Appendix

1. Motion Estimation

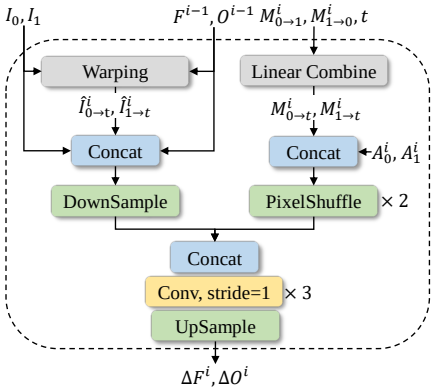

Figure 1. Pipeline of motion estimation.

For motion estimation, we follow a similar design schema in RIFE [1], which directly utilizing simple convolutional layers to iteratively updates the optical flow $\boldsymbol{F}$ for backward warp and the fusion map $\boldsymbol{O}$. In contrast to RIFE, without as many as eleven convolutional layers in each iteration and extra privileged distillation, only three convolutional layers are needed for high-performance motion estimation due to the sufficient information contained in the extracted motion and appearance feature.

The pipeline of our motion estimation is shown in Fig. 1. For the motion and appearance features extracted at i-th Transformer stage, we first acquire $\boldsymbol{M}^i_{0\to t}$ and $\boldsymbol{M}^i_{1\to t}$ by linear scaling $\boldsymbol{M}^i_{0\to 1}$ and $\boldsymbol{M}^i_{0\to 1}$ with t. $\boldsymbol{M}^i_{0\to t}$ and $\boldsymbol{M}^i_{1\to t}$ are then concatenated with $\boldsymbol{A}^i_0$ and $\boldsymbol{A}^i_1$. Due to the resolution of motion and appearance features being quite low, we apply two PixelShuffle [4] with $r = 2$ to quadruple the resolution of those features. To iteratively update $\boldsymbol{F}^{i-1}$ and $\boldsymbol{O}^{i-1}$ estimated in the previous stage, we also combine $\boldsymbol{F}^{i-1}$ and $\boldsymbol{O}^{i-1}$ with the warped original images as extra input to further boost the performance. Then the two stream inputs are concatenated together and fed to three convolution layers to generate the residual. The residual is upsampled by bilinear interpolation to the original resolution of inputs and added to the $\boldsymbol{F}^{i-1}$ and $\boldsymbol{O}^{i-1}$ to synthesize motion at current stage:

$$\boldsymbol{F}^i = \boldsymbol{F}^{i-1} + \Delta\boldsymbol{F}^i, \tag{1}$$

$$\boldsymbol{O}^i = \boldsymbol{O}^{i-1} + \Delta\boldsymbol{O}^i \tag{2}$$

2. RefineNet

We adopt a simplified U-Net [3] architecture for refining the warped results $\tilde{\boldsymbol{I}}_t$ obtained with $\boldsymbol{F}$ and $\boldsymbol{O}$, as shown in Fig. 2. The only difference is that we add the acquired low-level features L and inter-frame enhanced appearance features A into the corresponding stage to provide additional appearance information for better appearance refinement.

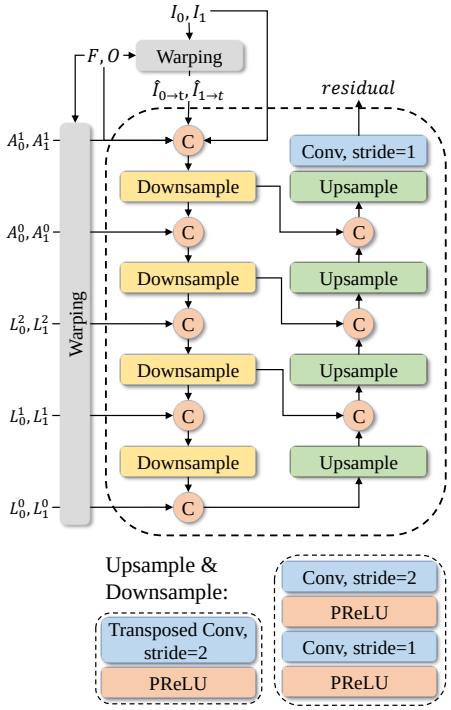

Figure 2. Structure of RefineNet.

3. Loss Functions

The training loss is composed of two parts: warp loss and reconstruction loss. The warp loss is to directly supervise the result $\tilde{\boldsymbol{I}}_t$ obtained by warping and fusing inputs with $\boldsymbol{F}$ and $\boldsymbol{O}$, which implicitly supervise the motion estimation, as:

$$\mathcal{L}^i_{warp} = f(\tilde{\boldsymbol{I}}^i_t, \boldsymbol{I}^{GT}_t)\ , \tag{3}$$

where $\mathcal{L}^i_{warp}$ represents the warps loss for i-th motion estimation, $\boldsymbol{I}^{GT}_t$ is the ground truth, f is usually a pixel-wised loss. Following previous work [2], we employ the Laplacian loss, which denotes the L_1 loss between the Laplacian pyramids of the warped frame and the ground truth, as f. The reconstruction loss is to supervise the reconstruction quality of the final synthesized frame, as:

$$\mathcal{L}_{rec} = f(\hat{\boldsymbol{I}}_t, \boldsymbol{I}^{GT}_t)\ . \tag{4}$$

The full loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \sum_i \mathcal{L}^i_{warp}\ , \tag{5}$$

where λ is the loss weight for warp loss, we set $\lambda = 0.5$ to maintain the balance between losses.

4. Detailed Runtime/Memory Comparisons

More comparisons, conducted on the 2080 Ti, are provided in Tab. 1. Our method shows efficiency compared to high-performance models (VFIFormer and ABME), and Ours-small is comparable to real-time models (AdaCoF).

Table 1. More Runtime/Memory Comparisons.

Input	Ours	VFIFormer	ABME	Ours-small	AdaCoF
256×256	56ms/1.49GB	214ms/2.41GB	84ms/1.50GB	13ms/1.14GB	6ms/1.19GB
512×512	132ms/2.01GB	892ms/6.13GB	206ms/2.20GB	25ms/1.42GB	21ms/1.58GB

5. Affect of window size

As shown in Tab. 2, 7 is a decent choice for the attention window size.

Table 2. Affect of window size.

Winow Size	Vimeo90k	Xiph-2k	Xiph-4k
5	36.04/**0.9797**	36.40/0.9418	34.21/0.9015
7	**36.07/0.9797**	**36.55/0.9421**	**34.25/0.9019**
9	36.05/0.9795	36.53/0.9420	34.18/0.9011

References

[1] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 1

[2] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 1

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[4] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 1