Signal Processing: Image Communication 99 (2021) 116441

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

STC-Flow: Spatio-temporal context-aware optical flow estimation

Xiaolin Song, Yuyang Zhao, Jingyu Yang [*]

School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, PR China

ARTICLE INFO

Keywords:
Spatio-temporal network
Context modeling
Optical flow estimation
Deep learning

ABSTRACT

In this paper, we propose a spatio-temporal contextual network, STC-Flow, for optical flow estimation. Unlike previous optical flow estimation approaches with local pyramid feature extraction and multi-level correlation, we propose a contextual relation exploration architecture by capturing rich long-range dependencies in spatial and temporal dimensions. Specifically, STC-Flow contains three key context modules, *i.e.*, pyramidal spatial context module, temporal context correlation module and recurrent residual contextual upsampling module for the effect of feature extraction, correlation, and flow reconstruction, respectively. Experimental results demonstrate that the proposed scheme achieves the state-of-the-art performance of two-frame based methods on Sintel and KITTI 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 via convolutional neural networks (CNNs). Despite the great efforts and rapid developments, the advancements are not as significant as those in single 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, which would contain variable motion displacements; thus optical flow estimation requires the efficient representation of features to match objects or scenes of different motions.

Conventional methods propose mathematical algorithms of optical flow estimation such as EpicFlow [1] by matching features of two frames. Most of these methods, however, are complicated with heavy computational complexity, and usually fail for motions with large displacements. CNN-based methods, which usually utilize encoder–decoder architectures with pyramidal feature extraction and flow reconstruction like FlowNet [2], SpyNet [3], PWC-Net [4], boost the state-of-the-art performance of optical flow estimation and outperform conventional methods. However, the stacked convolutional layers they utilize are limited of which the features in lower level contain rich details, while the corresponding receptive field of a single convolutional layer is small, which is not effective to catch the larger displacement of motion. The features in higher level capture the overall outlines or shapes of objects and can catch larger displacement with less details, and they may cause the misalignment for objects with complex shapes or non-rigid motions. So it is essential to capture context information with large receptive field and long-range dependencies, and build the global relationship for each level of CNNs, which could both catch larger displacement and retain more details.

In this paper, as shown in Fig. 1, we propose an end-to-end architecture which jointly explores spatio-temporal context for optical flow estimation. The network contains three key context modules. (a) *Pyramidal spatial context module* aims to enhance the discriminant ability of feature representations in the spatial dimension. (b) *Temporal context correlation module* is designed to model the global spatio-temporal relationships of the cost volume calculated from correlation operation, which is used to measure the effect of correspondence relationships. (c) *Recurrent residual context upsampling module* leverages the underlying content of predicted flow field between adjacent feature levels, to learn high-frequency features and preserve edges within a large receptive field.

In summary, the main contributions of this work are summarized as:

- We propose a general framework, *i.e.* contextual attention framework, for efficient feature representation learning, which explores the multiple level features and comprehensive operation of feature fusion.
- We propose three corresponding context modules in the contextual attention framework, for feature extraction, correlation and optical flow reconstruction, aiming at improving the overall performance via better feature representation and correlation and enhancing high-frequency details with context information.

[*] Corresponding author.
E-mail addresses: songxl@tju.edu.cn (X. Song), yuyangzhao@tju.edu.cn (Y. Zhao), yjy@tju.edu.cn (J. Yang).

https://doi.org/10.1016/j.image.2021.116441
Received 11 October 2020; Received in revised form 28 June 2021; Accepted 11 August 2021
Available online 22 August 2021

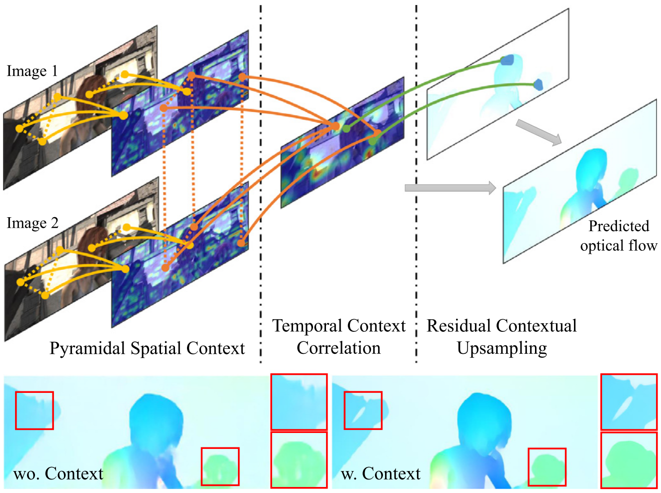
Image 1
Image 2
Pyramidal Spatial Context
Temporal Context Correlation
Residual Contextual Upsampling
Predicted optical flow
wo. Context
w. Context

Fig. 1. Overview of spatio-temporal contextual network for optical flow estimation. The context modules aim to build the relationship in spatial and temporal dimensions. With multiple context modeling, STC-Flow achieves the better performance with rich details.

2. Related work

Optical flow estimation. Horn and Schunck [5] pioneer the study on optical flow estimation, of which method takes advantage of illumination changes, and uses an iterative implementation. Brox et al. [6] propose a warping-based optical flow prediction method. Brox et al. [7] use rich descriptors for feature matching to estimate a dense optical flow field with large displacement. Zimmer et al. [8] propose the anisotropic smoothness as the constraint term for data regularization. Weinzaepfel et al. [9] propose DeepFlow to correlate multi-scale patches to retrieve quasi-dense correspondences efficiently. Tu et al. [10] present a combined post-filtering with a 3D nonlinear structure tensor based edge detector to identify edges and a piecewise occlusion detector to detect flow occlusion. Revaud et al. [1] propose EpicFlow that uses dense matching via the interpolation of a sparse set of matches between the two consecutive images to obtain dense flow. Tu et al. [11] propose a weighted local intensity fusion method to fuse optical flow proposals to handle large displacements and estimate the smoothness parameter. Tu et al. [12] propose an energy function with edge-aware constraints integrated image fidelity term for jointly optical flow estimation and image restoration. Chen et al. [13] use TV-wavelet regularization for lost optical flow information. Inspired by the success of CNNs, many deep networks for optical flow estimation have been proposed. Dosovitskiy et al. [2] propose FlowNetS and FlowNetC networks with the encoder–decoder architecture design for optical flow estimation. However, the number of parameters is quite large, which results in heavy computation cost. Ilg et al. [14] propose a cascaded network, *i.e.* FlowNet2, with better performance with huge number parameters and expensive computation complexity. Some methods use CNN models for image patches matching. Thewlis et al. [15] utilize Deep Matching formulation into a CNN for end-to-end training. Gadot [16] and Bailer et al. [17] use patch matching for Siamese network architectures with heavy computation cost. Deep DiscreteFlow [18] utilizes a local network and a context network for optical flow estimation. Chen et al. [19] propose a coarse-to-fine segmentation-based PatchMatch with sparse seeds for optical flow estimation. Moreover, patch matching based methods lack the capacity to explore larger context of the image because of the small image patch based operator. PatchFlow [20] introduces a patch-based consistency for unsupervised optical flow and occlusion estimation. In addition, many unsupervised methods are proposed to estimate optical flow. Wang et al. [21] propose a unified framework for unsupervised learning of optical flow and design a rigid-aware direct visual odometry module to handle rigid regions. DDFlow [22] utilizes data distillation from the teacher network to guide a student network to learn the flow field.

To reduce the number of parameters, Ranjan et al. [3] present a compact SPyNet for spatial pyramid with multi-level representation learning, which is utilized with a light architecture of feature-level matching and warping motivated by conventional methods. Nevertheless, the performance is not significant. Hui et al. [23] propose LiteFlowNet and Sun et al. [4] propose PWC-Net, which explore the lightweight optical flow prediction networks. LiteFlowNet [23] uses the flow inference of a cascaded network for flow warping and feature matching. PWC-Net [4] uses feature pyramid extraction in the backbone network, and warp the feature at each level to construct the cost volume. Hierarchical Discrete Distribution Decomposition (HD^3) [24] decomposes the full match density into hierarchical features to estimate the local matching, with high computational complexity. Iterative Residual Refinement (IRR) [25] involves the iterative residual refinement, and integrates occlusion prediction as an additional auxiliary supervision. SelFlow [26] uses reliable flow predictions from non-occluded pixels, to compensate optical flow for invisible occlusions. Bao et al. [27] introduce Kalman filtering into CNN-based methods to improve the robustness against the change of illumination and occlusion. Chen et al. [28] adopt different filtering operations for regularization with respect to consistency. MaskFlownet [29] filters useless areas by a learned rough occlusion mask. ScopeFlow [30] modifies the common training protocols by cropping randomly sized scene scopes to enhance the performance. Instead of coarse-to-fine manner, RAFT [31] operates on a single high-resolution flow field, with a recurrent and lightweight update operator for iterative refinement. Inspired by these works, we use lightweight pyramid networks to accelerate the calculation process and introduce additional spatial and temporal information interaction to improve accuracy.

Bio-inspired motion estimation. Bio-inspired motion estimation [32–36] allows to explain and understand human behavior on optical flow, which estimates motion along the visual dorsal pathway in primates. Pauwels et al. [32] emulate large parts of the dorsal stream in an abstract way and implement an architecture with optical flow feature extraction stages, which are used to reliably extract moving objects in real time. Kruger et al. [33] propose functional principles of deep hierarchical processing in the primate visual system. Tschechne et al. [34] propose a simplified version of initial stages of cortical processing combines filters with spatio-temporal tunings to represent movements along the dorsal pathway. Chessa et al. [35] propose a neural feed-forward model that mimic the V1–MT primary motion pathway, to yield a population of pattern cells that encodes the local velocities of the visual stimuli. Solari et al. [36] propose a computational model which uses hierarchical cells' layers that model the neural processing stages of the dorsal visual pathway, and produces selectivity for specific patterns of optical flow.

Context modeling in neural networks. Context modeling has been successfully applied to capture long-range dependencies. Since a typical convolution operator has a local receptive field, context learning can affect an individual element by aggregating information from all elements. Many recent works utilize spatial self-attention to emphasize features of the key local regions. Object relation module [37] extends original attention to geometric relationship, which could be used to improve the performance of object detection and other tasks. DANet [38] introduces the channel-wise attention via self-attention mechanism. Global context network [39] effectively models the global context with a lightweight architecture. Non-local network [40] uses 3D convolution layers to aggregate spatial and temporal long-range dependencies for video frames.

In the optical flow estimation task, spatial contextual information helps to refine details and deal with occlusion. PWC-Net [4] consists of the context network with stacked dilated convolution layers for

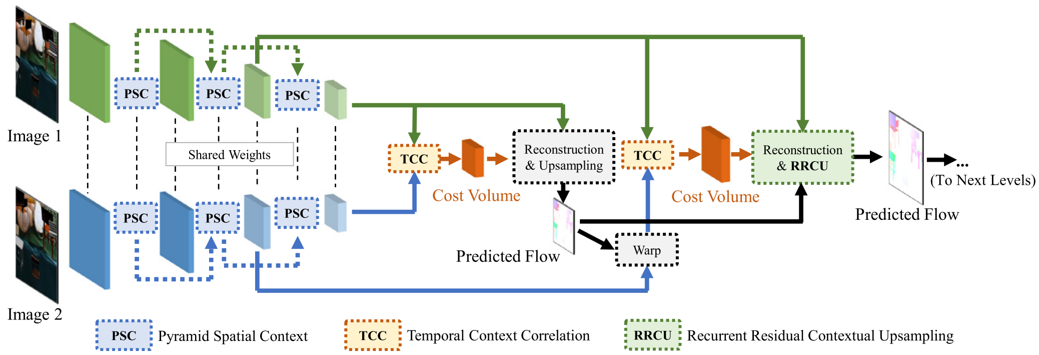

Fig. 2. Overall architecture of our proposed spatio-temporal contextual network for optical flow estimation (STC-Flow), which builds the contextual information in spatial and temporal dimensions. Pyramidal spatial context (PSC), temporal context correlation (TCC), and recurrent residual contextual upsampling (RRCU) modules are flexible to adopt to model spatial and temporal relationships of intra-/extra-features in each level, which improve the overall performance and preserve details for optical flow. We only show these modules at the top two levels.

flow post-processing. In LiteFlowNet [23], flow regularization layer is applied to ameliorate the problem of outliers and fake edges. IRR [25] uses bilateral filters to refine blurry flow and occlusion. Mei et al. [41] exploit weighted regularization of the unequal probability with non-local information, to estimate stable optical flow despite illumination changes. Zhai et al. [42] integrate local features with their global dependencies and focus on important features and suppress unimportant spatial features. Young et al. [43] propose a robust optical flow estimation method based on Gaussian graph Laplacians. Nevertheless, previous work of context modeling in optical flow estimation mainly focuses on spatial representation. For motion context modeling, it is essential to provide an elegant framework to explore both spatial and temporal information. In this work, we introduces spatial and temporal context module to efficiently explore the global spatial and temporal contexts.

3. STC-Flow

Given a pair of video frames, scene or objects are diverse on movement velocity and direction. They change in scales, views, and luminance. Convolutional operations in CNNs are in general performed just in a local neighborhood. The pixels of the same non-rigid object may have similar textures and features, even though they may have different motions. For instance, (1) The objects are non-rigid with different obvious motion situation of each part, such as the girl walking in the street and the dragon flying through the path in Sintel dataset, and (2) The objects are near the boundaries of the view, and the motion intensity is different with the camera moving, such as the trees and railings on both sides of the road in KITTI dataset. These would result in false-positive correlation and thus wrong prediction of optical flow, and it is essential to utilize the global contextual corresponding modules for modeling objects/scenes.

To address this issue, our method of STC-Flow models contextual information by building global associations of features with the self-attention mechanism in spatial and temporal dimensions, respectively. The network adaptively aggregates long-range contextual information for optimizing feature representation in feature extraction, correlation, and reconstruction stages, as shown in Fig. 2. In this section, we first introduce the contextual attention framework with single or multiple inputs for efficient feature representation learning. Based on this framework, we propose three key contextual modules: *pyramidal spatial context (PSC) module*, *temporal context correlation (TCC) module*, and *recurrent residual contextual upsampling (RRCU) module* for modeling contextual information.

3.1. Contextual attention framework

Analysis on Attention Mechanism. To capture long-range dependencies and model contextual details for single images or video clips, the non-local network [40] aggregates pixel-wise information via self-attention mechanism. We denote X and Z as the input and output signals. The non-local block can be modeled as:

$$Z_i = X_i + \boldsymbol{\Phi}_z \sum_j \frac{f(X_i, X_j)}{\mathcal{N}(X)} (\boldsymbol{\Phi}_v X_j), \tag{1}$$

where i and j are the indices of feature positions. $f(X_i, X_j)$ denotes the affinity between features of positions i and j, and we normalize them by a factor $\mathcal{N}(X)$. The matrix multiplication operation is used to strengthen details of each query position. Embedded Gaussian is a widely-used instantiation of $f(X_i, X_j)$, to compute similarity in an embedding space. The non-local block with Embedded Gaussian is modeled as follows:

$$Z_i = X_i + \boldsymbol{\Phi}_z \sum_j \frac{\exp((\boldsymbol{\Phi}_q X_i)^\top (\boldsymbol{\Phi}_k X_j))}{\sum_m \exp((\boldsymbol{\Phi}_q X_i)^\top (\boldsymbol{\Phi}_k X_m))} (\boldsymbol{\Phi}_v X_j), \tag{2}$$

where $\boldsymbol{\Phi}_q$, $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}_v$ are linear transformation matrices.

Why attention for optical flow estimation? Here, we discuss the relation between correlation in optical flow estimation and matrix multiplication in self-attention mechanism. We aim to explore the contextual information from the input feature nodes. We denote the two feature maps by F_1 and F_2. We denote the size of features with height H_1 and H_2, width W_1 and W_2, channel C_1 and C_2, and position coordinate $\mathbf{x}_1 \in [1, H_1] \times [1, W_1]$ and $\mathbf{x}_2 \in [1, H_2] \times [1, W_2]$, channel index $c_1 \in C_1$ and $c_2 \in C_2$, and here $H_1 = H_2$, $W_1 = W_2$ and $C_1 = C_2$.

As the key operation in optical flow estimation, the "correlation" operation between two patches, $\mathbf{f}_1$ and $\mathbf{f}_2$, from F_1 and F_2 respectively, is defined as:

$$Corr(\mathbf{f}_1, \mathbf{f}_2) = \sum_{\mathbf{o}} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle, \tag{3}$$

where $Corr(\mathbf{f}_1, \mathbf{f}_2)$ denotes the cost volume calculated via correlation. $\mathbf{o} \in [-n, n] \times [-n, n]$ denotes the offset of correlation operation with search region. In consideration of matrix multiplication in the attention mechanism of $f(X_i, X_j)$, the different order of the two matrices (X_i and X_j) in multiplication leads to great disparity of correlation, and thus causes the disparity of the direction of optical flow.

The expression is defined as $F_2(\mathbf{x}_2, c_2)(F_1(\mathbf{x}_1, c_1))^\top \in \mathbb{R}^{H_2 W_2 \times H_1 W_1}$, which is shown in Fig. 4(a). If $F_1 = F_2$, this operation strengthens the detail representation of each position via aggregating information across channels from other positions, which would indicate the spatial attention integration at full resolution. However, if $F_1 \neq F_2$, as the inputs of correlation operation, different elements present the correlation with different displacements, and only the diagonal elements present no displacement. On the contrary, the expression is defined as

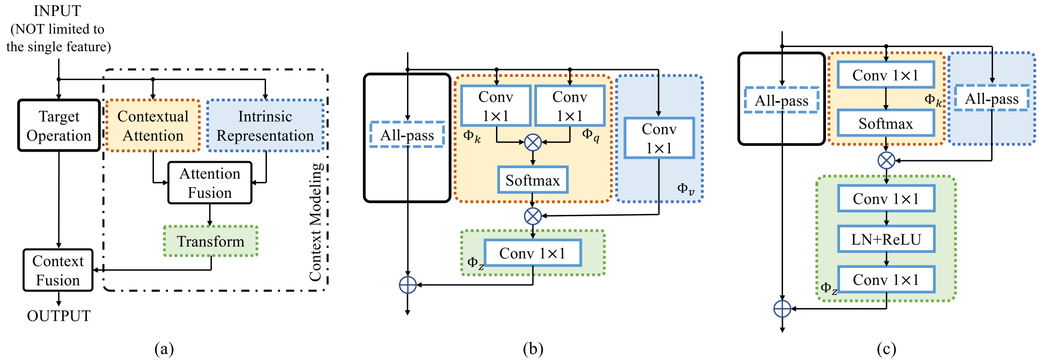

Fig. 3. The contextual attention framework (a) with modularization; and the specified forms of (b) the non-local block [40], and (c) global context (GC) block [39].

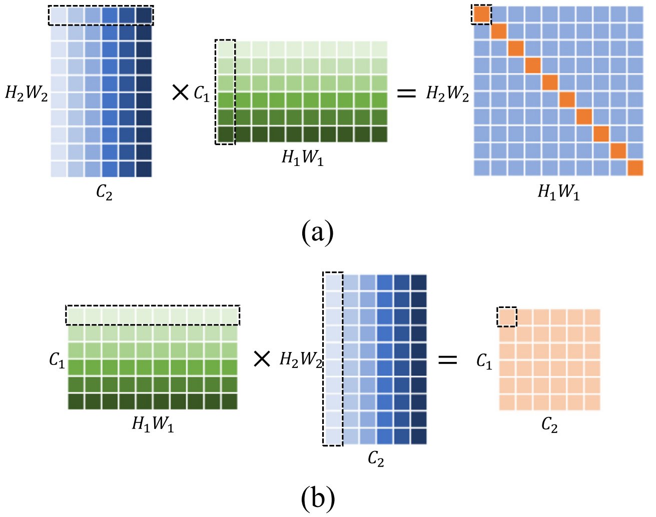

Fig. 4. The matrix multiplication with different contextual information. (a) The position-wise attention embedding; (b) the channel-wise embedding, also the global correlation of feature pairs.

$(F_1(\mathbf{x}_1, c_1))^\top F_2(\mathbf{x}_2, c_2) \in \mathbb{R}^{C_1 \times C_2}$ in Fig. 4(b), which is a global correlation representation at full resolution among channels, and is essential to the naive correlation operation between feature pairs. For different matrix multiplication approaches, the attention maps catch dependencies with corresponding concepts in spatial features and temporal dynamics, which enhance the representation feature extraction and correlation calculation, respectively.

Contextual Attention Framework. In general, the input of CNNs is not limited to the single feature through the single path, and the attention block needs to be adapted to more than one inputs, *e.g.* two input feature maps of the correlation operation. As shown in Fig. 3(a), the components of the attention block can be abstracted as follows:

- *Attention aggregation.* To aggregate the attention integration features to the intrinsic feature representation in each corresponding dimension, where the intrinsic representation often adopts basic operators like interpolation, convolution and transposed convolution.
- *Context transformation.* To transform the aggregated attention via the multi-layer perceptron (MLP) with 1 × 1 convolution, and obtain the contextual attention features of all positions and channels.
- *Target fusion.* To aggregate the output feature from target operation with the contextual attention, where the target operation is the main function to attain the objective from input features.

Denote $X^{(k)}$ as the multiple input features. We regard this abstraction as a contextual attention framework defined as follows:

$$Z = \mathcal{G}\left(T\left(X^{(k)}\right), \mathcal{F}\left(A\left(X^{(k)}\right), \sum_k \phi_k X^{(k)}\right)\right), \qquad (4)$$

where $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$ are the fusion operations for attention aggregation and target fusion. $T(\cdot)$ and $A(\cdot)$ denote target operation and attention integration for the input features, ϕ is the factor of linear transformation. The non-local block or the other attention modules are the specific form of context attention block with the single input feature, *e.g.* $A_{ij}(X) = f(X_i, X_j)/\mathcal{N}(X)$, and $T(X)$ is the all-pass function in the non-local block.

Lite matrix multiplication. Considering the runtime of the flow prediction, the matrix multiplication in contextual attention block needs to be simplified with less computational complexity. In Fig. 5, according to the neighbor similarity of images or frame pairs, we propose the polyphase decomposition and reconstruction scheme to simplify matrix multiplication operation, which would obtain better approximation than the naive downsampling–upsampling scheme, and reduce the computation complexity compared to the conventional matrix multiplication. Denote the polyphase decomposition factor as s ($s > 1$). Polyphase decomposition separates elements in one $s \times s$ area, and divides the original matrix into s^2 small matrices. Each small matrix contains $\frac{R}{s}$ rows and $\frac{C}{s}$ columns, where R and C is the number of rows and columns in the original matrix. After decomposition, each small matrix multiplies the corresponding decomposed part. In polyphase reconstruction stage, all elements in small matrices gather together in the location of the original matrix to compose the target matrix. Given a reshaped feature $\tilde{F} \in \mathbb{R}^{M \times N}$, the FLOPs of the entire multiplication is reduced from $O\left(NM^2\right)$ to $O\left(\frac{NM^2}{s}\right)$. The comparison of different factors is presented in Section 4.

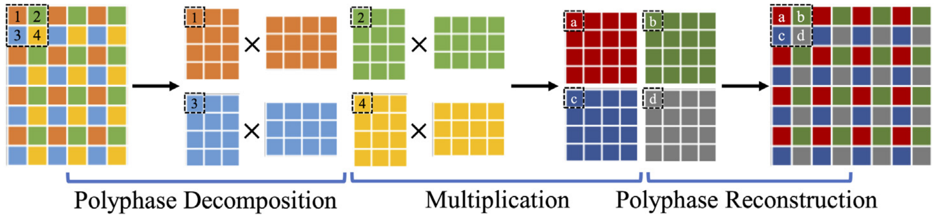

Fig. 5. The proposed simplified matrix multiplication with polyphase decomposition and reconstruction.

3.2. Pyramidal spatial context module

Inspired by the non-local network and global context network, we propose a pyramidal spatial context module with a tight dual-attention block to enhance the discriminative ability of feature representations in spatial position and channel dimensions. As shown in Fig. 6, given a local feature $F^{(k)} \in \mathbb{R}^{C\times H\times W}$ at level k, attention information is calculated by inner product and non-linear operations, which aggregate long-range correlated information. We combine the attention information with local feature to obtain position-wise context and channel-wise context, which are added back to local feature to improve feature representation. The calculation of the spatial context module is formulated as:

$$\tilde{F}^{(k)} = F^{(k)} + C_P^{(k)} + C_C^{(k)}, \tag{5}$$

where $C_P^{(k)}$ and $C_C^{(k)}$ are contextual attention at level k fused with that of level $k-1$, which is to aggregate context from different granularity:

$$\begin{aligned} C_P^{(k)} &= \Phi_z^{(k)}\left[\sum_j A_{P,ij}^{(k)} F_j^{(k)}, C_P^{(k-1)} \Downarrow\right], \\ C_C^{(k)} &= \Phi_z^{(k)}\left[\sum_j A_{C,ij}^{(k)} F_j^{(k)}, C_C^{(k-1)}\right], \end{aligned} \tag{6}$$

where "$\Downarrow$" denotes max-pooling, and "$[\cdot]$" denotes the concatenation operator. $A_{P,ij}$ and $A_{C,ij}$ are attention integrations in position and channel, defined as follows to learn the spatial and channel interdependencies:

$$\begin{aligned} A_{P,ij}(F) &= \frac{\exp((\Phi_q F_i)^\top(\Phi_k F_j))}{\sum_m \exp((\Phi_q F_i)^\top(\Phi_k F_m))} \in \mathbb{R}^{HW\times HW}, \\ A_{C,ij}(F) &= \frac{\exp(\Phi_k F_j)}{\sum_m \exp(\Phi_k F_m)} \in \mathbb{R}^{HW\times 1}. \end{aligned} \tag{7}$$

3.3. Temporal context correlation module

After the spatial context module learns query-independent context relationships at the feature extraction stage, the temporal context module is adopted to model the relationships of correlation. As the analysis on matrix multiplication, correlation with long-range dependencies is used to describe the global context of correlation operation. As shown in Fig. 7(a), given the local feature pairs $F_1, F_2 \in \mathbb{R}^{C\times H\times W}$ from feature extraction, the contextual correlation is formulated as:

$$Z_i = \Phi_c Corr_i(F_1, F_2) + \Phi_z \sum_j \left(A_{T,ij} \cdot \Phi_v(F_1, F_2)\right), \tag{8}$$

where $A_{T,ij}$ is the temporal attention integration with the *cross-attention* mechanism, which is defined as follows:

$$A_{T,ij}\left(F_1, F_2\right) = \frac{\exp((\Phi_q F_{1,i})^\top(\Phi_k F_{2,j}))}{\sum_m \exp((\Phi_q F_{1,i})^\top(\Phi_k F_{2,m}))} \in \mathbb{R}^{C\times C}. \tag{9}$$

Notice that the linear transformation of $\Phi_v(F_1, F_2)$ is modeled by a 3D convolution and a 1×1 convolution, which aims to explore the temporal information across time dimension. Since the max displacement of correlation is selected to 4, the kernel of 3D convolution needs to cover all frames in the temporal dimension, and the height and width are larger than or equal to the max displacement, *i.e.* 5 in the proposed module. In addition, "CN" in TCC denotes "Channel Normalization". Denote the input feature by $F \in \mathbb{R}^{C\times H\times W}$, the formulation of CN is expressed as follows:

$$CN(\mathbf{x}, c) = F(\mathbf{x}, c) \Bigg/ \left(\alpha \sum_{\mathbf{x}} (F(\mathbf{x}, c))^2 + \epsilon\right)^{\beta}, \tag{10}$$

where $\mathbf{x} \in [1, H] \times [1, W]$ and $c \in [1, C]$ denote the position and the channel of F, respectively. α, β and ϵ denote the multiplier, the exponent and the additive constant with a small value for normalization term, respectively. This normalization aims to indicate the relative value in each channel.

The TCC module is a flexible correlation operator and it can be used in PWC-Net [4] as "Contextual PWC" module, to learn long-dependencies between the reference features and the warped features.

3.4. Recurrent residual contextual upsampling

Different from the spatial and temporal context representation modeling, the reconstruction context learning is a detail-aware operation to learn high-frequency feature and to preserve edges with a large receptive field. In view of the multi-level structure of reconstruction, we propose an efficient recurrent module for upsampling, which leverages the underlying content information between the current level and the previous level.

In general, RRCU module aims at upsampling predicted flow at current level with the information compensation from the previous level. The predicted optical flow $Y^{(k)}$ at level k and the upsampled optical flow $\tilde{Y}^{(k+1)}$ obtained from RRCU module at level $k+1$ are encoded by 1×1 convolution. Denote the residual between $Y^{(k)}$ and $\tilde{Y}^{(k+1)}$ as $R^{(k)} = Y_i^{(k)} - \tilde{Y}_i^{(k+1)}$, and then the context modeling is utilized for $R^{(k)}$ to explore the up-sampling attention kernels A_U in each corresponding source position, and A_U is fused back to the bilinear interpolated $R^{(k)}$. Finally, the fined residual feature $\tilde{R}^{(k)}$ is resembled to $Y^{(k)}$ to obtain the refined upsampled flow $\tilde{Y}^{(k)}$ with rich details. The architecture is illustrated in Fig. 8, and the formulation is expressed as follows:

$$\tilde{Y}_i^{(k)} = deconv(Y_i^{(k)}) + \Phi_z \sum_i \left(A_{U,i} * \Phi_v R_i^{(k)}\right), \tag{11}$$

where "*" denotes the position-wise convolution operator, and here W_v is a bilinear interpolation operator for $R^{(k)}$. $A_{U,ij}$ denotes the adaptive attention kernels to model the detail context defined as follows:

$$A_{U,i}(R) = \frac{\exp(ps(\Phi_r R_i))}{\sum_m \exp(ps(\Phi_r R_m))} \in \mathbb{R}^{\sigma^2\times H\times W}, \tag{12}$$

where ps denotes the "Pixel Shuffle [44]" operator for sub-pixel convolution, which is a periodic shuffling operator that rearranges the elements of a $H\times W\times C\cdot\sigma^2$ tensor to a tensor of shape $\sigma H\times\sigma W\times C$. Pixel Shuffle reconstructs the sub-pixel information to preserve edges and textures. σ is the upsampling factor, and here $\sigma = 2$.

3.5. Overall architecture

Given the proposed contextual attention modules, we now describe the overall architecture of the proposed STC-Flow. The input is the frame pairs I_1 and I_2 with size $3\times H\times W$, and the goal of STC-Flow is to obtain the optical flow map Y with size $2\times H\times W$. The contextual representations are modeled via three key components — pyramidal spatial context (PSC) module, temporal context correlation (TCC) module, and recurrent residual contextual upsampling (RRCU) module, to capture long-range dependencies relationship in feature extraction, correlation and flow reconstruction, respectively. The entire network is trained jointly, shown in Fig. 2.

Since PWC-Net [4] and LiteFlowNet [23] provide superior performance with lightweight architectures, we take a simplified version of PWC-Net, with layer reduction in feature extraction and reconstruction, as the baseline of our STC-Flow. For successive image/frame pairs, the backbone network with PSC outputs pyramidal feature maps for each image. With the feature maps of each level converted to cost volumes via correlation operation, the cost volumes are decoded and reconstructed to optical flow by TCC. With the guidance of backbone features and warping alignments, the predicted flow field goes through the RRCU module and the fined flow is obtained.

Training Loss. Considering the semi-dense ground truth of the KITTI benchmark, we propose a novel multi-scale loss function, *i.e.*, pyramid mask-invariant loss, to retain valid flow values and masks at different levels as shown in Fig. 9. We denote Θ as the learnable parameters

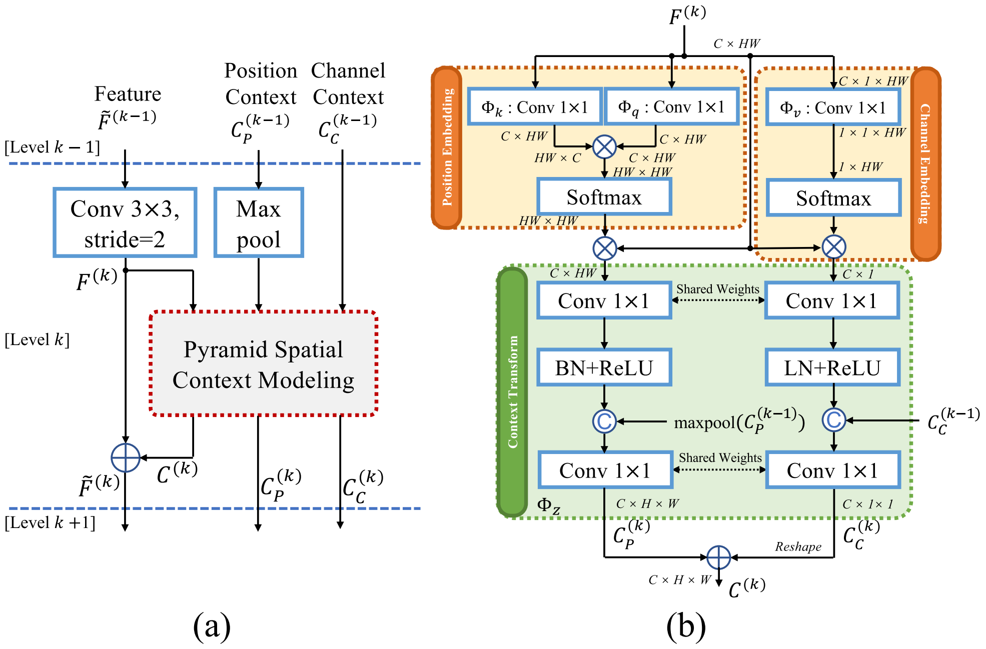

Fig. 6. The pyramidal spatial context (PSC) module. (a) The framework of PSC in the network; (b) The details of "Pyramidal Spatial Context Modeling" in (a).

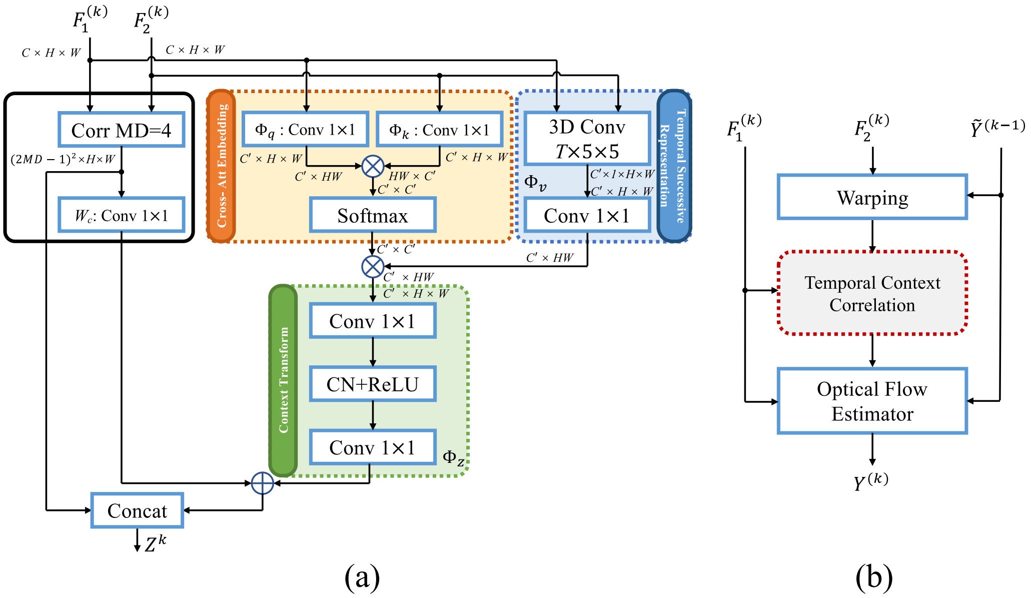

Fig. 7. The temporal context correlation (TCC) module. (a) The details of TCC module; (b) The Contextual PWC Module utilized with TCC in (a). "MD" is the max displacement of correlation. The temporal successive representation utilizes 3D convolution with kernel size $T \times 5 \times 5$, and T is the frame number of input, *i.e.* 2.

of our network. $Y^{(k)}$ and $Y_{GT}^{(k)}$ denote the predicted optical flow at the kth level and the corresponding ground truth flow, respectively. The ground truth in different resolution $Y_{GT}^{(k)}$ is obtained through average pooling. However, the value of the ground truth would be erroneous if the ground truth is semi-dense. Therefore, we use the same average pooling operation on the mask, denoted as $M_{avg}^{(k)}$, with the same resolution of $Y_{GT}^{(k)}$, as the factor to rectify the deviation of average-pooled label. And $Y_{GT}^{(k)}/M_{avg}^{(k)}$ is restored as the label at level k. In addition, the corresponding mask at level k is generated from full-resolution mask via max pooling operation, denoted as $M_{max}^{(k)}$. According to the general Charbonnier function, the proposed pyramid mask-invariant loss is as follows:

$$\mathcal{L}(\Theta) = \sum_k \alpha_k \sum_{\mathbf{x}} \left(|Y^{(k)}(\mathbf{x}) - Y_{GT}^{(k)}(\mathbf{x})/M_{avg}^{(k)}| \cdot M_{max}^{(k)} + \epsilon \right)^q + \gamma |\Theta|_2, \qquad (13)$$

where $|\cdot|$ denotes the L1 norm, and q gives the penalty of the difference between the label and predicted flow and $q < 1$. ϵ is a small positive constant. Specially, for fully-dense datasets, such as Sintel, each mask is an all-one matrix with the same height and width of input frames.

4. Experiments

In this section, we introduce the implementation details, and evaluate our method on public optical flow benchmarks, including MPI Sintel [45], KITTI 2012 [46] and KITTI 2015 [47], and compare it with state-of-the-art methods.

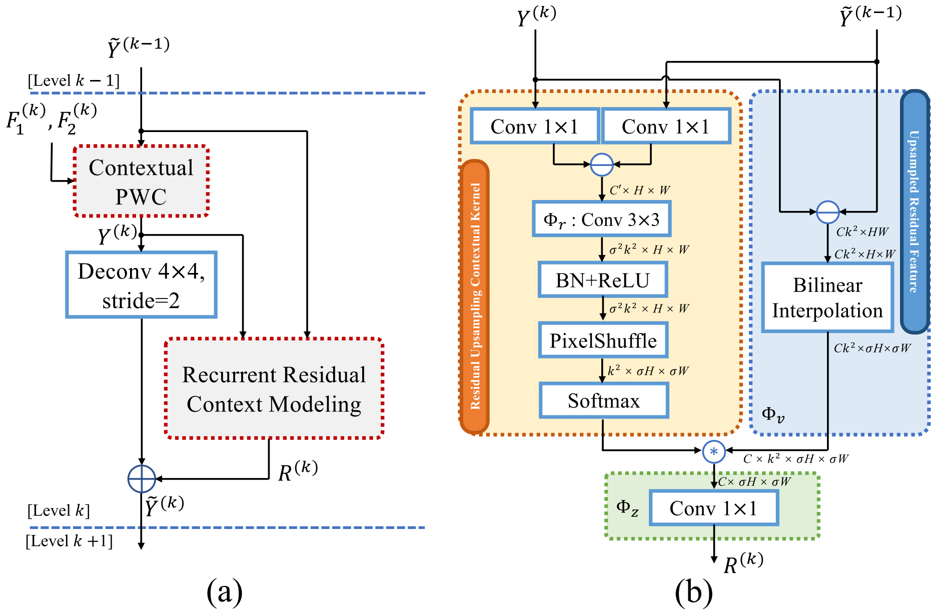

Fig. 8. The recurrent residual contextual upsampling (RRCU) module. (a) The framework of RRCU in the network, with the Contextual PWC module in Fig. 7; (b) The details of "Recurrent Residual Context Modeling" in (a).

Table 1
Ablation study of our component choices of the network. Average end-point error (AEE) and percentage of erroneous pixels (Fl-all) Results of our STC-Flow with different components of PSC, TCC and RRCU on Sintel training *Clean* and *Final* passes, and KITTI *2012/2015*.

(a) Pyramidal Spatial Context Module improves quantity results significantly. "w/PSC$_{3-5}$" means "using PSC in level 3, 4 and 5"

Methods	Sintel		KITTI 2012	KITTI 2015	
	Clean	Final	AEE	AEE	Fl-all
Baseline	2.924	4.088	4.621	11.743	36.53%
w/PSC$_3$	2.802	3.891	4.565	11.031	35.37%
w/PSC$_{3-4}$	2.747	3.873	4.545	10.677	34.84%
w/PSC$_{3-5}$	2.741	3.864	4.494	10.332	34.45%
w/2D-NL$_{3-5}$	2.785	3.968	4.523	10.482	34.76%
Full model	2.412	3.601	4.196	10.181	32.23%

(b) Temporal Context Correlation Module is critical and outperforms single correlation module.

Methods	Sintel		KITTI 2012	KITTI 2015	
	Clean	Final	AEE	AEE	Fl-all
Baseline	2.924	4.088	4.621	11.743	36.53%
w/TCC$_6$	2.787	3.863	4.523	10.712	35.59%
w/TCC$_{3-6}$	2.641	3.780	4.389	10.313	34.58%
w/2D-NL$_{3-6}$	2.764	3.869	4.498	10.564	35.25%
w/3D-NL$_{3-6}$	2.635	3.745	4.393	10.324	34.63%
Full model	2.412	3.601	4.196	10.181	32.23%

(c) Recurrent Residual Context Upsampling has better performance.

Methods	Sintel		KITTI 2012	KITTI 2015	
	Clean	Final	AEE	AEE	Fl-all
Baseline	2.924	4.088	4.621	11.743	36.53%
w/RRCU	2.696	3.794	4.432	10.332	34.65%
TCC+RRCU	2.567	3.722	4.368	10.295	33.89%
Full model	2.412	3.601	4.196	10.181	32.23%

4.1. Implementation and training details

We take a simplified version of PWC-Net as the baseline, with the same number of levels. Since the PSC Module has great effect of feature representation and the RRCU Module improves the reconstruction significantly, we reduce the layers in feature extraction and reconstruction in PWC-Net and utilize PSC, TCC and RRCU to construct our network. PSC Module and RRCU Module are used at level 3, 4 and 5 for feature extraction and reconstruction respectively. TCC Module is applied at level 3, 4, 5 and 6 for correlation of feature pairs or warped features. The training loss weights among levels are 0.32, 0.08, 0.02, 0.01, 0.005. We first train the models with the FlyingChairs dataset [2] with L2 loss and the S_{long} learning rate schedule, with augmentation scheme of random flipping and cropping of size 448 × 384 introduced by [14]. Secondly, we fine-tune the models on the FlyingThings3D dataset [48] using the S_{fine} schedule with cropping size of 768 × 384. Finally, the model is fine-tuned on Sintel and KITTI datasets using the proposed pyramid mask-invariant loss as the robust training loss. We use both the

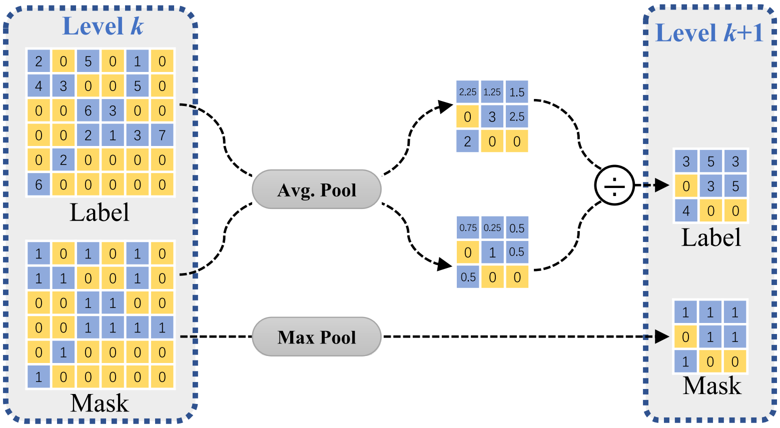

Fig. 9. Illustration of labels and masks in level k and $k + 1$ for pyramid mask-invariant loss.

clean and *final* pass of the training data throughout Sintel fine-tuning process, with cropping size of 768 × 384; and we use the mixed data of KITTI 2012 and 2015 training for KITTI fine-tuning process, with cropping size of 896 × 320.

4.2. Ablation study

To demonstrate the effectiveness of individual contextual attention module in our network, as shown in Table 1 and Fig. 10, we conduct an ablation study of PSC, TCC, and RRCU, respectively. From Table 1, we observe that these three modules improve the performance clearly, thanks to the capturation of the semantic information with long-range dependencies. The baseline is trained on FlyingChairs and finetuned on FlyingThings3D. We also discuss the efficacy of Lite matrix multiplier in Table 2.

Pyramidal spatial context module. STC-Flow uses PSC Module at level 3, 4 and 5. Table 1(a) demonstrates that using PSC Module can improve the performance on both Sintel and KITTI datasets, since this module enhances the ability of discriminating feature texture in feature extraction stage, and PSC at level 3 is more beneficial, for the low-level discriminative details matter.

Temporal context correlation module. TCC Module describes the relationship of correlation with the spatial and temporal context. In Table 1(b), we compare the performance of our network using TCC Module with naive correlation operator, and also compare with 2D non-local block for concatenated feature and 3D non-local block for feature pairs. It demonstrates that fusion of correlation with spatial and temporal context is better than single correlation. Notice that 3D non-local blocks perform better in Sintel, however, with heavy computational complexity. TCC can achieve the comparable performance with fewer FLOPs.

Recurrent residual contextual upsampling. We use the RRCU Module to learn high-frequency context features and preserve edges. In Table 1(c), we compare the quantity of our method using RRCU with single transpose convolutional layers, which demonstrates that reconstruction context learning could preserve details and improve performance.

Lite matrix multiplication. Lite matrix multiplication is an efficient scheme to reduce the computational complexity. We compare the performance of this scheme with different polyphase decomposition factor s on Sintel training. As shown in Table 2, lite matrix multiplication has a margin influence on AEE, but increases the frame rate conspicuously. Fig. 11 shows the visualization of feature maps at level 3 with the proposed lite matrix multiplication of different polyphase decomposition factors. These features are normalized by the same value for visualization. We can see that the feature of polyphase decomposition factor $s = 2$ is more similar to that of $s = 1$ ($s = 1$ means the naive matrix multiplication), while the feature of $s = 4$ is a bit different from the naive matrix multiplication. Considering the feature representation in Fig. 11 and the tradeoff between accuracy and time consumption in Table 2, we select $s = 2$ for the full model.

Table 2
Detailed results of lite matrix multiplication with different polyphase decomposition factor s on Sintel training *clean* and *final* pass dataset on AEE and frame rate, and structural similarity index (SSIM) of context features in level 4 between lite multiplication and naive multiplication. (Inference on Intel Core i5 CPU and NVIDIA GEFORCE 1080 Ti GPU for the frame rate.)

s	AEE/SSIM (*Clean*)	AEE/SSIM (*Final*)	Runtime (fps)
1	2.407/—	3.588/—	20
2	2.412/0.9765	3.601/0.9982	22
4	2.515/0.9061	3.856/0.8990	25

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

As shown in Table 3, compared with state-of-the-art methods, we achieve the comparable quantity results in Sintel and KITTI datasets. Some samples of visualization results are shown in Fig. 12. STC-Flow performs better on AEE among the methods on Sintel *Clean* and *Final* passes, with the decrease of [3.12, 3.49], [0.64, 0.87], [1.02, 0.51] and [0.87, 0.17] compared with SpyNet, FlowNet2, LiteFlowNet and PWC-Net, respectively, and on KITTI 2012/2015 datasets with the decrease of [2.6, 27.08%], [0.3, 3.49%], [0.1, 1.39%] and [0.2, 1.61%], respectively. We can see that the finer details are well preserved via context modeling of spatial and temporal long-range relationships, with fewer artifacts and lower end-point error. In addition, our method is based on only two frames without additional information (like occlusion maps for IRR [25] and additional datasets) used, but it outperforms state-of-the-art multi-frames methods, *e.g.* SelFlow [26]. In addition, Fig. 13 shows the tradeoff chart between accuracy, *i.e.*, average end-point error (AEE) in Sintel test *Clean* pass and the number of CNN parameters. STC-Flow reaches the best balance between accuracy and network size among CNN models for optical flow estimation. Specifically, STC-Flow is lightweight with far fewer parameters, *i.e.* 9M instead of 163M of FlowNet2 [14] and 40M of HD^3 [24]. We believe that our flexible scheme is helpful to achieve better performance for other baseline networks, including multi-frame based methods.

In addition, our method would compromise its performance when the network is trained using FlyingChairs and FlyingThings3D datasets, such as FlowNet2 for Sintel, and EpicFlow for KITTI. FlowNet2 is a large network architecture with a greater number of parameters (160M) than ours (9M), containing several sub-networks (FlowNetC, FlowNetS, and etc.). For the Sintel dataset, FlowNet2 model has higher fitting capability of the small and moderate displacements. For KITTI datasets, patch-matching based methods, like EpicFlow and PatchFlow,

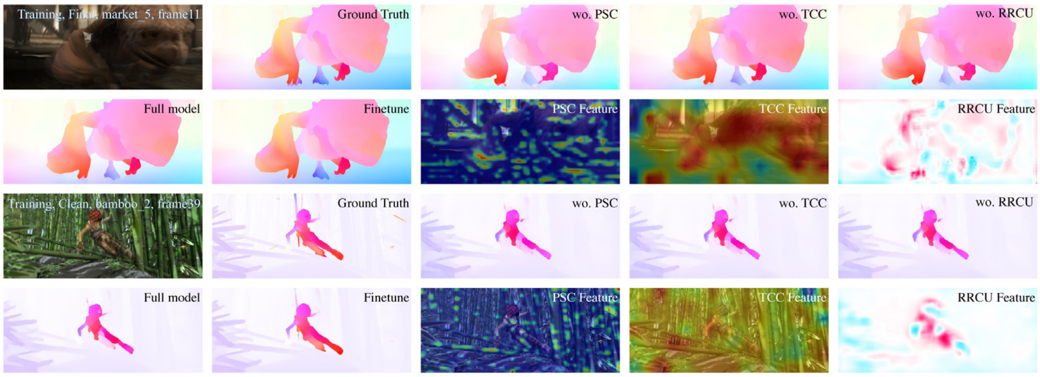

Fig. 10. Results of ablation study on Sintel training *Clean* and *Final* passes. We also indicate the learned features on corresponding modules — PSC and RRCU in level 4 and TCC in level 6. (Zoom in for details.)

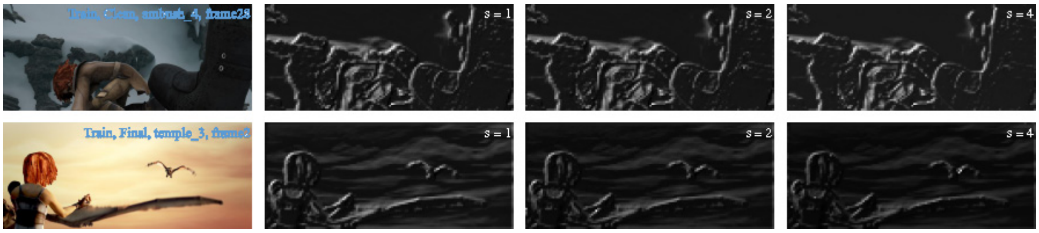
Train, Clean, ambush_4, frame28
s = 1
s = 2
s = 4
Train, Final, temple_3, frame2
s = 1
s = 2
s = 4

Fig. 11. Comparison of feature visualization at level 3 with different polyphase decomposition factors of lite matrix multiplication.

are suitable to handle the large displacements. Moreover, parameters of EpicFlow and DeepFlow are optimized on the training set, and thus the performance would be better. Occlusion branch is used to train PatchFlow, which estimates the occlusion mask for the explicit auxiliary of optical flow estimation. However, since the potential contextual modeling, our performance is better when the network is finetuned using corresponding training datasets. Specifically, IRR [25],

Table 3
AEE and Fl-all of different methods on Sintel and KITTI . The "-ft" suffix denotes the fine-tuned networks using the target dataset. The values in parentheses are the results on the data they were trained on, and hence are not directly comparable to the others. Methods in gray are used with auxiliary utilization.

Methods	Sintel *Clean*		Sintel *Final*		KITTI 2012		KITTI 2015		
	train AEE	test AEE	train AEE	test AEE	train AEE	test AEE	train AEE	train Fl-all	test Fl-all
DeepFlow [9]	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%
Deep DiscreteFlow [18]	–	3.86	–	5.73	–	3.4	–	–	21.17%
Patch Matching [17]	–	3.78	–	5.36	–	3.0	–	–	19.44%
3DFlow [28]	–	3.92	–	6.13	–	–	–	–	26.19%
DCFlow+KF2 [27]	(2.07)	3.65	(3.25)	5.07	–	–	–	–	–
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 [14]	**2.02**	3.96	**3.54**	6.02	4.01	–	10.08	29.99%	–
FlowNet2-ft [14]	(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%
PWC-Net+KF2 [27]	(1.75)	3.75	(2.28)	4.98	–	–	–	–	–
STC-Flow (Ours)	2.41	4.25	3.60	5.56	4.20	3.7	10.18	32.23%	31.59%
STC-Flow-ft (Ours)	(1.36)	**3.52**	(1.73)	**4.87**	(0.98)	**1.5**	(1.46)	(5.43%)	**7.99%**
SelFlow-ft[a] [26]	(1.68)	3.74	(1.77)	4.26	(0.76)	1.5	(1.18)	–	8.42%
PatchFlow[b] [20]	(4.45)	7.7	(4.99)	7.98	3.34	4.0	6.91	21.82%	23.46%
IRR-PWC-ft[b] [25]	(1.92)	3.84	(2.51)	4.58	–	–	(1.63)	(5.32%)	7.65%
MaskFlownet-ft[b] [29]	–	2.52	–	4.17	–	1.1	–	–	6.11%
ScopeFlow-ft[b] [30]	–	3.59	–	4.10	–	1.3	–	–	6.82%
RAFT-ft[c] [31]	(1.09)	2.77	(1.53)	3.61	–	–	(1.07)	(3.90%)	6.30%

[a] SelFlow utilizes 7 frames as the input to estimate optical flow.
[b] PatchFlow, IRR-PWC and MaskFlownet utilize the occlusion supervision as the auxiliary of optical flow estimation.
[c] RAFT utilizes an update operator for multi-frame optical flow training.

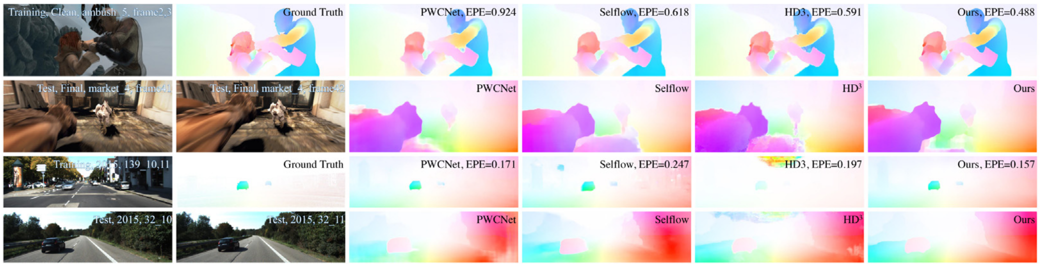

Fig. 12. Examples of predicted optical flow from different methods on Sintel and KITTI datasets. Our method achieves the better performance and preserves the details with fewer artifacts. (Zoom in for details.)

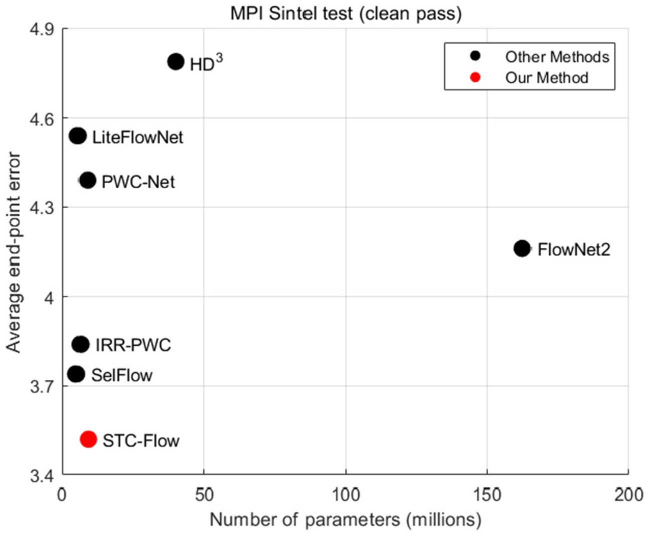

Fig. 13. The performance about the average end-point error (AEE) in Sintel test *Clean* pass and the model size of CNNs.

Table 4
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 [14]	163	105.5	10
LiteFlowNet [23]	5.4	72.1	14
PWC-Net [4]	8.8	37.2	27
IRR-PWC-ft [25]	6.4	182.2	5.2
RAFT-ft [31]	5.3	106.7	9.4
STC-Flow (Ours)	9.0	45.5	22

MaskFlownet [29], and ScopeFlow [30] are utilized the occlusion supervision branch as the auxiliary prediction, which promotes the accuracy of the non-occluded optical flow prediction. SelFlow [26] and RAFT [31] leverage multi-frame training strategies for more information complementation than two-frame scheme. SelFlow uses 7 frames to predict one flow map. RAFT uses an update operator for multi-frame training, to deposit and update optical flow by using the current estimation result, which is helpful to train the network from consecutive frames.

We measure the running time 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 4, our method is more than 2 times faster than FlowNet2, and 1.5 times faster than LiteFlowNet. Due to the calculation of contextual modules, our method is slower than PWC-Net. However, our method models the global contextual information via these modules for accurate optical flow estimation. Moreover, IRR and RAFT require more running time, due to the extra occlusion estimation for IRR and high-resolution feature used for RAFT, respectively.

5. Conclusion

To explore the motion context information for accurate optical flow estimation, we propose a spatio-temporal context-aware network, STC-Flow, for optical flow estimation. We propose three context modules in feature extraction, correlation, and optical flow reconstruction, *i.e. pyramidal spatial context (PSC) module*, *temporal context correlation (TCC) module*, and *recurrent residual contextual upsampling (RRCU) module*, respectively. These three modules utilize contextual information to deal with long-range dependencies and thus improve the overall performance and maintain high-frequency details of optical flow. We have validated the effectiveness of each component. Our proposed scheme achieves the state-of-the-art performance without multi-frame or additional information used.

CRediT authorship contribution statement

Xiaolin Song: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Yuyang Zhao:** Data curation, Software, Writing – original draft, Investigation. **Jingyu Yang:** Supervision, Writing – review & editing.

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.

Acknowledgment

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, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[2] A. Dosovitskiy, P. Fischer, E. Ilg, et al., Flownet: Learning optical flow with convolutional networks, in: IEEE International Conference on Computer Vision, 2015.
[3] A. Ranjan, M.J. Black, Optical flow estimation using a spatial pyramid network, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[4] D. Sun, X. Yang, M.-Y. Liu, et al., PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[5] B.K. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1981) 185–203.
[6] T. Brox, A. Bruhn, N. Papenberg, et al., High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, 2004.
[7] T. Brox, J. Malik, Large displacement optical flow: Descriptor matching in variational motion estimation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2010) 500–513.

[8] H. Zimmer, A. Bruhn, J. Weickert, Optic flow in harmony, Int. J. Comput. Vis. 93 (2011) 368–388.

[9] P. Weinzaepfel, J. Revaud, Z. Harchaoui, et al., DeepFlow: Large displacement optical flow with deep matching, in: IEEE International Conference on Computer Vision, 2013.

[10] Z. Tu, V.D.A. Nico, C. Van Gemeren, R.C. Veltkamp, A combined post-filtering method to improve accuracy of variational optical flow estimation, Pattern Recognit. 47 (2014) 1926–1940.

[11] Z. Tu, R. Poppe, R.C. Veltkamp, Weighted local intensity fusion method for variational optical flow estimation, Pattern Recognit. 50 (2016).

[12] Z. Tu, W. Xie, J. Cao, C. Van Gemeren, R. Poppe, R.C. Veltkamp, Variational method for joint optical flow estimation and edge-aware image restoration, Pattern Recognit. 65 (2017) 11–25.

[13] 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, http://dx.doi.org/10.1109/TCSVT.2020.2974490.

[14] E. Ilg, N. Mayer, T. Saikia, et al., FlowNet 2.0: Evolution of optical flow estimation with deep networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[15] J. Thewlis, S. Zheng, P.H.S. Torr, et al., Fully-trainable deep matching, in: British Machine Vision Conference, 2016.

[16] D. Gadot, L. Wolf, PatchBatch: A batch augmented loss for optical flow, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[17] C. Bailer, K. Varanasi, D. Stricker, CNN-based patch matching for optical flow with thresholded hinge embedding loss, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[18] F. Güney, A. Geiger, Deep discrete flow, in: Asian Conference on Computer Vision, 2016.

[19] 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 (2019) 3595–3607, http://dx.doi.org/10.1109/TCSVT.2018.2885246.

[20] Z. Ren, J. Yan, X. Yang, A.L. Yuille, H. Zha, Unsupervised learning of optical flow with patch consistency and occlusion estimation, Pattern Recognit. 103 (2020) 107191.

[21] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, W. Xu, Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[22] P. Liu, I. King, M.R. Lyu, J. Xu, Ddflow: Learning optical flow with unlabeled data distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

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

[24] Z. Yin, T. Darrell, F. Yu, Hierarchical discrete distribution decomposition for match density estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[25] J. Hur, S. Roth, Iterative residual refinement for joint optical flow and occlusion estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[26] P. Liu, M.R. Lyu, I. King, et al., SelFlow: Self-supervised learning of optical flow, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[27] W. Bao, X. Zhang, L. Chen, Z. Gao, Kalmanflow 2.0: Efficient video optical flow estimation via context-aware Kalman filtering, IEEE Trans. Image Process. 28 (2019) 4233–4246, http://dx.doi.org/10.1109/TIP.2019.2903656.

[28] J. Chen, Z. Cai, J. Lai, X. Xie, A filtering-based framework for optical flow estimation, IEEE Trans. Circuits Syst. Video Technol. 29 (2019) 1350–1364, http://dx.doi.org/10.1109/TCSVT.2018.2805101.

[29] S. Zhao, Y. Sheng, Y. Dong, E.I.-C. Chang, Y. Xu, MaskFlownet: Asymmetric feature matching with learnable occlusion mask, in: CVPR, 2020.

[30] A. Bar-Haim, L. Wolf, ScopeFlow: Dynamic scene scoping for optical flow, in: CVPR, 2020.

[31] Z. Teed, J. Deng, Raft: Recurrent all-pairs field transforms for optical flow, in: ECCV, 2020.

[32] K. Pauwels, N. Krüger, M. Lappe, F. Wörgötter, M.M. Van Hulle, A cortical architecture on parallel hardware for motion processing in real time, J. Vis. 10 (2010) 18.

[33] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A.J. Rodriguez-Sanchez, L. Wiskott, Deep hierarchies in the primate visual cortex: What can we learn for computer vision? IEEE Trans. Pattern Anal. Mach. Intell. 35 (2012) 1847–1871.

[34] S. Tschechne, R. Sailer, H. Neumann, Bio-inspired optic flow from event-based neuromorphic sensor input, in: IAPR Workshop on Artificial Neural Networks in Pattern Recognition, Springer, 2014, pp. 171–182.

[35] M. Chessa, S.P. Sabatini, F. Solari, A systematic analysis of a V1–MT neural model for motion estimation, Neurocomputing 173 (2016) 1811–1823.

[36] F. Solari, M. Caramenti, M. Chessa, P. Pretto, H.H. Bülthoff, J.-P. Bresciani, A biologically-inspired model to predict perceived visual speed as a function of the stimulated portion of the visual field, Front. Neural Circuits 13 (2019) 68.

[37] H. Hu, J. Gu, Z. Zhang, et al., Relation networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[38] J. Fu, J. Liu, H. Tian, et al., Dual attention network for scene segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[39] Y. Cao, J. Xu, S. Lin, et al., GCNet: Non-local networks meet squeeze-excitation networks and beyond, in: IEEE International Conference on Computer Vision Workshop, 2019.

[40] X. Wang, R. Girshick, A. Gupta, et al., Non-local neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[41] 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 (2020) 495–508, http://dx.doi.org/10.1109/TCSVT.2019.2890861.

[42] 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, http://dx.doi.org/10.1109/TCSVT.2019.2943140.

[43] S.I. Young, A.T. Naman, D. Taubman, Graph Laplacian regularization for robust optical flow estimation, IEEE Trans. Image Process. 29 (2020) 3970–3983, http://dx.doi.org/10.1109/TIP.2019.2945653.

[44] W. Shi, J. Caballero, F. Huszar, et al., Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[45] D.J. Butler, J. Wulff, G.B. Stanley, et al., A Naturalistic Open Source Movie for Optical Flow Evaluation, in: European Conference on Computer Vision, 2012.

[46] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[47] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[48] N. Mayer, E. Ilg, P. Hausser, et al., A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.