Global Matching with Overlapping Attention for Optical Flow Estimation Supplementary Document

Shiyu Zhao[1,*] Long Zhao[2] Zhixing Zhang[1] Enyu Zhou[3] Dimitris Metaxas[1]

[1]Rutgers University [2]Google Research [3]SenseTime Research

Abstract

In this supplementary document, we elaborate the architecture of our large context feature extraction module and describe the optimization stage in detail. Additionally, we provide more visual results to further justify the effectiveness of the proposed GMFlowNet and describe how we illustrate the cost volume.

1. Architecture Details

1.1. Large Context Feature Extraction

In the paper, we exploit Transformer blocks to extract large context features to improve the matching step in GMFlowNet. In the original Transformer block [5], input features are updated by a Multi-head Self-Attention (MSA) followed by a Multilayer perceptron (MLP). MSA is able to extract the long-term dependency, and MLP projects the features to the required dimension. Both MSA and MLP calculate residuals that are added to the input features as the output features. The update in a transformer block can be formulated as,

$$\begin{aligned} \hat{x}^l &= \text{MSA}(\text{LN}(x^{l-1})) + \hat{x}^l \\ x^l &= \text{MLP}(\text{LN}(\hat{x}^l)) + \hat{x}^l, \end{aligned} \tag{1}$$

where LN refers to layer norm, and x^{l-1} and x^l represent output features of the previous block and the current block, respectively. The MSA is originally designed for language tasks and takes the whole 1D features as input, but it is computationally prohibitive to apply it on 2D feature maps for optical flow estimation. To extract the long-term dependency with an acceptable computation cost, we propose the patch-based overlapping attention (POLA) to replace MSA of the original attention block and call our attention block as multi-head POLA (M-POLA).

In our large context feature extraction module (Section 3.1), we take 3 convolutional layers (3-Convs) to extract initial features and 6 M-POLA blocks to extract large context information based on initial features. The detailed structure of this module is listed in Table 1.

	Layer name(s)
3-Convs	Conv(3, 64, 7, 2), ReLU
	Conv(64, 128, 3, 2), ReLU
	Conv(128, 256, 3, 2), ReLU
6 M-POLA	M-POLA (dim=256, head=8, win_size=7)
	M-POLA (dim=256, head=8, win_size=7)
	M-POLA (dim=256, head=8, win_size=7)
	M-POLA (dim=256, head=8, win_size=7)
	M-POLA (dim=256, head=8, win_size=7)
	M-POLA (dim=256, head=8, win_size=7)

Table 1. Large context feature extraction. The arguments in Conv(·) are the input channel number, the output channel number, the kernel size, and the convolution stride, respectively.

1.2. Optimization Network

We adopt the iterative update operator proposed in RAFT [4] as the optimization step of GMFlowNet. As stated in [4], this operator mimics the steps of an optimization algorithm and iteratively outputs a series of flow predictions $\{f_{1\to2}^{(1)}, f_{1\to2}^{(2)}, \ldots, f_{1\to2}^{(T)}\}$. For the t-th iteration, the flow prediction $f_{1\to2}^{(t)}$ is calculated by a Convolutional GRU [2] (ConvGRU) as,

$$\begin{aligned} x^{(t)} &= [f_{1\to2}^{(t-1)}, F_1, \text{lookup}(C, f_{1\to2}^{(t-1)}, r)], \\ r^{(t)} &= \sigma(\text{Conv}([h^{(t-1)}, x^{(t)}])), \\ \widetilde{h}^{(t)} &= \sigma(\text{Conv}([r^{(t)} \odot h^{(t-1)}, x^{(t)}])), \\ z^{(t)} &= \mu(\text{Conv}([h^{(t-1)}, x^{(t)}])), \\ h^{(t)} &= (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot \widetilde{h}^{(t)}, \\ \Delta f_{1\to2}^{(t)} &= \text{Conv}(h^{(t)}), \\ f_{1\to2}^{(t)} &= f_{1\to2}^{(t-1)} + \Delta f_{0\to1}^{(t)} \end{aligned} \tag{2}$$

*Correspondence to: Shiyu Zhao (sz553@rutgers.edu).

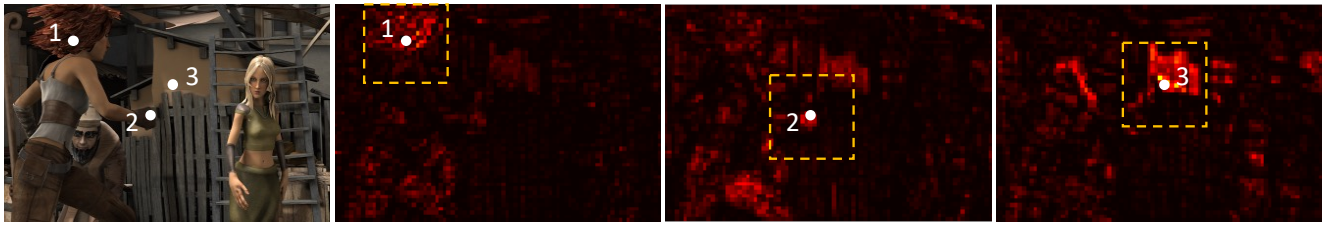


Figure 1. Visualization of attention scores. The more red a pixel is, the higher the score is.

where F_1 is the context features, C is the 4D cost volume (See Section 3.2 of the paper), Conv$(\cdot)$ refers to a convolution layer, $\sigma(\cdot)$ means sigmoid, and $\mu(\cdot)$ means tanh. lookup$(\cdot)$ represents the cost volume within the range of r. For each location $\mathbf{x}$ in I_1, lookup$(\cdot)$ is defined as,

$$\text{lookup}(\cdot) = \{C(\mathbf{x}, f_{1\to 2}^{(t-1)}(\mathbf{x}) + \delta\mathbf{x}) \mid r > \| \delta\mathbf{x} \|_1\}. \quad (3)$$

Different iterations share the weights in the ConvGRU.

2. More Visualizations

2.1. Attention maps

Fig. 1 visualizes full attention score maps of the first POLA for three pixels highlighted in white. The more red a pixel is, the higher the score is. Yellow dash boxes indicate the local regions that are used in POLA. As shown, a pixel is more likely to attend to those that are visually similar to the pixel.

2.2. Coarse Flows

Figure 2 displays the coarse flows from our matching step as well as the final flow estimation for samples from Sintel [1] and KITTI [3] datasets. We compare our GMFlowNet with RAFT [4] because they share the same optimization architecture. For Sintel, both models are trained on C+T. For KITTI, they are trained on all the training data. As shown, the coarse flow results in better predictions especially in large motion areas and textureless regions. For example, the hand of the character in Fig. 2b moves fast, leading to failures of RAFT. On the contrary, our matching step finds the optical flow for the hand and improves the final prediction.

2.3. More Visual Results

Figure 3 provides the qualitative evaluation of GMFlowNet and RAFT on the Sintel test set. We highlight with white arrows and red dash boxes the regions where our method outperforms RAFT. Fig. 4 exhibits the visualization of more samples from the KITTI test set. Red dash boxes highlights the regions where our method outperforms RAFT.

3. How We Visualize Cost Volumes

In order to compare the $4D$ cost volumes C of RAFT [4] and our method, we extract the matrix $F_{x,y}$ as the matching matrix for the point (x, y),

$$F_{x,y} = \text{softmax}(C[x, y, (x + \delta x - 40) : (x + \delta x + 40), (y + \delta y - 40) : (y + \delta y + 40)]) \quad (4)$$

where δx and δy are indicated by the ground truth flow at (x, y). The symbol $C[\cdot]$ means to fetch values from C within a given range. Then, we average $F_{x,y}$ on all points within a specific displacement range for all images in Sintel and visualize the averaged matching matrix.

We visualize the cost volume for different ranges of displacements in Fig. 5. The larger the value at the center of the averaged matching matrix is, the higher quality the cost volume has. As shown, GMFlowNet outperforms RAFT in all displacement ranges, which indicates that our approach provides better cost volumes not only for small displacements but also for large ones.

References

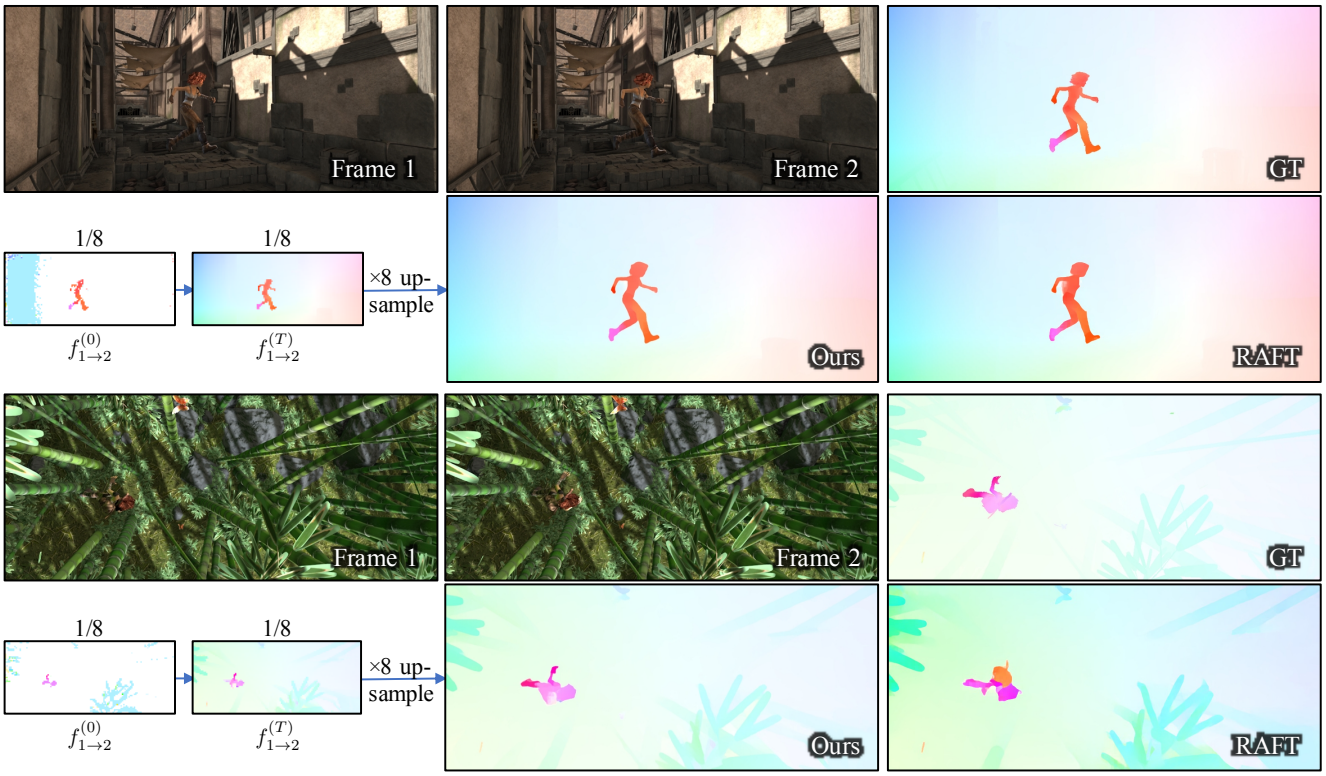
[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625. Springer, 2012. 2, 3, 4

[2] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 1

[3] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, pages 3061–3070, 2015. 2, 3, 5

[4] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 1, 2, 3, 5

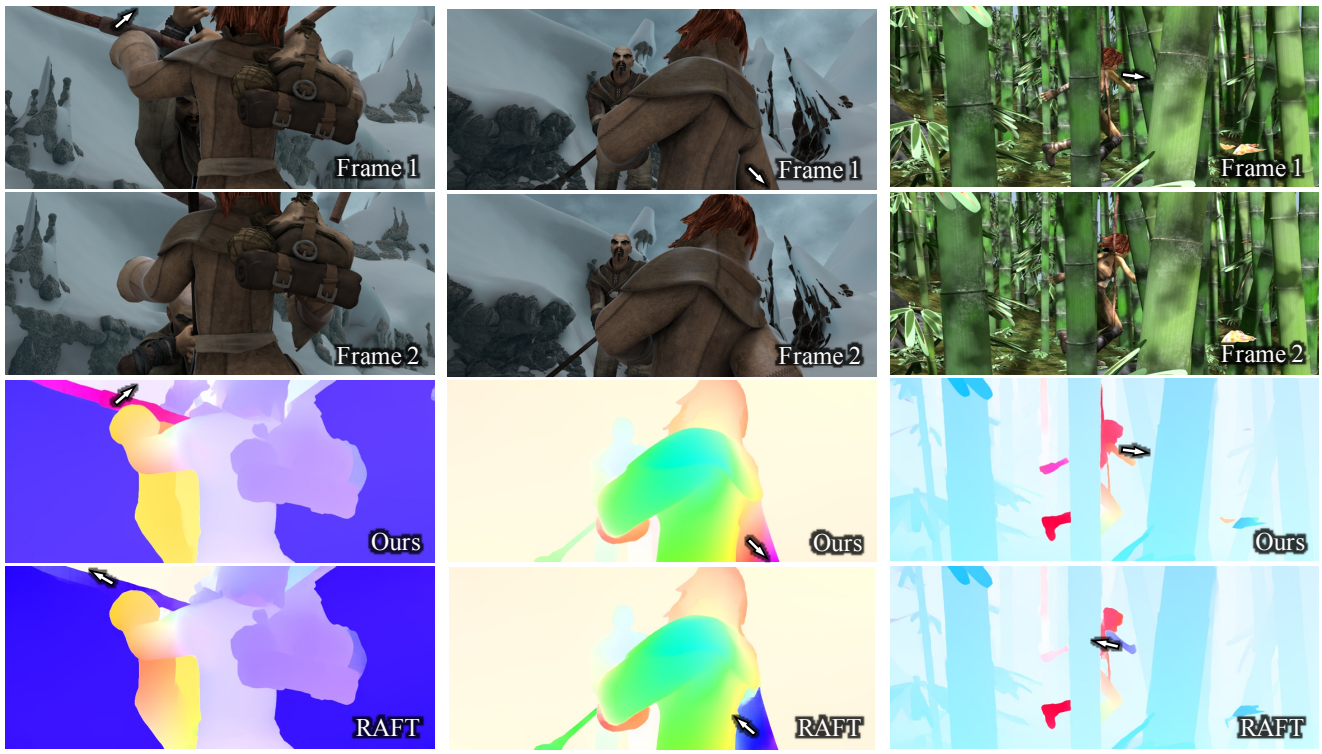
[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1



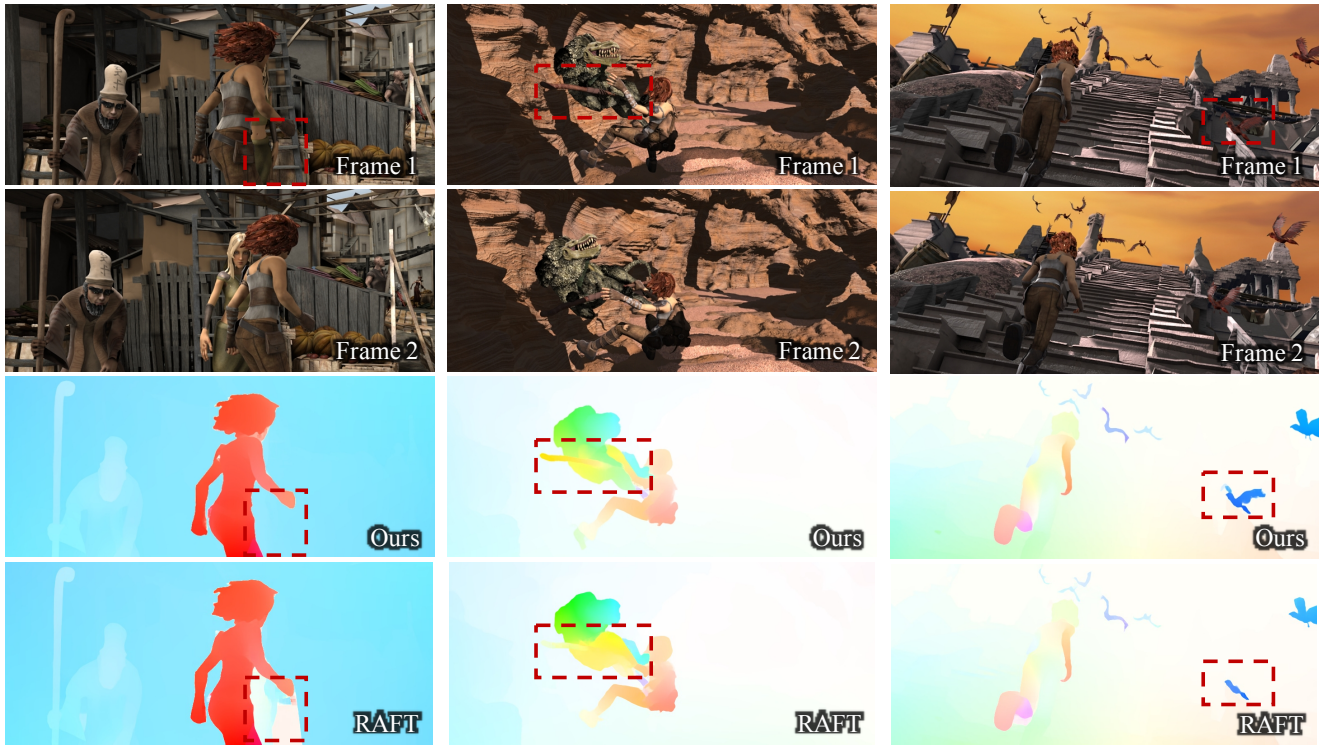
(a) Samples from Sintel training set [1]

(b) Samples from KITTI test set [3]

Figure 2. **Visualizations of coarse flow.** For (a) Sintel, models are trained on C+T. For (b) KITTI, models are trained on C+T+S+K+H. Ground-truth flows for KITTI are unavailable and thus are not shown. With the coarse flow, our method outperforms the most popular optimization-only method RAFT [4]. Red dash boxes highlight the main differences between RAFT's predictions and ours.

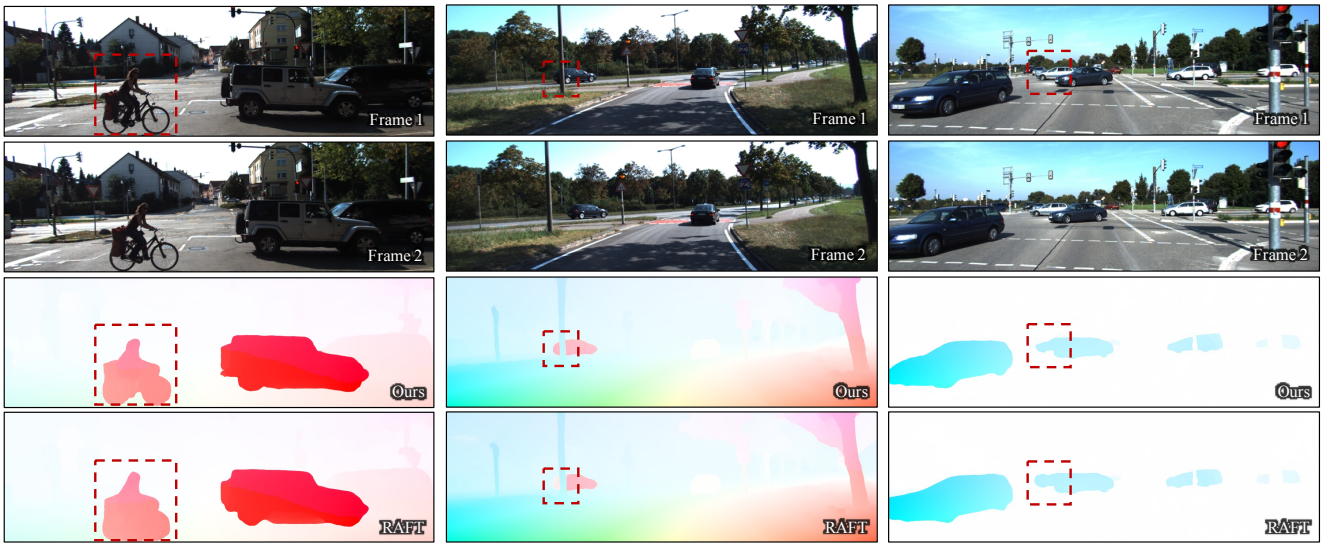


(a)



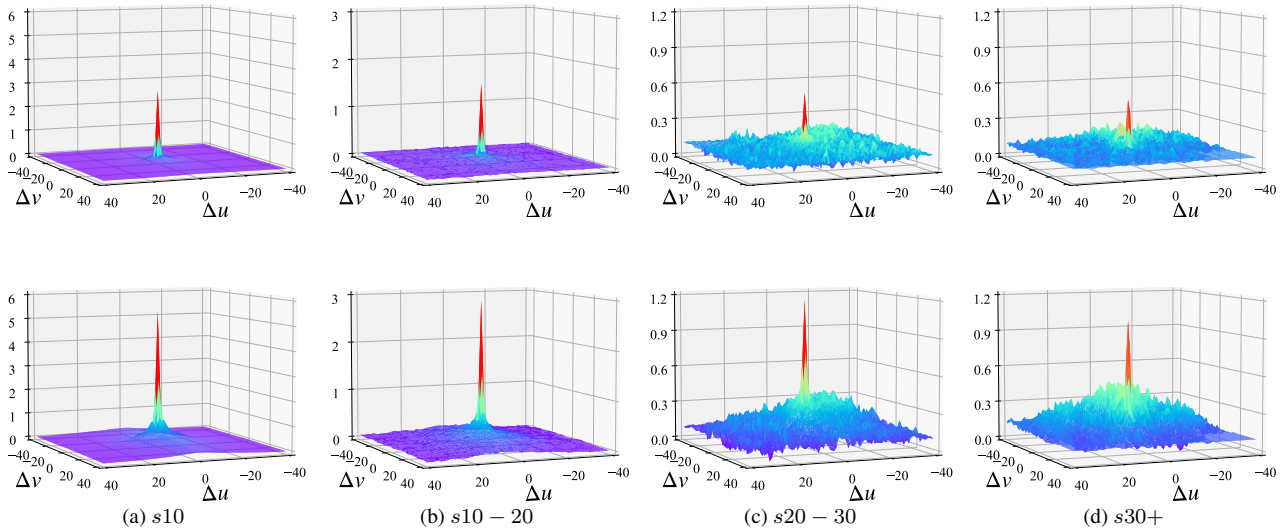
(b)

Figure 3. **Qualitative evaluation** on the Sintel test set [1]. White arrows in (a) and red dash boxes in (b) highlight the differences between our method and RAFT. Ground-truth optical flows are not available and are not shown. Models are trained on the same training data.



(a)

Figure 4. **Qualitative evaluation** on the KITTI test set [3]. Red dash boxes highlight the differences between our method and RAFT. Models are trained on the same training data.



(a) $s10$ (b) $s10-20$ (c) $s20-30$ (d) $s30+$

Figure 5. **Visualization of cost volumes in different range of displacements.** The first row is for RAFT [4], and the second row is ours. $s10$ refers to regions with displacements below 10 pixels, $s10-20$ for displacements between 10 and 20 pixels, $s20-30$ for displacements between 20 and 30 pixels, and $s30+$ for displacements larger than 30 pixels.