Supplementary: Error Compensation Framework for Flow-Guided Video Inpainting

Jaeyeon Kang[1], Seoung Wug Oh[2], and Seon Joo Kim[1]

[1]Yonsei University, [2]Adobe

1 Network Details

Our local temporal network LTN and the error compensation network ECN have a similar structure (encoder, multi spatio-temporal transformer layers [5,9] and decoder), although the role of the two is obviously different. The main reason behind the design is that both networks share the commonality that the information must be obtained by deriving correlations with other frames. We found that the spatial-temporal transformer network satisfies the needs of both tasks, as it has a strong ability to learn the spatio-temporal relationship (both intra and inter frames), making the results more temporally coherent. The structural design of our network is shown in Fig. 1.

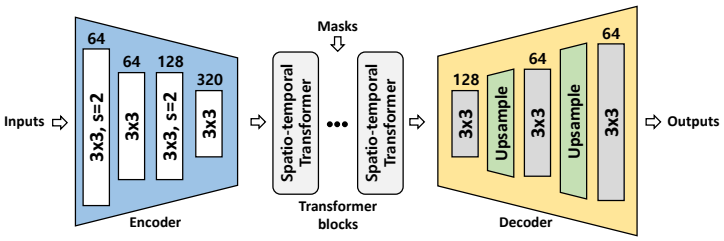


Fig. 1: Network details. We use vanilla convolution in the encoder and gated convolution [8] in decoder. s indicates stride. Note that all the inputs are forwarded to the same encoder.

Encoder. All inputs are forwarded to the same encoder. We use vanilla convolution as a basic building block. The encoder downsamples the input feature map up to 1/4 scale of the original size to enlarge the receptive field.

Transformer layers. The transformer layers merge the information from the encoded features in the deep encoding space. As similar patches are widely distributed in inter-intra frames, we set various patch sizes. We extract patches by reducing the original size by scale factors s. In our experiments, five different scale factors that increase in multiples of two are used. We use 8 transformer blocks in error compensation network ECN. A detailed illustration is shown in

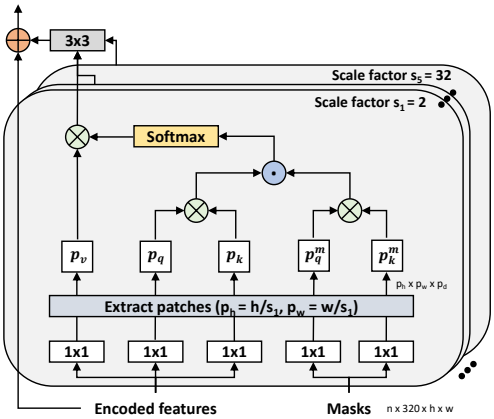


Fig. 2: Our spatio-temporal transformer layers in the error compensation network. p_h, p_w denote patch size of height and width respectively. p_d is the total number of patches. We extract patches by reducing the original size by scale factor s. In our experiments, five different scale factors that increase in multiples of two are used. $\otimes, \oplus, \odot$ represent the matrix multiplication, element-wise addition and element-wise multiplication respectively.

Fig. 2. The propagation masks m^p and remaining masks m^r are concatenated along the channel axis before being fed into the transformer layers.

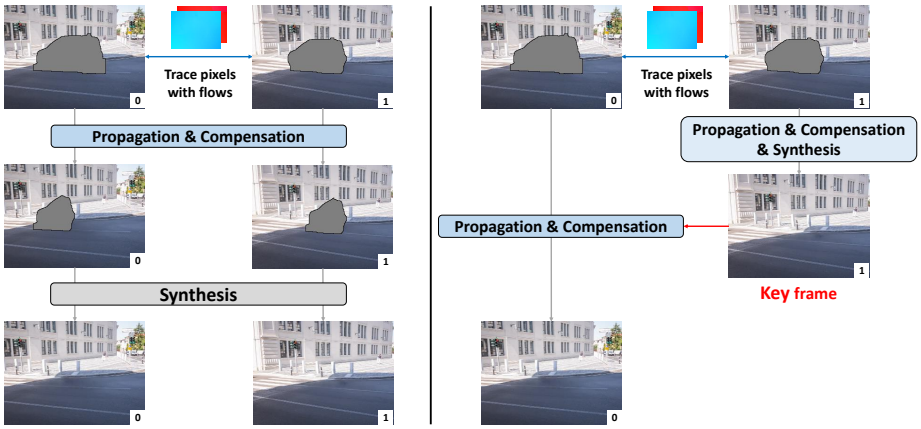
On the other hand, we use 4 transformer blocks in the local temporal network LTN. Since the locally coherent frames $\bar{x}$ aim to guide the flow completion, there is no need to design the network deeply to restore fine details. Note that actual inpainting results are obtained in later stages. In the local temporal network, only the original masks m are forwarded into the transformer blocks.

Decoder. The decoder takes the output of the transformer layers to reconstruct results. Inspired by [8], we use gated convolution in the decoder. The nearest neighbor upsampling is used to generate the same resolution with the input's size.

Discriminator. For discriminator, we use the same network design from Temporal PatchGAN [2] composed of six 3D convolution layers. We use compensated frames $\tilde{y}$ and ground-truth frames $\hat{y}$ as fake and real for training the discriminator.

2 Entire Process for Video Inpainting

After all frames are filled with propagation with compensation stages, there may be remaining regions to fill. It is mainly due to occlusion, where pixels cannot be traced with the guidance of completed flows. Like the naïve approach in Fig. 3, the problem is that if every frame goes through the synthesis after propagation,



(a) Naïve approach (b) Using key frames

Fig. 3: The entire process for video inpainting with (a) naïve approach and (b) using key frames. For simplicity, we only show the two frames x_0 and x_1. In (a), the two compensated frames are forwarded to the synthesis network after propagation. On the other hand, in (b), we select the key frame and complete using our full procedures. Then, the remaining frame (non-key frame) is completed using the key frame with propagation and compensation. In our experiment, about 4 frames in total 50 frames are defined as key frames and forwarded to the synthesis network after propagation.

it may lead to temporal inconsistency due to a sliding window manner to inpaint entire frames.

Instead, we have observed that some corrupted frames can be inpainited using only specific completed frames. Here, we define the specific frames as key frames. For inpainting whole frames, we first find those key frames and complete them with our full procedures, including synthesis. Then the filled pixels in key frames are propagated to the remaining corrupted frames (non-key frames) only with iterative propagation and compensation stages. Non-key frames can be filled by referencing only the key frames, which is shown in Fig. 3. The purpose of completing key frames first is to reduce the number of synthesis procedures as much as possible and enhance the temporal consistency.

We define the key frames as frames with large remaining holes that require the synthesis procedure. Specifically, at first, we run the propagation stage only on the masks of all frames. We select the frame with the largest remaining mask as the first key frame. Note that we do not compensate for errors in this process since it only aims to check the remaining regions. Actual inpainting is proceeded after finding all key frames. Then, assuming the first key frame is completed, we set the remaining mask of the first key frame as 0. Now, we may complete other frames only using the key frame since most of the pixels that cannot be filled due to occlusion are synthesized. But if there are still remaining holes on

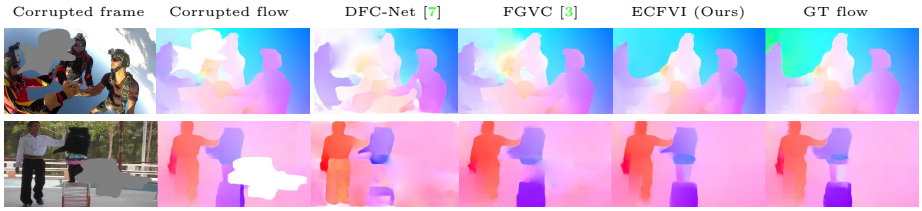


Fig. 4: Flow completion results compared with [3, 7]. The corrupted flows are estimated through the corrupted frames as input to the flow estimator.

	FVI [10]			Youtube-VI	
Mask types	Stationary	Circle	Curve	Stationary	Moving
	VFID/EWarp	VFID/EWarp	VFID/EWarp	VFID/EWarp	VFID/EWarp
FGVC* [3]	0.5323/0.3336	0.6452/0.3269	0.5885/0.3833	0.1411/0.2434	0.0687/0.2407
FuseFormer [5]	0.3122/0.3387	0.5036/0.3371	**0.4345**/0.3625	0.1219/0.2446	0.0890/0.2481
ECFVI(Ours)	**0.2963 / 0.3299**	**0.4678 / 0.3210**	0.4453/**0.3552**	**0.1073 / 0.2407**	**0.0570 / 0.2398**

Table 1: Quantitative evaluation. For both metrics, the lower is better.

some frames, we select another key frame by finding the largest remaining mask again. We iterate until all non-key frames can be filled with only the detected key frames.

3 Experiments

3.1 Comparisons

Besides the PNSR and SSIM metrics, we additionally compare our methods with state-of-the-art methods on other metrics in Table 1. Video-based Fréchet Inception Distance (VFID) [6] is used for scoring the perceptual quality, and optical flow based warping error (EWarp) [4] is used for measuring the temporal consistency. Our method achieves the best performance except for the curve mask, demonstrating that it can produce more visually pleasing results while satisfying the temporal consistency.

3.2 More Results

We show visual comparisons on flow completion in Fig. 4. In [3, 7], the errors in corrupted flow affect the flow completion, resulting in unfaithful motion estimation. On the other hand, we can estimate more accurate motion information by designing our flow completion module to be aware of RGB values.

Additional visual comparisons on video inpainting are shown in Fig. 7 To visualize the temporal consistency, we show the temporal profile [1] of the resulting videos below the completed frames. We recommend watching our demo video.

	Youtube-VI								
	Stationary Mask				Moving Mask				
	PSNR	SSIM	VFID	EWarp	PSNR	SSIM	VFID	EWarp	Time(s)
ECFVI(Naïve)	33.39	0.9784	0.1129	0.2441	36.63	0.9852	0.0601	0.2420	132
ECFVI(Ours)	**33.53**	**0.9795**	**0.1073**	**0.2407**	**36.80**	**0.9860**	**0.0570**	**0.2398**	121

Table 2: An ablation study on Youtube-VI dataset. ECFVI(Naïve) denotes without using key frames.

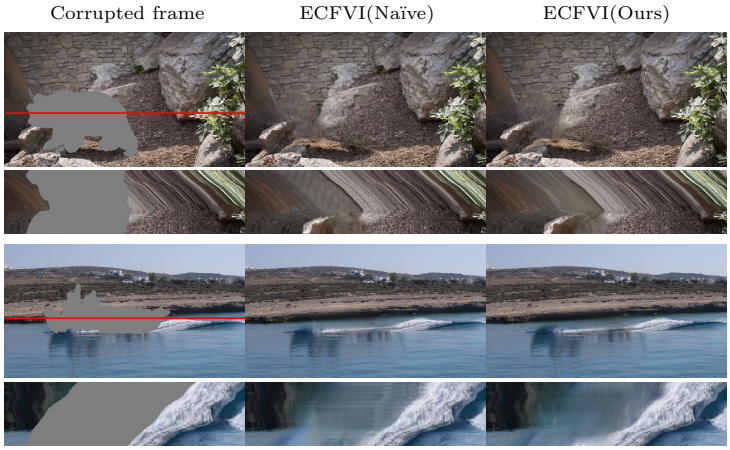


Fig. 5: An ablation study on using key frames. The temporal profiles of the red scan line are shown below the results. **Best viewed in zoom.**

3.3 Ablation study

To show the effectiveness of using key frames, we use the naïve approach in Fig. 3. We first estimate all compensated frames by propagating pixels from reference frames. Then, the compensated frames are forwarded into the synthesis network with a sliding window manner to complete the remaining regions m^r. For example, the compensated frames $\tilde{y}_{0:10}$ are synthesized, then $\tilde{y}_{1:11}$. Although the performance gap is minor in Table 2, the blurry edges and visual artifacts are shown in Fig. 5. In contrast, our model shows sharp and smooth edges since we warp the synthesized pixels with the guidance of flows, which can further guarantee temporally consistent results.

The naïve approach is slightly slower than ours. Because one or two key frames are enough to complete some non-key frames, the number of propagation and compensation iterations reduces.

4 Discussion

The flow-based methods have a core limitation that the results depend on the performance of the flow estimator. As can be seen in Fig. 6, large motion between frames usually results in wrong flow estimation. Although our method can deal

Fig. 6: A failure case due to large motion.

with the error to some extent, it still affects our performance. In the future, we will alleviate the extreme cases by adopting a newly designed flow completion module for video inpainting.

References

1. Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: CVPR. pp. 4778–4787 (2017) 4
2. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: ICCV. pp. 9066–9075 (2019) 2
3. Gao, C., Saraf, A., Huang, J.B., Kopf, J.: Flow-edge guided video completion. In: ECCV. pp. 713–729. Springer (2020) 4, 6, 7
4. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV. pp. 170–185 (2018) 4
5. Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., Wang, X., Dai, J., Li, H.: Fuseformer: Fusing fine-grained information in transformers for video inpainting. In: ICCV. pp. 14040–14049 (2021) 1, 4, 6, 7
6. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. arXiv preprint arXiv:1808.06601 (2018) 4
7. Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. In: CVPR. pp. 3723–3732 (2019) 4
8. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV. pp. 4471–4480 (2019) 1, 2
9. Zeng, Y., Fu, J., Chao, H.: Learning joint spatial-temporal transformations for video inpainting. In: ECCV. pp. 528–543. Springer (2020) 1, 6, 7
10. Zou, X., Yang, L., Liu, D., Lee, Y.J.: Progressive temporal feature alignment network for video inpainting. In: CVPR. pp. 16448–16457 (2021) 4

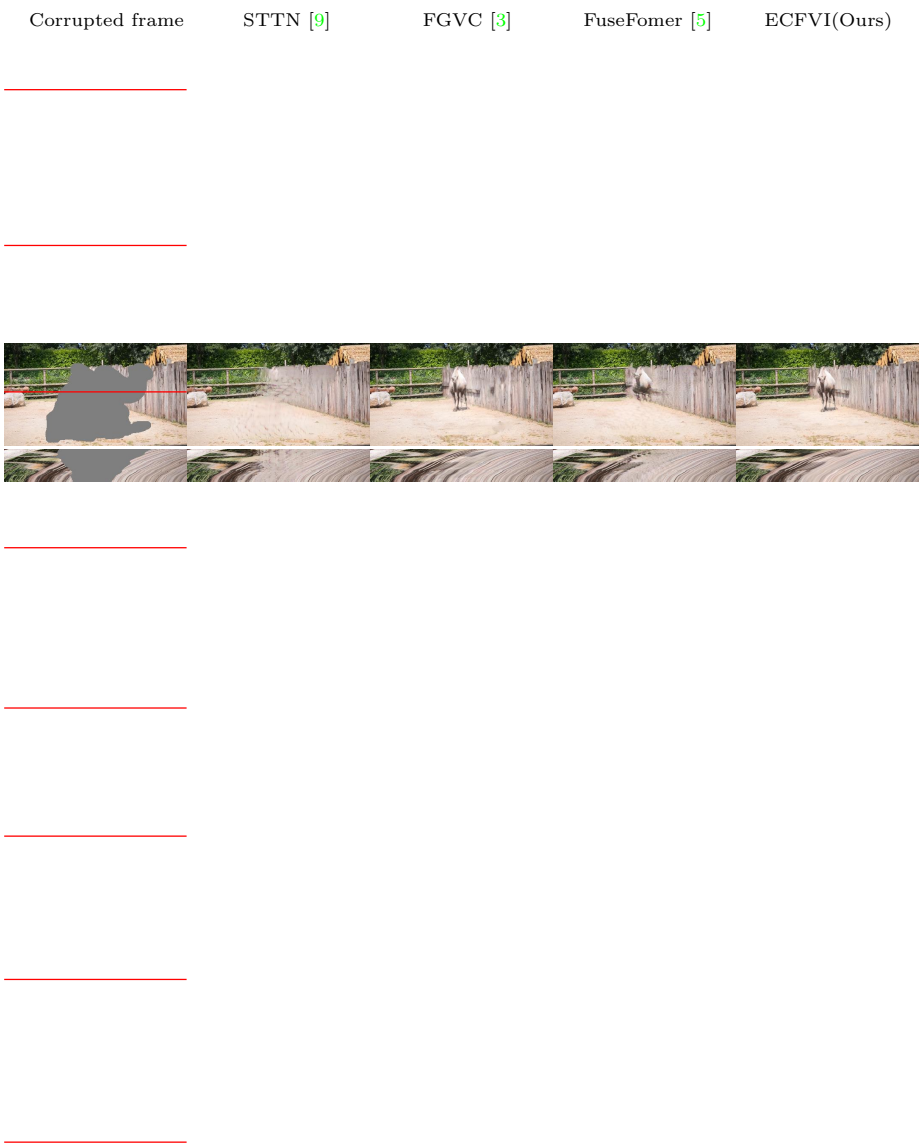


Fig. 7: Qualitative results compared with [3, 5, 9]. The temporal profiles of the red scan line are shown below the results. **Best viewed in zoom.**