hur and S. Roth, "Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation," in *IEEE International Conference on Computer Vision*, 2017, pp. 312–321.

[27] M. Neoral, J. Šochman, and J. Matas, "Continual occlusions and optical flow estimation," in *Asian Conference on Computer Vision*, 2018, pp. 159–174.

[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[29] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[30] J. Xu, R. Ranftl, and V. Koltun, "Accurate optical flow via direct cost volume processing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1289–1297.

[31] C. Strecha, R. Fransens, and L. Van Gool, "A probabilistic approach to large displacement optical flow and occlusion detection," in *International Workshop on Statistical Methods in Video Processing*, 2004, pp. 71–82.

[32] A. Ayvaci, M. Raptis, and S. Soatto, "Occlusion detection and motion estimation with convex optimization," in *Advances in Neural Information Processing Systems*, 2010, pp. 100–108.

[33] D. Sun, E. B. Sudderth, and M. J. Black, "Layered image motion with explicit occlusions, temporal consistency, and depth ordering," in *Advances in Neural Information Processing Systems*, 2010, pp. 2226–2234.

[34] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *European Conference on Computer Vision*, 2010, pp. 438–451.

[35] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[36] C. Vogel, S. Roth, and K. Schindler, "An evaluation of data costs for optical flow," in *German Conference on Pattern Recognition*, 2013, pp. 343–353.

[37] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European Conference on Computer Vision*, 1994, pp. 151–158.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[39] R. Urtasun, P. Lenz, and A. Geiger, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[40] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.

[41] F. Güney and A. Geiger, "Deep discrete flow," in *Asian Conference on Computer Vision*, 2016, pp. 207–224.

[42] D. Gadot and L. Wolf, "Patchbatch: A batch augmented loss for optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4236–4245.

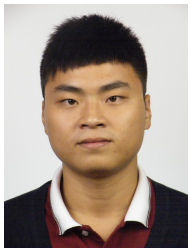
[43] C. Bailer, K. Varanasi, and D. Stricker, "Cnn-based patch matching for optical flow with thresholded hinge embedding loss," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3250–3259.

[44] J. Wulff and M. J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 120–130.

[45] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[46] M. Leordeanu, A. Zanfir, and C. Sminchisescu, "Locally affine sparse-to-dense matching for motion and occlusion estimation," in *IEEE International Conference on Computer Vision*, 2013, pp. 1721–1728.

[47] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.



Long Tian received the master degree in the College of Information and Engineering at Sichuan Agricultural University, China, June 2020. Now he is pursuing his doctorate at Queen Mary University of London, UK. He is a member of the IEEE and China Society for Industrial Applied Mathematics(CSIAM), also a reviewer of IEEE ACCESS and TMCE 2020. His research interests include natural language processing, computer vision, image processing and machine learning. Specially for optical flow, motion estimation and stereoscopic vision.

Zhigang Tu started his Master Degree in image processing at the School of Electronic Information, Wuhan University, China, 2008. In 2015, he received the Ph.D. degree in Computer Science from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. Then from 2016 to 2018, he was a research fellow at the School of EEE, Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Specially for motion estimation, object segmentation, action recognition and localization, and anomaly detection.

Dejun Zhang received the Ph.D. degree from the department of computer school, Wuhan University, China, in 2015. He is currently a lecturer with the School of Geography and Information Engineering, China University of Geosciences, China. Since 2015, he has been serving as a senior member of the China Society for Industrial and Applied Mathematics (CSIAM) and a committee member of the geometric design & computing of CSIAM. His research areas include computer graphics, computer vision, image and video processing, natural language processing.

Baoxin Li received the PhD degree in electrical engineering from the University of Maryland, College Park, in 2000. He is currently a full professor and the Chair of computer science and engineering with Arizona State University, Phoenix, US. From 2000 to 2004, he was a senior researcher with SHARP Laboratories of America, Camas, WA, where he was the technical Lead in developing SHARP's HiIMPACT Sports technologies. From 2003 to 2004, he was also an adjunct professor with the Portland State University, Portland, OR. His current research interests include computer vision and pattern recognition, image/video processing, and multimedia. He won the SHARP Laboratories' President Award twice, in 2001 and 2004. He also received the SHARP Laboratories' Inventor of the Year Award in 2002. He received the National Science Foundation's CAREER Award from 2008 to 2009. He is a senior member of the IEEE.

Junsong Yuan (M'08–SM'14) received his Ph.D. from Northwestern University and M.Eng. from National University of Singapore. He is currently an associate professor at Computer Science and Engineering department of State University of New York at Buffalo. Before that, he was an associate professor at Nanyang Technological University (NTU), Singapore. His research interests include computer vision, video analytics, gesture and action analysis. He received best paper award from Intl. Conf. on Advanced Robotics (ICAR'17), 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'09), and outstanding EECS Ph.D. Thesis award from Northwestern University.

He is currently Senior Area Editor of Journal of Vis. Communications and Image Repres. (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP) and IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT). He is Program Co-chair of ICME'18, and Area Chair of CVPR'17/19/20, ACM MM'18, ICPR'18, ICIP'18'17, ACCV'18'14 etc. He is a Fellow of International Association of Pattern Recognition (IAPR).