Neurocomputing 471 (2022) 346–357

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

FPCR-Net: Feature pyramidal correlation and residual reconstruction for optical flow estimation

Xiaolin Song [a], Yuyang Zhao [a], Jingyu Yang [a,*], Cuiling Lan [b], Wenjun Zeng [b]

[a] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, PR China
[b] Microsoft Research Asia, Beijing 100080, PR China

ARTICLE INFO

Article history:
Received 4 July 2020
Revised 26 August 2021
Accepted 14 November 2021
Available online 25 November 2021
Communicated by Zidong Wang

Keywords:
Pyramid correlation mapping
Contextual residual reconstruction
Optical flow estimation
Deep learning

ABSTRACT

Optical flow estimation is a challenging problem in the field of video analytics yet. Features of different semantics levels in a convolutional neural network (CNN) provide information of different granularity. To exploit such flexible and comprehensive information, we propose a Feature Pyramidal Correlation and Residual Reconstruction Network (FPCR-Net) for optical flow estimation from frame pairs. It consists of two main modules: pyramid correlation mapping and residual reconstruction. The pyramid correlation mapping module takes advantage of the multi-scale correlations of global/local representation by aggregating features of different scales to form a multi-level cost volume. The residual reconstruction module aims at reconstructing the sub-band high-frequency residuals of finer optical flow at each stage. Based on the pyramid correlation mapping, we further propose a correlation-warping-normalization (CWN) module to efficiently exploit the correlation dependency. Furthermore, considering the characteristics of flow warping and alignment, we integrate unsupervised and supervised losses to explore the implicit relevance and explicit constraint. Experimental results show that the proposed network achieves the promising performance for two-frame-based optical flow estimation on the challenging Sintel and KITTI 2012/2015 datasets.

1. Introduction

Optical flow estimation is an important yet challenging problem in the field of video analytics. Recently, deep learning based approaches have been extensively exploited to estimate optical flow. Despite the great efforts and rapid developments, the advancements are not as significant as those achieved in static image based computer vision tasks. The main reason is that optical flow is not directly measurable in the wild and it is challenging to model motion dynamics with pixel-wise correspondence between two consecutive frames. Optical flow estimation requires the deep learning model to be trained with more samples consisting of different displacements, and most importantly formulated with more effective architecture. To increase the utilization of training data, we propose a novel loss function to incorporate both the supervision and unsupervision. The supervised training ensures reliable supervision for learning features, while the unsupervised design reduces dependencies of the network on the ground-truth optical flow and is helpful for quick convergence from a few of training pairs.

Conventional methods attempts to propose mathematical algorithms of optical flow estimation such as EpicFlow [1] by matching features of two frames. However, these methods are complex with high computational complexity, and usually cause failure cases for motions with large displacements and details. Convolutional neural networks (CNNs) like FlowNet [2], SpyNet [3], PWC-Net [4] advance the state-of-the-art performance over conventional methods, with effective structures for feature correlation exploration and warping. Most of these models, however, suffer from difficulties in model training and have limitations, such as lacking joint feature correlation modeling for overall performance or residual learning for high-frequency details. Moreover, limited by the network designs, most of those approaches cannot leverage detail refinement for performance improvement.

In this paper, an end-to-end architecture is proposed with joint and effective feature-wise pyramid correlation representation and residual learning at each stage, which is capable of exploring dynamics at different granularities of consecutive frames. As shown in Fig. 1, the proposed pyramid correlation mapping explores correlations from multi-level features, and jointly embeds

* Corresponding author.
E-mail addresses: songxl@tju.edu.cn (X. Song), yuyangzhao@tju.edu.cn (Y. Zhao), yjy@tju.edu.cn (J. Yang), culan@microsoft.com (C. Lan), wezeng@microsoft.com (W. Zeng).

https://doi.org/10.1016/j.neucom.2021.11.037
0925-2312/© 2021 Elsevier B.V. All rights reserved.

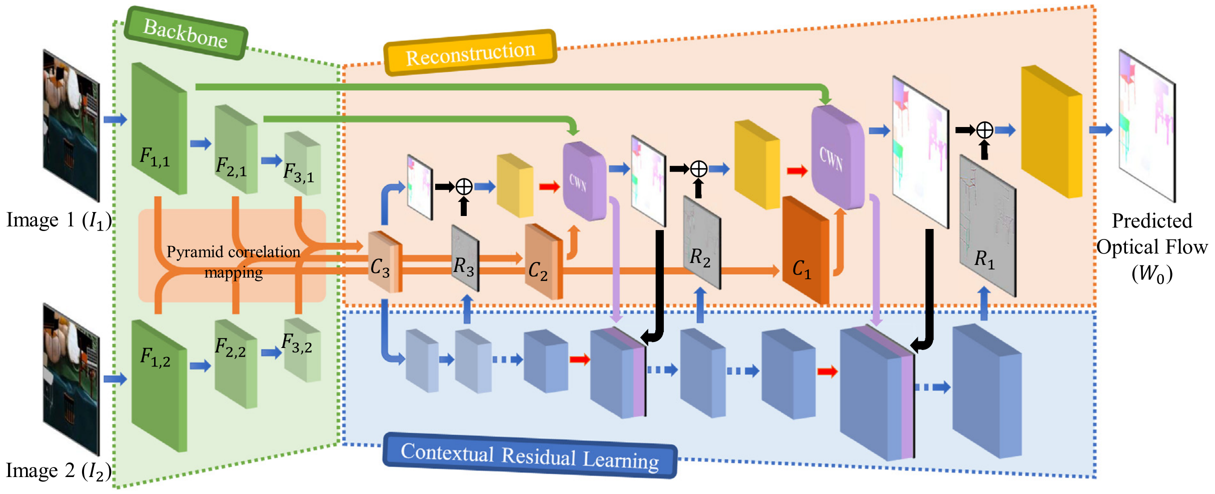


Fig. 1. The overall framework with our Feature Pyramid Correlation and Residual Learning Network (FPCR-Net) for optical flow estimation, to explore dynamics at different granularities (only a 3-stage design is shown). The reconstruction branch yields the correlation features $\mathscr{C}_k$ from the backbone which are optimized by the Correlation-Warping-Normalization (CWN) module (the purple volumes in the reconstruction branch), and the generated optical flow is refined by residual features R_k from the residual learning branch. The blue and red arrows are convolution and transposed convolution operations, respectively.

the cost volumes. Cost volumes are calculated from correlation operation in different scales to measure the effect of correspondence relationships and to capture more edge or texture details of correlation operation. The contextual residual reconstruction module is used to refine optical flow details with the context for multi-stage optical flow estimation networks, which predicts the residuals through a coarse-to-fine architecture. In addition, we present a correlation-warping-normalization (CWN) module for pyramid feature warping and normalization of the input frames. Based on the pyramid representations of the multi-level correlation features, it is essential to use an efficient structure, *i.e.* CWN module. CWN deploys features from the backbone to calculate multi-level cost volumes, which are multi-correlation learning to estimate flow accurately, especially for small displacements.

To summarize, the main contributions of this work are threefold:

- We propose a pyramid correlation mapping operation for jointly exploring the cost volumes in different scales, to better capture detailed motion information from multi-scale features. Besides, we present a CWN module to fuse features from the backbone and cost volumes in the reconstruction branch, to refine intrinsic estimated flow by optimizing warped feature and cost volume.
- We propose a contextual residual learning branch for optical flow reconstruction, which leverages sub-band high-frequency residuals to help reconstruct fine granularity optical flow details at each stage.
- We propose a novel loss function for optical flow estimation, which uses supervised as well as unsupervised learning cues to achieve better performance. Besides, we introduce the flow regularization into loss function, to preserve the smoothness of flow areas and allow for discontinuities.

2. Related work

Optical flow estimation has made great progress in recent years. Weinzaepfel et al. [5] propose DeepFlow which performs to correlate multi-scale patches to calculate cost volume, and uses a built-in smoothing algorithm on the set of output correspondences. Revaud et al. [1] propose EpicFlow that computes a sparse set of matches between the two images and then interpolates them to dense flow. Bailer et al. [6] propose a dense correspondence field approach with patch matching techniques and multi-scale matching strategy.

Many machine learning techniques are also applied in this task. Sun et al. [7] study both the brightness constancy error and the spatial properties of optical flow in a high-order random field. Rosenbaum et al. [8] model the local statistics of optical flow using Gaussian mixtures. Wulff and Black [9] propose PCA-Flow that utilizes a set of basis flow fields to estimate optical flow from sparse feature matches. Nir et al. [10] represent image motions by over-parametrization model. However, these methods are limited with the large number of coefficients and heavy computational complexity.

With the development of deep learning, CNNs provide better performance with deep discriminative features in many computer vision tasks, such as image recognition [11,12] and segmentation [13,14]. Moreover, CNN-based methods also achieve a breakthrough on optical flow estimation. Zagoruyko et al. [15] first introduce CNN-feature matching and then some methods use CNN models for image patches matching in optical flow estimation. Güney et al. [16] utilize feature representation and formulate optical flow estimation in Markov random field. Zweig and Wolf [17] propose InterpoNet which utilizes a convolutional network based sparse-to-dense interpolation to estimate optical flow. Thewlis et al. [18] propose a Deep Matching formulation method as an end-to-end CNN. Gadot [19] and Bailer et al. [20] use patch matching algorithm under Siamese network architectures with heavy computing. Chen et al. [21] use a coarse-to-fine PatchMatch method with sparse seeds of over-segmentation to estimate optical flow. Moreover, patch matching based methods lack the capacity to apply to the larger context of the entire image because of the small image patches operator. Dosovitskiy et al. [2] design FlowNet, which is an important CNN exploration on optical flow estimation with encoder-deconder architecture, of which FlowNetS and FlowNetC are proposed with simple operations. However, the number of parameters is large with heavy calculation on correlation. Ilg et al. [22] propose FlowNet2 with milestone performance, which is a cascaded network based on FlowNetS and FlowNetC. However, it has huge number of parameters and expensive computation complexity.

To improve the performance and save the calculation cost, several methods attempt to balance speed and accuracy for optical flow estimation. Ranjan et al. [3] present a compact network named SPyNet with multi-level representation learning. Nevertheless, the performance is not satisfactory with simple network architecture. To improve the performance of lightweight networks, Hui et al. [23] leverage a compact LiteFlowNet and Sun et al. [4] propose PWC-Net, which both introduce lightweight networks with high accuracy into optical flow estimation task. They utilize light feature-level matching and warping motivated by conventional methods. LiteFlowNet [23] involves cascaded flow inference for flow warping and feature matching, and feature-driven local convolution (f-lconv) for flow regularization. PWC-Net et al. [4] proposed a novel cost volume constructed by feature pyramid extraction and feature warping, and uses context network for optical flow refinement. Chen et al. [24] utilize TV-wavelet regularization to make up for lost motion information. Chen et al. [25] adopt different filtering operations for regularization with respect to consistency. Mei et al. [26] exploits the unequal probability as the weight with non-local information to estimate stable optical flow despite illumination changes. Zhai et al. [27] integrate local features with their global dependencies and focuses on important features and suppresses unimportant spatial features.

Occlusion is one of the key challenges in optical flow estimation. Several methods utilize additional information or novel modules to predict the occlusion map and help to estimate optical flow. MirrorFlow [28] exploits the occlusion-disocclusion symmetry in joint optical flow to deal with occlusion. Hur et al. [29] are based on popular networks, including FlowNet and PWC-Net, and use bilateral filters to refine blurry flow and occlusion. ContinualFlow [30] integrates occlusion estimation into decoder part and incoporates information of past frames to improve flow estimation.

Due to the difficulties of getting ground truth of real-world data, many unsupervised and self-supervised methods are proposed. Wang et al. [31] utilize forward and backward warping to facilitates the learning of large motion. UnFlow [32] introduces a novel bidirection census loss to deal with occlusion and improve the accuracy of unsupervised optical flow estimation. Janai et al. [33] utilize three frames to get temporal constraints, which improve the unsupervised optical flow estimation in occluded regions. SelFlow [34] combines supervised learning and self-supervised learning to enhance multi-frame optical flow estimation.

Inspired from these methods, our method utilizes an encoder-decoder architecture for two-frame based optical flow estimation without the auxiliary of occlusion. To improve the generalization ability and involve flow regularization, our method introduces an unsupervised loss term and a flow penalty loss term. In addition, we propose a contextual refinement branch to learn reconstruction residuals for fine-grained global refinement of flow field estimation.

3. FPCR-Net

Optical flow denotes the dense motions between two consecutive frames, and optical flow estimation aims to estimate the dense optical flow from RGB images. The features of different layers/stages of a convolutional neural network can provide representations of different granularities and details. Fig. 1 shows the overall flowchart of our framework called *Feature Pyramid Correlation and Residual Learning Network* (FPCR-Net) for optical flow estimation.

To exploit the flexible and comprehensive information for motion dynamics in space and time, we propose three key components — *pyramid correlation mapping*, *CWN module embedding*, and *contextual residual learning* in the backbone, the main reconstruction branch, and the residual learning branch, respectively. The encoder-decoder architecture is adapted for optical flow prediction like FlowNetC [2], with the *pyramid correlation mapping* and *CWN module embedding* to explore the representation of multi-level correlation feature efficiently, and the *contextual residual learning branch* utilizing for fine-grained feature reconstruction, as shown in Fig. 1.

3.1. Pyramid correlation mapping

Most CNN-based methods for optical flow estimation use frame feature extractor (backbone) modified from FlowNetC [2] with down-sampling and high-level feature correlation of input frames. Each layer extracts corresponding features to transform the two input frames to pyramidal multi-scale and multi-dimensional representation with shared weights. To reduce the computational complexity and take advantage of multi-level features, inspired by LiteFlowNet [23] and PWC-Net[4], we perform short-range correlation for each pyramid level features, instead of long-range correlation at a single level like FlowNetC. The backbone utilizes a pyramid correlation architecture to extract multi-scale feature maps. Cost volumes at different scales are used for coarse-to-fine optical flow prediction. For the high-level cost volume, large displacements could be represented by features in adjacent pixels, which is essential for challenging cases like motion blur and large deformation. Likewise, low-level features contain detailed information such as edges, shapes, and textures, and small displacements could be more efficiently represented by the low-level cost volume. Moreover, to reduce the burden of correlation, we use an atrous convolution pyramid module with stride=2 and output channels of 16, 32, 64, 96, and 128 in feature extraction for sparse correlation instead of traditional convolution. Besides, we propose a pyramid correlation mapping operation for multi-level cost volume, to yield more edge or texture correlation details for different scales. Fig. 2 indicates the process of the operation. For each frame of an input pair, atrous convolutions with the dilation rates of 1, 2, 4, and 8 are used for extracting feature maps at each convolutional layer to expand the correlation search region and preserve the details of edges and textures.

Let $F_{k,i}$ denote the k-th level feature extracted from the network of input image I_i. Specially, $F_{0,i}$ means the original image I_i. The single correlation at the k-th level is calculated as follows.

$$\mathscr{C}_k^s(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o}} (\mathbf{f}_{k,1}(\mathbf{x}_1 + \mathbf{o}))^\top \mathbf{f}_{k,2}(\mathbf{x}_2 + \mathbf{o}), \tag{1}$$

where $\mathbf{o}$ denotes the offset of correlation operation, and $\mathbf{o} \in [-n, n] \times [-n, n]$ for search region. $\mathbf{f}_k$ is the flattened column vector of F_k. We leverage a pyramid correlation mapping operation based on single correlation for aggregating different level cost volumes. The details of this calculation are shown as:

$$\mathscr{C}_k = \begin{cases} \mathscr{C}_1^s, & k = 1, \\ \mathscr{C}_k^s \odot (\mathscr{C}_{k-1} \Downarrow), & k > 1, \end{cases} \tag{2}$$

where $\Downarrow$ is the down-sampling operator with average pooling and channel reduction at the rate of $\frac{1}{2}$, and $\odot$ denotes the concatenating across the channel dimension.

3.2. Correlation-warping-normalization module

Inspired by FlowNet2 [22] and PWC-Net [4], the multi-component warping and correlation module is proposed within the reconstruction branch, *i.e.* Correlation-Warping-Normalization (CWN) Module, shown in Fig. 3. This module fuses the cost volumes from pyramid correlation mapping and involves the normalization operator to feed into multi-layer CNN flow estimator, for refining the up-sampled optical flow with both large and small displacement motion modeling.

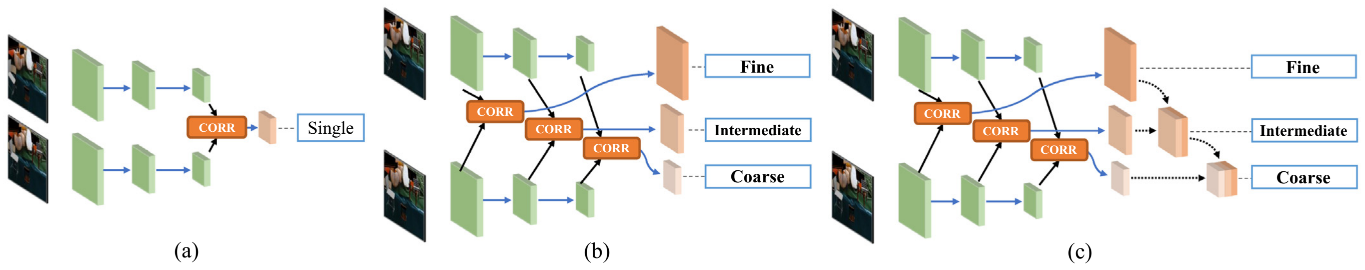


Fig. 2. Illustration of the pyramid correlation mapping operation, which calculates and aggregates multi-scale cost volumes. (a) single correlation of high-level low-resolution features, (b) correlation of multi-level features, (c) correlation and mapping of multi-level features.

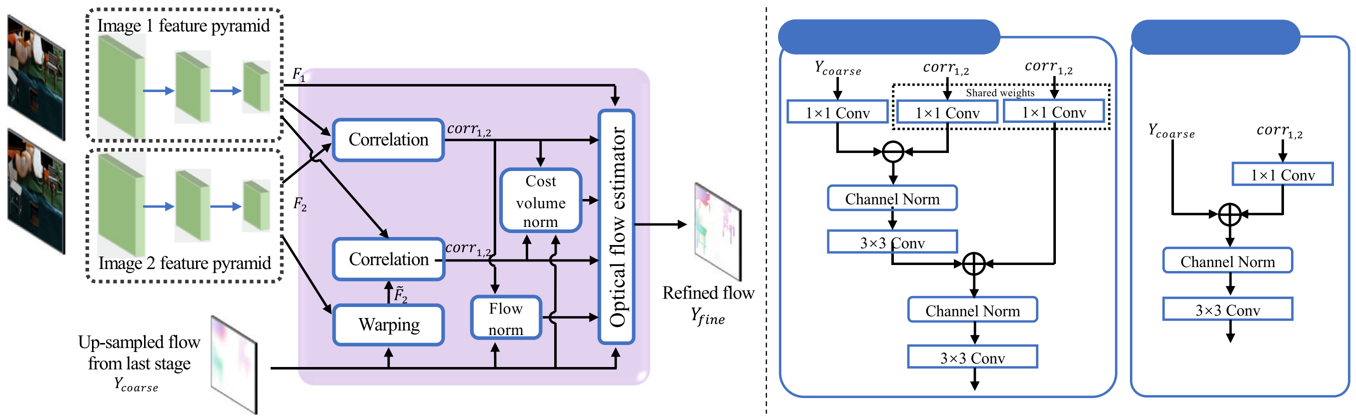


Fig. 3. Details of CWN module with correlation, warping and normalization units.

Correlation and Warping. For stage k of the reconstruction, the features $F_{k,i}$ construct a cost volume which contain the patch-wise matching scores by pyramid correlation mapping $\mathscr{C}_k$. In addition, the feature map $F_{k,2}$ is warped to the view of $F_{k,1}$, denoted by $\tilde{F}_{k,2}$, via the bicubic up-sampling flow from stage $k-1$. Then $\tilde{F}_{k,2}$ is correlated with $F_{k,1}$, and the cost volume contains errors from the coarse features, especially around the edges on I_1, indicating the implicit optimization attention region for auxiliary learning of $\mathscr{C}_k$. Besides, the motion range of this operation is set to a small value of 4, further reducing parameters in correlation mapping.

Normalization. Compared with other normalization methods, channel-wise normalization is much more useful in this task to aid generalization for feature representation. We propose the channel normalization operation, and establish the cost volume normalization and flow normalization for correlation feature fusion and normalization. Denote the feature as V, and the channel normalization defines as:

$$\mathscr{N}(\mathbf{x},c) = V(\mathbf{x},c)/\left(\alpha\sum_{\mathbf{x}}(V(\mathbf{x},c))^2+\epsilon\right)^{\beta}, \tag{3}$$

where c denotes the channel of V. α, β and ϵ denote the multiplier, the exponent and the additive constant with a small value for normalization term, respectively. We use $\alpha = 0.99, \beta = 0.5$ and $\epsilon = 0.01$.

3.3. Contextual residual reconstruction

To explore motion details and occlusion compensation between frames to learn fine-grained residual representation, we construct our network by utilizing the pyramid contextual framework for coarse-to-fine residual learning, as shown in Fig. 1. For stage k, the residual learning branch is independently structured with refining the residual map from stage $k-1$. As shown in Fig. 4 (f), this branch contains the cascades of convolution stack module for refining residual features and one transposed convolutional layer for 2× up-sampling to fuse with the output of CWN module. Compared with the single convolutional layer (Fig. 4 (d)) and the original residual block in ResNet [12] (Fig. 4 (e)), the proposed convolution stack module is a cascade of two parts: 1) the atrous projection block and 2) the single residual block. The atrous projection block utilizes atrous pyramid convolution with dilated rates of 1, 2 and 4, to explore the contextual information with different receptive fields. Due to the similarity of function at each stage, the parameters of the convolution stack module are partially shared except the atrous convolution and the last convolution layer, in order to reduce the parameter number and increase the non-linearity of the network. For the up-sampling layer, as shown in Fig. 4 (a) (b) and (c), we use three schemes of refinement operations with different up-sampling location.

3.4. Training loss function

According to the optical flow characteristics, we proposed a loss function incorporating both supervised and unsupervised constraints. For the k-th level, the total loss function is as follows.

$$\mathscr{L}_{\phi,k} = \mathscr{L}^S_{\phi,k} + \lambda\mathscr{L}^U_{\phi,k} + \eta\mathscr{L}^R_{\phi,k} \tag{4}$$

where ϕ denote the network parameters that predict the optical flow, and $\mathscr{L}^S_{\phi,k}$, $\mathscr{L}^U_{\phi,k}$ and $\mathscr{L}^R_{\phi,k}$ denote the supervised, unsupervised and regularization loss, respectively. Fig. 5 shows the training loss on the original input images I_i (or $F_{0,i}$), ground truth $\hat{W}_0$ and corresponding predicted optical flow W_0. λ and η are the coefficients to balance the three loss term. Actually, the loss is used for multi-stage optical flow prediction with different loss weights as follows.

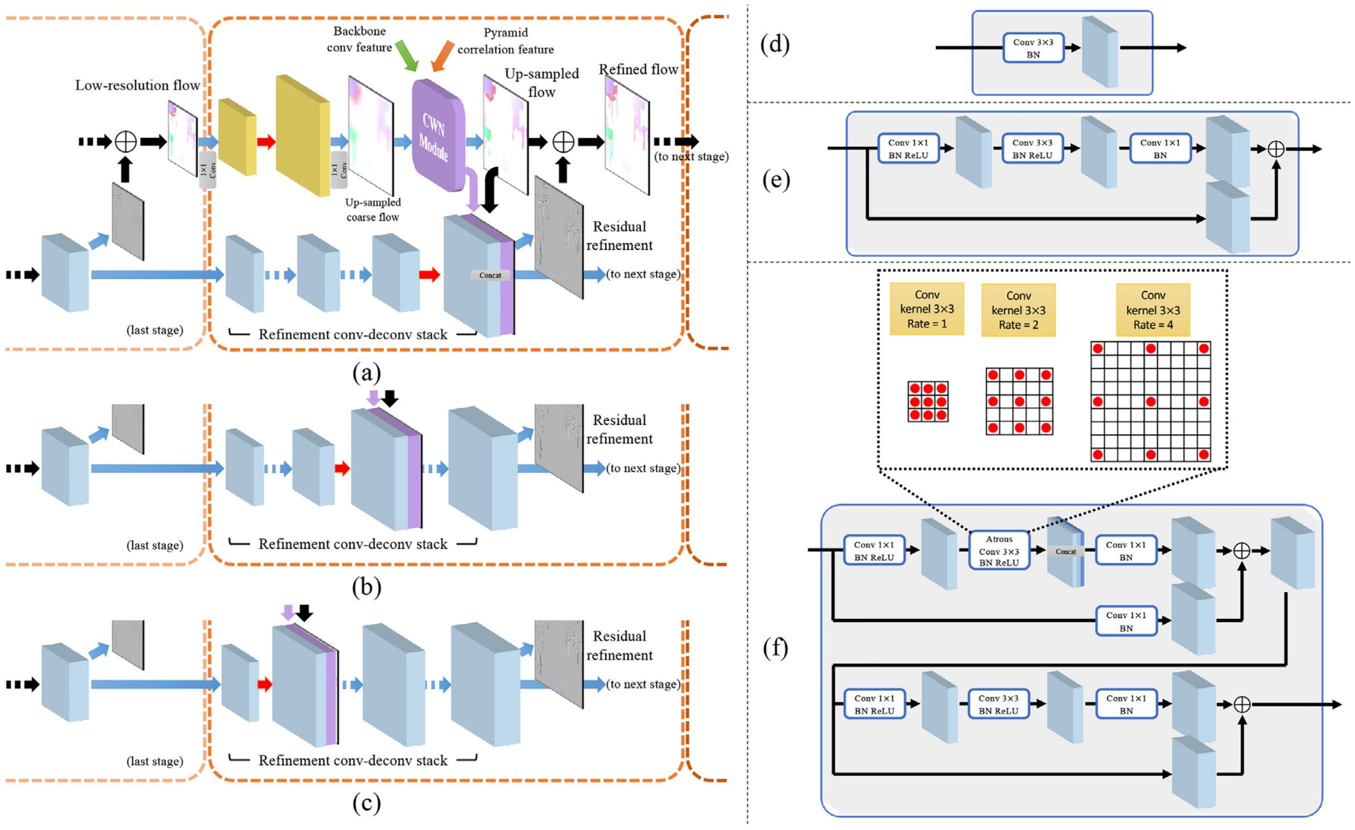


Fig. 4. Details of feature residual learning branch. We compare three types of refinement operations with different up-sampling location. (a) late up-sampling; (b) middle up-sampling; (c) early up-sampling. The red arrows are the transposed convolution operation, and the blue dotted-line arrows are the stacks of convolution unit in (d) 3×3 convolutional layer, (e) the original residual block in ResNet[12], or (f) the proposed atrous residual block, which are stacked by different output channels. for refinement.

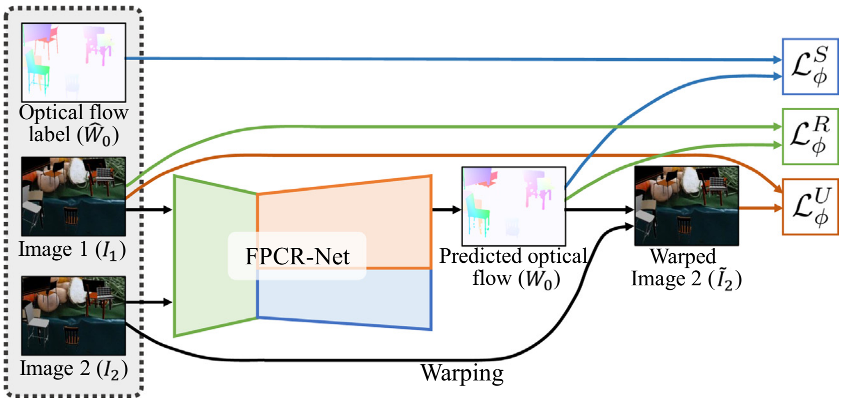


Fig. 5. Illustration of the loss function with multiple losses.

Supervised Loss. The supervised loss term measures the pixel-wise deviation between the predicted optical flow from two frames and the ground truth, and we utilize the end-point error loss as the supervised loss function according to the standard metric on the test of optical flow estimation results. In addition, the smooth-ℓ_1 loss is used for spatial constraint optimization.

$$\mathcal{L}^S_{\phi,k} = \sum_{\mathbf{x}}\left(\sum_{d\in\{u,v\}}\left(W_k(\mathbf{x},d)-\hat{W}_k(\mathbf{x},d)\right)^2\right)^{\frac{1}{2}} + \sum_{\mathbf{x}}\sum_{d\in\{u,v\}}\left|W_k(\mathbf{x},d)-\hat{W}_k(\mathbf{x},d)\right|, \tag{5}$$

where d denotes the direction index to indicate the component of the optical flow (W_k and $\hat{W}_k$), *i.e.*, either the horizontal component, marked by u, or the vertical component, marked by v.

Unsupervised Loss. The unsupervised term of loss function optimizes the direct image or feature alignment error according to data fidelity as follows.

$$\mathcal{L}^U_{\phi,k} = sum_{\mathbf{x}}\left|F_{k,1}(\mathbf{x}) - F_{k,2}(\omega(\mathbf{x},W_k(\mathbf{x})))\right|, \tag{6}$$

where $\omega(\mathbf{x},W_k(\mathbf{x}))$ is the warping function to generate rectified images or features from $F_{k,2}$ to the view of $F_{k,1}$ by predicted optical flow $W_k(\mathbf{x})$. We evaluate the multi-level alignment loss at the values of $F_{k,1}$ and the reconstructed features of $F_{k,2}$ that warp to a valid location into the second image using linear interpolation for subpixel-level warping due to occlusion or overlapping.

Regularization Loss. Note that optical flows are piece-wise smooth signals: neighbouring pixels within an object tend to have similar motions while pixels of different objects tend to have different motions. As a result, optical flow appears as smooth areas separated by curves of object contours, which can be characterized by signal priors such as Markov random field (MRF), and sparse gradient. The gradient sparseness of optical flow has been widely exploited in optical flow estimation based on variational optimization [35,1], and has shown powerful regularization in reconstructing both smooth areas and sharp discontinuities, such as edges and contours. Encouraged by these works, we introduce the prior of gradient sparseness into our training loss. Instead of using the ℓ_1-norm as sparse prior, we introduce the pixel-wise weighting term to suppress large variations in smooth regions while allow sharp transitions around discontinuities of optical flow. To this end, we design the pixel-wise weighting term as the negative exponential of the pyramidal feature gradients, $\exp(-|\nabla F_{k,1}(\mathbf{x})|)$, as the discontinuities of $W_k(\mathbf{x},d)$ are usually consistent with image edges estimated by $|\nabla F_{k,1}(\mathbf{x})|$. Our regularization loss is formulated as the following weighted ℓ_1-norm of optical-flow gradient.

$$\mathscr{L}^R_{\phi,k} = \sum_{\mathbf{x}} \sum_{d\in\{u,v\}} |\nabla W_k(\mathbf{x},d)| \circ \exp(-|\nabla F_{k,1}(\mathbf{x})|), \tag{7}$$

where $\circ$ denotes the Hadamard product operator.

4. Experiments

In this section, we first describe the training and test datasets, and implementation details. Then we study the effects of different factors in our designed network. Finally, we compare our approach with several state-of-the-art approaches.

4.1. Datasets

FPCR-Net is trained on the FlyingChairs dataset [2] and the FlyingThings3D dataset [36]. The FlyingChairs dataset contains about 22 k image pairs and their optical flow of chairs on different background images with only planar motions (translation and rotation). The FlyingThings3D dataset [36] is a 3D-version motion object dataset which consists of 22 k random scenes with 3D models moving in front of static 3D background scenes.

The effectiveness of the proposed framework is validated on the Sintel dataset [37] and KITTI [38,39] datasets, which are popular optical flow estimation benchmarks. The MPI Sintel dataset is a large dataset with dense ground truth for both small and large displacement magnitudes. There are 1,041 training image pairs for two passes — *Clean* and *Final*. The *Final* pass contains motion blur and atmospheric effects, while the *Clean* pass has clear edges and single light environment. The labels of optical flow are acquired from rendered synthetic scenes to realistic image appearance. The KITTI 2012 dataset [38] and the KITTI 2015 dataset [39] contains 394 training pairs totally with large displacements. The ground truth is sparse obtained from real world scenes by capturing the scenes by the RGB camera and LiDAR - a 3D laser scanner simultaneously. Limited to the devices, the labels are sparse and captured from only street scene.

4.2. Data augmentation

We use the common augmentations including geometric transformations (translation, rotation and scaling), dynamic changes in low-level attributes (brightness, contrast, gamma and color) and Gaussian noise injection. In addition, we adapt two special schemes for data augmentation.

Motionless Injection. Considering the dynamic range of predicted optical flow and the complexity of the network, our proposed network are trained by including motionless frames with all-zero optical flow maps, to reduce the jitters of the predicted flow. 3 k training images are selected randomly from both FlyingChairs dataset and FlyingThings3D dataset, and each image is filled into both the first and the second frame. Naturally, the optical flow map is all-zero filling.

Motion Reversal. To improve the utilization of the training data, we take advantage of the dataset containing the optical flow labels referred to both the previous frame (forward flow) and the next frame (backward flow), *i.e.* FlyingThings3D dataset. Generally, the frame pair and the forward optical flow are used as the training data. Besides, we reverse the frame pairs with the opposite number of the backward optical flow as the augmented training data.

As shown in Table 1, both of the two schemes improve the performance on FlyingChairs and FlyingThings3D datasets. The results illustrate that motion augmentation is useful in model training of optical flow estimation task.

4.3. Implementation details

Training from Scratch. According to the training strategy in FlowNet2, the same loss weights are used among stages with 0.32, 0.08, 0.02, 0.01, 0.005. At each stage, the trade-off weights λ and η of loss function $\mathscr{L}_{\phi,k}$ are set to 0.05 and 0.005, respectively. We train our network by the following steps: 1) The backbone and the main branch of reconstruction are trained firstly for 120 k epochs using the learning rate schedule S_{long} for 150 k epochs on the FlyingChairs dataset, and fine-tuning on the FlyingThings3D dataset by the S_{fine} learning rate schedule for 125 k epochs introduced in [22] [1]. 2) The residual learning branch is trained after the last step with fixed parameters of the backbone and the main branch for 100 k epochs using the same schedule as step 1. 3) The whole network is trained without fixed parameters followed by step 2 using the S_{fine} learning rate schedule on the FlyingThings3D dataset. We scale the ground truth flow by 20 for easy training and downsample it as the supervision labels at different levels. Experiments are conducted with the batch size of 8 on two NVIDIA GTX 1080 Ti GPUs (11 GB$\times$2).

Fine-tuning. The object and motion are not comprehensive in each task and it is not quite sufficient to transfer the model to another dataset with different scene and movable objects. We fine-tune the networks on the target datasets for better performance, *e.g.* the Sintel dataset and the KITTI datasets, with the learning rate schedule S_{long} for 100 k epochs. The networks are fine-tuned with a batch of 8 on two NVIDIA GTX 1080 Ti GPUs as well.

4.4. Ablation study

In this subsection, we will analyze the effectiveness of the proposed pyramid correlation mapping component and the CWN module, and discuss the architecture design of residual learning, respectively.

Effectiveness of Pyramid Correlation Mapping. Aggregation of multi-level correlated cost volume provides the opportunity to jointly explore the motion details at the video level. The performance of our scheme in comparison with the baseline scheme on the Sintel training *Clean* dataset is shown in Table 2. As the results indicated, the "pyramid correlation mapping" scheme, with multi-level respective fields, achieves 0.20 and 0.32 decrease in average end-point error (AEE) compared with baseline measured in Sintel training *clean* trained by the FlyingChairs dataset and fine-tuned by the FlyingThings3D dataset.

Effectiveness of CWN Module. The CWN module fuses correla-

[1] S_{long} schedule starts from learning rate of 10^{-4} and reduces the learning rate by 0.5 at 0.4 M, 0.6 M, 0.8 M, and 1 M iterations, and then learning rate of S_{fine} schedule starts at 1.2 M iterations from 10^{-5} and reduces at 1.4 M, 1.5 M, 1.6 M iterations.

Table 1
Comparison of augmentation with different schemes on the Sintel training *Clean* pass. The network is trained on FlyingChairs and fine-tuned on FlyingThings3D. “MI” and “MR” denote *Montionless Injection* and *Motion Reversal* in data augmentation, respectively.

Dataset	w/o MI/MR	w/ MI	w/ MR	w/ MI + MR
FlyingChairs	4.23	4.13	3.91	**3.82**
FlyingThings3D	3.34	3.27	3.17	**3.11**

Table 2
Comparison of training pyramid correlation mapping with different correlation schemes in Fig. 2. Numbers indicate the AEE on Sintel training *Clean*. “sc.” “pc.” and “pcm.” denotes the three correlation schemes — single correlation, pyramid correlation and pyramid correlation mapping. The network is trained on the FlyingChairs dataset first and fine-tuning on the FlyingThings3D dataset.

Dataset	sc.	pc.	pcm.
FlyingChairs	3.82	3.67	**3.62**
FlyingThings3D	3.11	2.88	**2.79**

Table 4
Comparison of residual learning branch with different architectures in Fig. 4 on Sintel training *Clean*. the early-upsampling scheme performs better than the others with FlyingThings3D fine-tuned.

Dataset	late up-sampling	mid. up-sampling	early up-sampling
FlyingChairs	3.58	**3.49**	3.50
FlyingThings3D	2.87	2.76	**2.69**

tion features from pyramid correlation mapping and the coarse flow from the last stage efficiently for accurate flow estimation. We show the performance of architectures with different components and correlation features from pyramid correlation mapping in Table 3. The entire CWN module with pyramid correlation mapping gets the better performance and each component makes contributions to the network with more correlation details, less warping error and accurate relative values.

Comparisons on Residual Learning Branch Designs. We have designed a residual learning branch for finer optical flow reconstruction. This branch consists of a transposed convolution layer and convolution units in Fig. 4(f)(f) to explore motion details and occlusion compensation between frames to learn fine-grained residual representation. We have tried three network structures with different locations of up-sampling shown in Fig. 4(a)(b)(c). The numbers of convolution unit output channels are 64, 64, 128 and 256 in the entire branch, respectively. We show the results of these structures in Table 4 with experiments conducted on the Sintel training *Clean* pass. As the results indicated, the early up-sampling scheme achieves better performance than others with detail-preserving flow field learned from large-scale features. To evaluate the effectiveness of the residual branch, we have also compared our residual branch with different units, *i.e.*, the 3×3 convolutional layer, the original residual block, and the proposed atrous residual block in Fig. 4 (d) (e) and (f), respectively. As shown in Fig. 6 (b) and (c), the residual branch significantly improves the performance by leveraging high-frequency details. In addition, as shown in Table 5, compared with the original residual block shown in Fig. 6 (d), our proposed residual branch helps to obtain more accurate and informative residual maps by large receptive fields, and thus achieve higher prediction accuracy in terms of AEE.

Comparisons on Different Loss Combinations. We design the loss function for self-learning and regularization. Table 6 compares the loss of different components. Although utilizing unsupervised term only performs poorly, combining it with supervised term could indeed improve the performance. In addition, the regularization loss decreases the AEE by 0.38 on Sintel *Final* pass (S + U+R *vs.* S + U), while the training time only increases by 3.26% (1330 s/epoch for S + U+R *vs.* 1288 s/epoch for S + U). Moreover, the all-term utilized loss achieves the best performance on Sintel training *Clean* and *Final* passes. The results suggest that it is effective to utilize different types of loss term for parameter optimization.

Furthermore, to verified the effectiveness with unlabeled data, UCF101 dataset [40] is used for training without the supervised loss term. UCF101 is a human action recognition dataset without optical flow labels, including 13,320 video clips. Besides, the unlabeled target dataset (*i.e.* Sintel training set) is used as well. As shown in Table 7 the results show that the proposed loss with extra unlabeled data are effective for optimizing the network parameters by the implicit characteristics with warping and alignment. With the large number of the unlabeled training data, the training scheme with UCF101 achieves better performance, which decreases the AEE by 0.11 and 0.08 on Sintel training *Clean* and *Final*, respectively. The unlabeled Sintel scheme is also useful because of the objective distribution in the target feature space, which decreases the AEE by 0.06 and 0.05, respectively.

Effectiveness of Different Component Options. We explore the contribution of each component option by calculating the AEE with some of the components enabled or disabled in Table 8. The training time increases when different components are utilized with more comprehensive computation, and the training time of the full network increases by 27.5% against the baseline. Fig. 7 illustrates the component-accumulated examples of flow fields on the Sintel dataset. We can see that the small-magnitude artifacts are restrained and the smaller AEE is achieved with the components accumulated. In addition, we extract the features of warped image $\tilde{I}_2$ for unsupervised loss term and learned residual details at the last stage. By involving these useful items, the network addresses accurate optical flow with detail-preserving.

Table 3
Comparison of training CWN with different components on AEE on Sintel training *Clean*. “C-W” denotes the correlation and warping module and “C-W-N” denotes the entire CWN module.

Dataset	C-W	C-W-N w/ sc.	C-W-N w/ pc.	C-W-N w/ pcm.
FlyingChairs	3.54	3.50	3.44	**3.41**
FlyingThings3D	2.91	2.87	2.80	**2.66**

4.5. Runtime and computational complexity

For a fair comparison, we measure the runtime of different CNN methods with Intel Core i5 CPU and NVIDIA GTX 1080 Ti GPU. Timings are averaged over 100 runs for images in Sintel of size 1024×436. As summarized in Table 9, our method is 2 times faster than FlowNet2, and 1.5 times faster than LiteFlowNet. Due to the modules of pyramid correlation mapping and contextual residual reconstruction, our method is slower than PWC-Net. However, our method models the high-frequency information and preserves

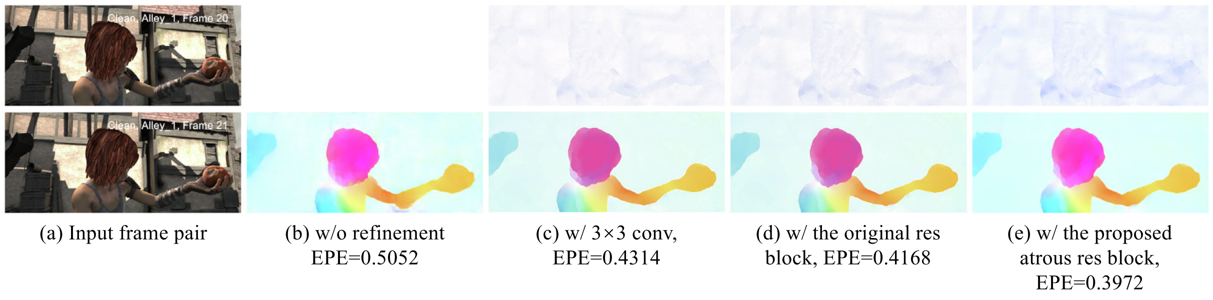

Clean, Alley_1, Frame 20
Clean, Alley_1, Frame 21
(a) Input frame pair
(b) w/o refinement EPE=0.5052
(c) w/ 3×3 conv, EPE=0.4314
(d) w/ the original res block, EPE=0.4168
(e) w/ the proposed atrous res block, EPE=0.3972

Fig. 6. Examples of learned residual maps and predicted flow maps from residual reconstruction refinement with different units. Our proposed atrous convolution scheme achieves the better performance. (Zoom in for details.).

Table 5
Comparison of residual learning branch with different units on Sintel training *Clean* and *Final* passes.

Dataset	w/o Refine.	w/ 3×3 Conv.	w/ Original Res Block	w/ Atrous Res Block
Sintel training *Clean*	2.66	2.43	2.35	**2.24**
Sintel training *Final*	3.70	3.59	3.57	**3.50**

Table 6
Comparison of different terms of loss function in Eq. 4 on Sintel training *Clean* and *Final* passes. The network is trained on FlyingChairs and fine-tuned on FlyingThings3D. “S” “U” and “R” denote Supervised, Unsupervised and Regularization terms in training loss, respectively.

Dataset	U*	S	S + U	S + R	S + U+R
Sintel training *Clean*	15.83	2.44	2.33	2.38	**2.24**
Sintel training *Final*	18.27	4.13	3.88	3.95	**3.50**
Training time (s/epoch)	1034	1183	1288	1222	1330

* “U” is trained on FlyingChairs with labels and finetuned by the unsupervised term with unlabeled FlyingThings3D. It would be not converged to train from scratch with all unlabeled FlyingChairs and FlyingThings3D.

Table 7
Comparison of training with/without extra unlabeled data on Sintel training *Clean* and *Final* passes. The unlabeled data is used for training with labeled FlyingChairs and FlyingThings3D. The values in parentheses are the results of the networks on the data they were trained on.

Dataset	w/o unlabeled data	w/ unlabeled UCF101	w/ unlabeled Sintel
Sintel training *Clean*	2.24	**2.13**	(2.18)
Sintel training *Final*	3.50	**3.42**	(3.45)

Table 8
Ablation study of our component choices of the network. Average end-point error Results of our FPCR-Net with different components of pyramid correlation mapping, CWN module and early up-sampling residual learning branch on Sintel training *Clean* and *Final* passes. The values in parentheses are the results of the networks on the data they were trained on.

Baseline	√	√	√	√	√	√
Pyramid corr.	×	√	√	√	√	√
CWN module	×	×	√	√	√	√
Residual reconstruction	×	×	×	√	√	√
Finetune	×	×	×	×	w/o label	w/ label
Sintel training *Clean*	3.11	2.78	2.66	**2.24**	(2.18)	**(1.58)**
Sintel training *Final*	4.57	3.82	3.70	**3.50**	(3.45)	**(1.97)**
Training time (s/epoch)	1043	1149	1227	1330	–	–

more details via the specific modules for accurate optical flow estimation.

4.6. Hyper-parameter analysis

We integrate three loss terms in FPCR-Net — supervised term, unsupervised term and regularization term. Empirically, regularization term is used to smooth the variation. Unsupervised part could optimize the model without annotation, while this term would involve artifacts with the large displacement and is difficult to converge. Consequently, we set the trade-off weights of unsupervised term and regularization term as smaller values. In Table 10, we compare individual contributions of different loss terms, demonstrating that $\lambda = 0.05$ and $\eta = 0.005$ is the best choice.

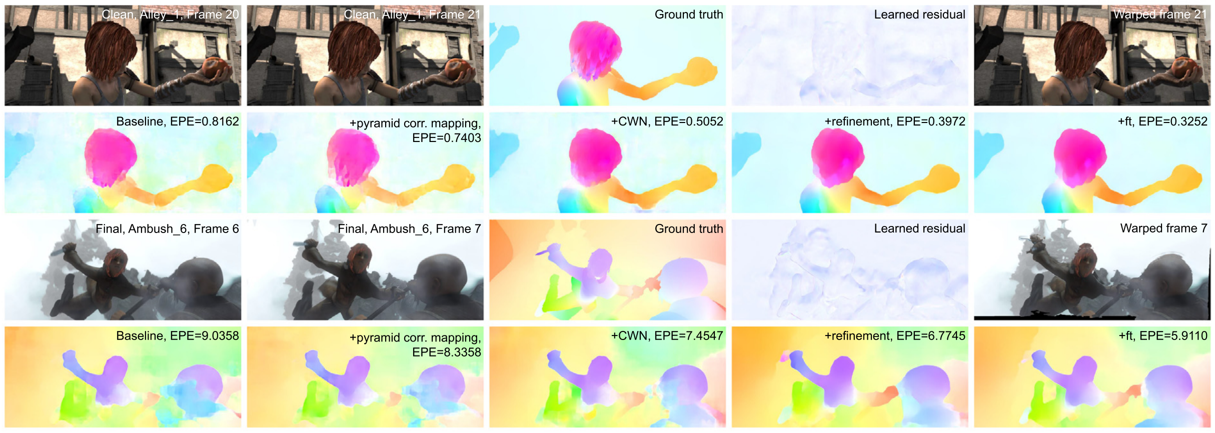


Fig. 7. Results on Sintel training *Clean* and *Final* passes. Pyramid correlation mapping, CWN module and residual learning all improve the performance. And we indicate the learned residual and warped I_2 at the top stage. (Zoom in for details.).

Table 9
Comparison of the runtime of different CNN methods (Inference on Intel Core i5 and NVIDIA GTX 1080 Ti).

Methods	No. of param. (M)	Runtime (ms)	Frame rate (fps)
FlowNetC [2]	39	38.8	26
FlowNet2 [22]	163	105.5	10
LiteFlowNet [23]	5.4	72.1	14
PWC-Net [4]	8.8	37.2	27
FPCR-Net(Ours)	12.4	45.6	22

Table 10
Comparison of different coefficients of loss terms on Sintel training *Clean* and *Final* passes. The network is trained on FlyingChairs and fine-tuned on FlyingThings3D.

λ	0	0.1	0.05	0.01	0	0	0
η	0	0	0	0	0.01	0.005	0.001
Sintel training *Clean*	2.44	2.38	2.33	2.34	2.40	2.38	2.43
Sintel training *Final*	4.13	3.93	3.88	3.90	3.98	3.95	4.04

4.7. Comparison with state-of-the-art methods

We compare our proposed scheme with state-of-the-art approaches in Table 11. We evaluate the performance on the Sintel and KITTI datasets. For both datasets, we use the provided evaluation protocol and report the AEE. For fair comparison, the extra unlabeled data are not used for training. We can see that our scheme achieves the best performance for two-frame based optical flow estimation, with 4.07 and 4.94 of AEE on the *Clean* and *Final* passes of Sintel dataset, especially on the Sintel *Final* pass with a significant performance, with improvement by 0.80, 1.15 and 0.10 in terms of AEE against FlowNet2, LiteFlowNet and PWC-Net, respectively; and 1.4 of AEE on the KITTI 2012 dataset and 7.61% of Fl-all on the KITTI 2015 dataset, respectively.

As shown in Fig. 8, our results are compared with some competing results from baseline methods for two-frame-based optical flow estimation — FlowNet2, LiteFlowNet, and PWC-Net on the Sintel and KITTI datasets. We can see that our FPCR-Net gets better performance, and finer details are well preserved with fewer artifacts of our method. However, the smaller or slimmer objects are the challenges for the network of which edges are not preserved effectively, and we will perform further study on the more challenging cases in the future.

5. Conclusion

To model the motion details in videos for accurate optical flow estimation, we propose a pyramid correlation mapping and residual reconstruction framework — FPCR-Net, to enable the joint analysis of pyramid cost volume and the refinement by stages. The pyramid correlation mapping module takes advantage of the multi-scale correlations of both global and local representation by aggregating features of different scales, while the residual reconstruction module aims to reconstruct the sub-band high-frequency residuals of finer optical flow at each stage. Experiment results show that the proposed scheme achieves the promising performance, with significant improvement against FlowNet2, LiteFlowNet and PWC-Net on the Sintel dataset and the KITTI 2012/2015 dataset.

Table 11
AEE and Fl-all of different methods on Sintel *Clean*/*Final* and KITTI 2012/2015. The "-ft" suffix denotes the fine-tuned networks using the target dataset. The "-wol" suffix denotes the network using the target dataset without labels. The values in parentheses are the results of the networks on the data they were trained on, and hence are not directly comparable to the others.

Method	Sintel *Clean*		Sintel *Final*		KITTI 2012		KITTI 2015		
	train	test	train	test	train	test	train	train(Fl-all)	test(Fl-all)
DeepFlow [5]	2.66	5.38	3.57	7.21	4.48	5.8	10.63	26.52%	29.18%
EpicFlow [1]	2.27	4.12	3.56	6.29	3.09	3.8	9.27	27.18%	27.10%
FlowFields [6]	**1.86**	3.75	3.06	5.81	3.33	3.5	8.33	24.43%	–
Full Flow [41]	–	**2.71**	3.60	5.90	–	–	–	–	–
Deep DiscreteFlow [16]	–	3.86	–	5.73	–	3.4	–	–	21.17%
Patch Matching [20]	–	3.78	–	5.36	–	3.0	–	–	19.44%
DC Flow [42]	–	–	–	5.12	–	–	–	–	14.86%
FlowNetS [2]	4.50	7.42	5.45	8.43	8.26	–	–	–	–
FlowNetS-ft [2]	(3.66)	6.96	(4.44)	7.76	7.52	9.1	–	–	–
FlowNetC [2]	4.31	7.28	5.87	8.81	9.35	–	–	–	–
FlowNetC-ft [2]	(3.78)	6.85	(5.28)	8.51	8.79	–	–	–	–
FlowNet2 [22]	2.02	3.96	3.54	6.02	4.01	–	10.08	29.99%	–
FlowNet2-ft [22]	(1.45)	4.16	(2.19)	5.74	(1.28)	1.8	(2.30)	(8.61%)	11.48%
SPyNet [3]	4.12	6.69	5.57	8.43	9.12	–	–	–	–
SPyNet-ft [3]	(3.17)	6.64	(4.32)	8.36	(3.36)	4.1	–	–	35.07%
LiteFlowNet [23]	2.48	–	4.04	–	4.00	–	10.39	28.50%	–
LiteFlowNet-ft [23]	(1.35)	4.54	(1.78)	5.38	(1.05)	1.6	(1.62)	(5.58%)	9.38%
PWC-Net [4]	2.55	–	3.93	–	4.14	–	10.35	33.67%	–
PWC-Net-ft [4]	(2.02)	4.39	(2.08)	5.04	(1.45)	1.7	(2.16)	(9.80%)	9.60%
3DFlow [25]	–	3.92	–	6.13	–	–	–	–	26.19%
WRT [26]	–	8.24	–	9.46	–	–	–	–	33.39%
SegFlow [21]	–	3.36	–	6.73	–	–	–	–	27.91%
TV-Wavelet-Flow [24]	–	3.36	–	6.91	–	–	–	–	22.41%
DCFlow + KF2 [43]	2.07	3.65	3.25	5.07	–	–	–	–	–
PWC-Net + KF2 [43]	1.75	3.75	2.28	4.98	–	–	–	–	–
Zhai et al. [27]	2.19	4.79	3.71	6.81	3.63	4.57	9.51	–	27.04%
FPCR-Net (Ours)	2.24	–	**3.50**	–	4.35	–	11.83	33.57%	–
FPCR-Net-ft-wol (Ours)	(2.18)	7.83	(3.45)	9.25	(4.09)	4.2	(10.46)	(29.68%)	30.78%
FPCR-Net-ft (Ours)	(1.58)	4.07	(1.97)	**4.94**	(0.93)	**1.4**	(1.45)	(5.38%)	**7.61%**

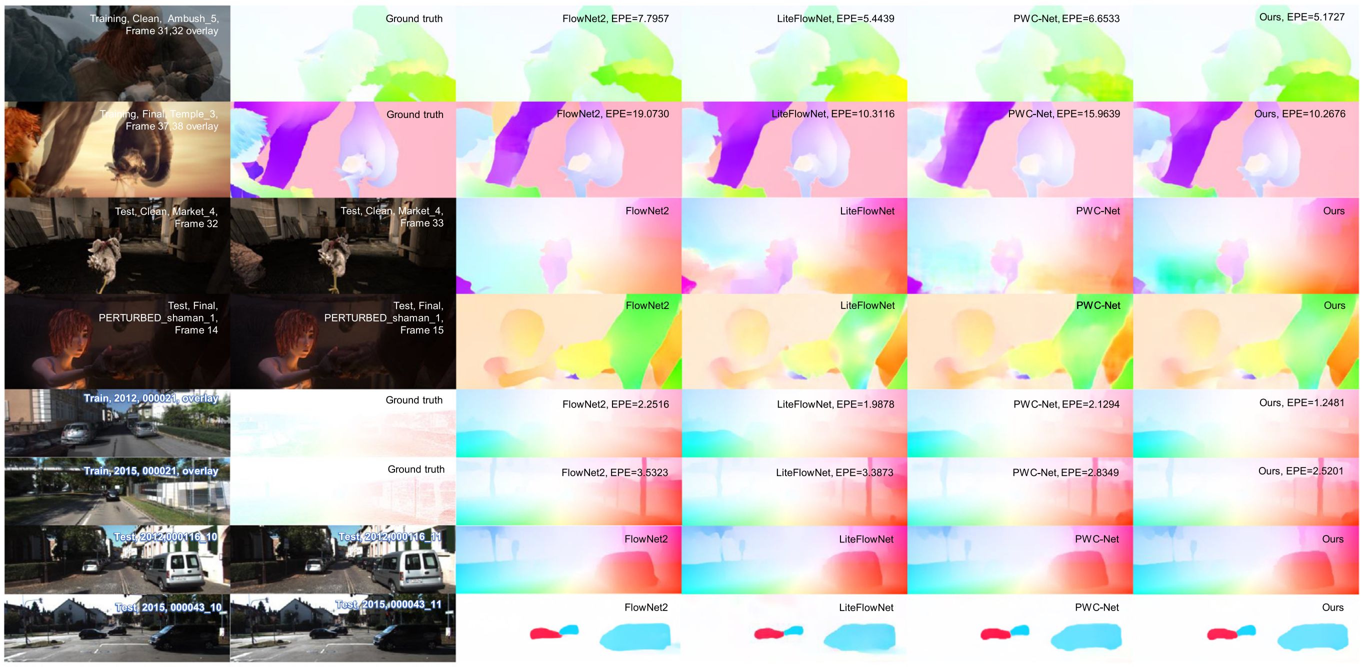


Fig. 8. Examples of predicted optical flow from different methods on Sintel training and test sets for *Clean* and *final* passes. Our method achieves the better performance and preserves the details with fewer artifacts. (Zoom in for details.).

CRediT authorship contribution statement

Xiaolin Song: Conceptualization, Methodology, Software. **Yuyang Zhao:** Software, Validation. **Jingyu Yang:** Writing - original draft. **Cuiling Lan:** Writing - review & editing. **Wenjun Zeng:** Supervision.

Declaration of Competing Interest

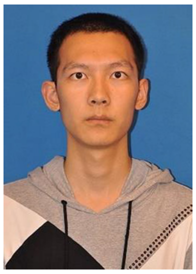
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61771339.

References

[1] J. Revaud, P. Weinzaepfel, Z. Harchaoui, et al., Epicflow: Edge-preserving interpolation of correspondences for optical flow, CVPR, 2015.
[2] A. Dosovitskiy, P. Fischer, E. Ilg, et al., Flownet: Learning optical flow with convolutional networks, in: ICCV, 2015..
[3] A. Ranjan, M.J. Black, Optical flow estimation using a spatial pyramid network, CVPR, 2017.
[4] D. Sun, X. Yang, M.-Y. Liu, et al., Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, CVPR (2018).
[5] P. Weinzaepfel, J. Revaud, Z. Harchaoui, et al., Deepflow: Large displacement optical flow with deep matching, in: ICCV, 2013..
[6] C. Bailer, B. Taetz, D. Stricker, Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation, in: ICCV, 2015..
[7] D. Sun, S. Roth, J. Lewis, M.J. Black, Learning optical flow, in: ECCV, Springer, 2008..
[8] D. Rosenbaum, D. Zoran, Y. Weiss, Learning the local statistics of optical flow, Advances in Neural Information Processing Systems (2013).
[9] J. Wulff, M.J. Black, Efficient sparse-to-dense optical flow estimation using a learned basis and layers, CVPR, 2015.
[10] T. Nir, A.M. Bruckstein, R. Kimmel, Over-parameterized variational optical flow, Int. J. Comput. Vision 76 (2) (2008) 205–216.
[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CVPR (2015).
[12] K. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, CVPR, 2016.
[13] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR (2014).
[14] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, CVPR, 2015.
[15] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, CVPR, 2015.
[16] F. Güney, A. Geiger, Deep discrete flow, in: ACCV, Springer, 2016..
[17] S. Zweig, L. Wolf, Interponet, a brain inspired neural network for optical flow dense interpolation, CVPR, 2017.
[18] J. Thewlis, S. Zheng, P.H.S. Torr, et al., Fully-trainable deep matching, in: BMVC, 2016..
[19] D. Gadot, L. Wolf, Patchbatch: a batch augmented loss for optical flow, CVPR, 2016.
[20] C. Bailer, K. Varanasi, D. Stricker, Cnn-based patch matching for optical flow with thresholded hinge embedding loss, CVPR (2017).
[21] J. Chen, Z. Cai, J. Lai, X. Xie, Efficient segmentation-based patchmatch for large displacement optical flow estimation, IEEE Trans. Circuits Syst. Video Technol. 29 (12) (2019) 3595–3607, https://doi.org/10.1109/TCSVT.2018.2885246.
[22] E. Ilg, N. Mayer, T. Saikia, et al., Flownet 2.0: Evolution of optical flow estimation with deep networks, CVPR (2017).
[23] T.-W. Hui, X. Tang, C. Change Loy, Liteflownet: a lightweight convolutional neural network for optical flow estimation, CVPR (2018).
[24] J. Chen, J. Lai, Z. Cai, X. Xie, Z. Pan, Optical flow estimation based on the frequency-domain regularization, IEEE Trans. Circuits Syst. Video Technol. (2020) 1, https://doi.org/10.1109/TCSVT.2020.2974490.
[25] J. Chen, Z. Cai, J. Lai, X. Xie, A filtering-based framework for optical flow estimation, IEEE Trans. Circuits Syst. Video Technol. 29 (5) (2019) 1350–1364, https://doi.org/10.1109/TCSVT.2018.2805101.
[26] L. Mei, J. Lai, X. Xie, J. Zhu, J. Chen, Illumination-invariance optical flow estimation using weighted regularization transform, IEEE Trans. Circuits Syst. Video Technol. 30 (2) (2020) 495–508, https://doi.org/10.1109/TCSVT.2019.2890861.
[27] M. Zhai, X. Xiang, R. Zhang, N. Lv, A. El Saddik, Optical flow estimation using dual self-attention pyramid networks, IEEE Trans. Circuits Syst. Video Technol. (2019) 1, https://doi.org/10.1109/TCSVT.2019.2943140.
[28] J. Hur, S. Roth, Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation, in: ICCV, 2017..
[29] J. Hur, S. Roth, Iterative residual refinement for joint optical flow and occlusion estimation, CVPR, 2019.
[30] M. Neoral, J. Šochman, J. Matas, Continual occlusion and optical flow estimation, ACCV, 2018.
[31] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, W. Xu, Occlusion aware unsupervised learning of optical flow, CVPR (2018).
[32] S. Meister, J. Hur, S. Roth, Unflow: Unsupervised learning of optical flow with a bidirectional census loss, in: AAAI, 2018..
[33] J. Janai, F. Guney, A. Ranjan, M. Black, A. Geiger, Unsupervised learning of multi-frame optical flow with occlusions, in: ECCV, 2018..
[34] P. Liu, M. Lyu, I. King, J. Xu, Selflow: Self-supervised learning of optical flow, CVPR, 2019.
[35] P. Thomas, Structure- and motion-adaptive regularization for high accuracy optic flow, in: ICCV, 2009..
[36] N. Mayer, E. Ilg, P. Hausser, et al., A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, CVPR (2016).
[37] D.J. Butler, J. Wulff, G.B. Stanley, et al., A naturalistic open source movie for optical flow evaluation, in: ECCV, 2012..
[38] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, CVPR (2012).
[39] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, CVPR (2015).
[40] K. Soomro, A.R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, Computer Science..
[41] Q. Chen, V. Koltun, Full flow: optical flow estimation by global optimization over regular grids, CVPR, 2016.
[42] J. Xu, R. Ranftl, V. Koltun, Accurate optical flow via direct cost volume processing, in: CVPR, 2017.
[43] W. Bao, X. Zhang, L. Chen, Z. Gao, Kalmanflow 2.0: Efficient video optical flow estimation via context-aware kalman filtering, IEEE Transactions on Image Processing 28 (9) (2019) 4233–4246. doi:10.1109/TIP.2019.2903656..



Xiaolin Song received the B.S. degree in Tianjin University, Tianjin, China, in 2016, and is currently working toward the Ph.D. degree in Tianjin Unviversity. His major research interests include computer vision problems related to video analytics.

Yuyang Zhao is currently working toward the B.S. degree in Tianjin Unviversity, Tianjin, China. His major research interests include computer vision problems.

Jingyu Yang received the B.E. degree from Beijing University of Posts and Telecommunications in 2003, and Ph.D. (Hons.) degree from Tsinghua University in 2009. He has been a Faculty Member with Tianjin University, China, since 2009, where he is currently a Research Professor with the School of Electrical and Information Engineering. His research interests include image/video processing, 3D imaging, and computer vision. He has authored or co-authored over 70 high quality research papers (including dozens of IEEE Transactions and top conference papers).

Cuiling Lan received the B.S. degree in electrical engineering and the Ph.D. degree in intelligent information processing from Xidian University, Xi'an, China, in 2008 and 2014, respectively. She joined Microsoft Research Asia, Beijing, China, in 2014. Her research interests include computer vision, video coding.

Wenjun (Kevin) Zeng is a Principal Research Manager and a member of the senior leadership team at Microsoft Research Asia. He has been leading the video analytics research empowering the Microsoft Cognitive Services and Azure Media Analytics Services since 2014. Wenjun has contributed significantly to the development of international standards (ISO MPEG, JPEG2000, and OMA). He received his B.E., M.S., and Ph.D. degrees from Tsinghua Univ., the Univ. of Notre Dame, and Princeton Univ., respectively. His current research interest includes mobile-cloud media computing, computer vision, social network/media analysis, and multimedia communications and security. He is a Fellow of the IEEE.