Beyond Natural Motion: Exploring Discontinuity for Video Frame Interpolation

Sangjin Lee  Hyeongmin Lee  Chajin Shin  Hanbin Son  Sangyoun Lee

Abstract

Video interpolation is the task that synthesizes the intermediate frame given two consecutive frames. Most of the previous studies have focused on appropriate frame warping operations and refinement modules for the warped frames. These studies have been conducted on natural videos having only continuous motions. However, many practical videos contain a lot of discontinuous motions, such as chat windows, watermarks, GUI elements, or subtitles. We propose three techniques to expand the concept of transition between two consecutive frames to address these issues. First is a new architecture that can separate continuous and discontinuous motion areas. We also propose a novel data augmentation strategy called figure-text mixing (FTM) to make our model learn more general scenarios. Finally, we propose loss functions to give supervisions of the discontinuous motion areas with the data augmentation. We collected a special dataset consisting of some mobile games and chatting videos. We show that our method significantly improves the interpolation qualities of the videos on the special dataset. Moreover, our model outperforms the state-of-the-art methods for natural video datasets containing only continuous motions, such as DAVIS and UCF101.

1. Introduction

Video interpolation is a low-level vision task to generate additional frames to improve video quality. When the time interval of each consecutive input frames is fixed we can get smoother video, and when the frame rate is fixed we can get slow-motion video. This can also be applied to video compression by controlling frame rate [2, 31], view synthesis [8, 14, 36], or other real-world applications [22, 26, 34].

Most of the previous works focus on the motion of the objects in videos. They utilize the estimated flow maps [12, 16], kernels [6, 15, 17, 20, 21], or pre-trained optical flow models [18, 19, 26] to place each object in the middle of its positions on the adjacent frames. However, many of the practical videos contain special objects such as GUI elements and subtitles, which do not move continuously. Also, even the typical videos without those elements contain some special scenarios such as brightness change and haze. Therefore, the concept of transition between the consecutive frames should be expanded further beyond the scope of motion.

In this paper, we propose three techniques to deal with the videos containing both continuous and discontinuous motions. First, we propose novel data augmentation called Figure-Text Mixing (FTM) which consists of Figure Mixing (FM) and Text Mixing (TM). FM is an augmentation of adding fixed random figures and TM is an augmentation of adding discontinuity moving random texts. The networks can learn both continuous and discontinuous motions using FTM without any additional datasets. Second, we propose an architecture based on AdaCoF [15] which can separate the areas of continuous and discontinuous motion given each frame. Our framework estimates a map called discontinuity map which determines whether the motion of each pixel is continuous or discontinuous. Since it is hard to figure out whether the object moves discontinuously by seeing only two frames, we give four frames as inputs to the network. Lastly, if we utilize both FTM and discontinuity map, it is possible to supervise the model by giving the ground-truth of the discontinuity maps. Therefore we propose an additional loss function to guide our model to estimate sharper discontinuity map.

We construct a special dataset called Game-graphic dataset to evaluate how our method and the competitive works deal with the discontinuous motions. Our approach shows significantly improved results compared to the other methods. Our main contributions can be summarized as follows:

• Data Augmentation. We propose a new data augmentation strategy called FTM that can be simply applied to existing video datasets to make models learn both continuous and discontinuous motions.

• New Architecture. We propose a new framework which can separate continuous and discontinuous motions.
• **Performance.** Our proposed network achieves state-of-the-art performance for general video frame interpolation on not only the dataset containing discontinuous motions, but also all other natural test videos.

2. Related Work

Most of existing video frame interpolation algorithms consist of the two parts: motion estimation and motion compensation. Motion estimation modules estimate the pixel level correspondences between two consecutive frames to get motion information. Then motion compensation parts warp the frames according to the estimated motion. The recent video frame interpolation researches utilize deep neural networks (DNN) to get high quality results in two ways.

One is the end-to-end learning approach. Several works train their neural networks which perform both motion estimation and compensation at the same time. Niklaus et al. [20] propose the network that estimate big kernel weights for all pixels of input frames. Then they adaptively convolve the input frames with the estimated kernels to get the output frame. Since excessively many weights are necessary due to the large kernel size, Niklaus et al. [21] solve this problem by using separable kernels. On the other hand, Liu et al. [16] and Jiang et al. [12] propose the neural network that estimates dense flow map which consist of the vectors directly pointing the reference pixels. However, above methods have limitation that the kernel based ones cannot deal with the motion beyond the kernels and the flow based ones only refer to one pixel for each output pixel. To solve the problem, Lee et al. [20] combine the two methods using the deformable convolution [5]. Some approaches propose the neural networks that directly estimate the intermediate frames without motion compensation. Long et al. [17] train a simple U-Net [25] to estimate the intermediate frames, but the results tend to be blurry. Therefore Choi et al. [4] propose a new architecture based on channel attention to get the sharper results.

The other is the optical flow-based approach. Recently, lots of approaches to estimate high-quality optical flow maps have been introduced [11, 28, 29]. Therefore, several works make use of the optical flow maps as motion information and train additional networks for motion compensation or output frame refinement. Niklaus et al. [18] utilize the context information extracted from ResNet-18 [10] along with the optical flows and refine the warped frames using their own neural network based on GridNet [9]. Using pre-trained optical flow has a problem that the flow maps consist of the vectors starting from the input frames. However, the vectors starting from the output frame are necessary to clearly warp the frames. To deal with this problem, Bao et al. [1] use the depth maps obtained from the hourglass architecture-based mono depth estimation network [3] to invert the optical flow maps clearly. Niklaus et al. [19] propose to combine all pixel values that are projected into the same locations using SoftMax function.

There are some researches to expand the domain of videos to be interpolated. Liu et al. [32] propose the quadratic video interpolation approach which utilize four frames to cover not only linear motions, but also quadratic motions. However, they still cannot deal with discontinuous motions. Lastly, Siyao et al. [26] propose the network that can interpolate cartoon videos. However they focus on the characteristics of cartoon images, not the motion of the videos. In this paper, we expand the video interpolation task to cover not only the natural motions, but also discontinuous transitions between the frames.

3. Proposed Approach

Given consecutive video frames $I_0$, $I_1$, $I_2$, and $I_3$, our study focuses on finding the intermediate frame $I_{out}$ between $I_1$ and $I_2$ by appropriately designing the frame interpolation network $F$. All the information required to produce $I_{out}$ can be obtained from $I_1$ and $I_2$. Our model obtained four frames as an input owing to estimate discontinuous motion area and the frames that model actually use to produce are $I_1$ and $I_2$. Thereafter, we introduced the proposed network based on the AdaCoF [15].

3.1. Revisit Previous model

In AdaCoF [15], they regarded the relationship as a warping operation $T$ (Adaptive Collaboration of Flows) from $I_1$ and $I_2$ to $I_{out}$. We can obtain $I_{out}$ as a combination of $T(I_1; \Theta_f)$ and $T(I_2; \Theta_h)$ for the weights and offset vectors for forward and backward warping $\Theta_f$ and $\Theta_h$. The input and output image sizes were $M \times N$, and the occlusion map was $V \in [0, 1]^{M \times N}$; the overall equation is as follows:

$$\Theta_f, \Theta_h, V = F_{AdaCoF}(I_1, I_2)$$

$$I_{out} = V \odot T(I_1; \Theta_f) + (J - V) \odot T(I_2; \Theta_h),$$

where $\odot$ denotes a pixel-wise multiplication, and $J$ represents an $M \times N$ matrix of ones. For the target pixel $(i, j)$, $V(i, j) = 1$ implies that the pixel is visible only in $I_1$, and $V(i, j) = 0$ implies that it is visible only in $I_2$.

However, we confirmed by directly applying previous models to test datasets that the distortion of the discontinuous motion area is large (Section 4.4). This phenomenon is attributed to the following two reasons: 1) Previous approaches have never seen the videos with discontinuous motion in the training stage. 2) Only two frames are not appropriate as an input of the network to collect information about where discontinuous motion area is.
3.2. Proposed Network

The first case that we attempted had four frames as input, which directly produced the result. In this case, it was hard for the network to get proper pixels from input frames to complete output because of using four frames as an input of the network. Therefore, we separated the mainstream that used only two frames and the sub-network that used all input frames to obtain a discontinuity map $D$. Hence, the equation of the overall network is as follows:

$$\Theta_f, \Theta_b, V, D = F_{General}(I_0, I_1, I_2, I_3).$$

(3)

**Discontinuity map.** Discontinuous motion area contained any objects with discontinuous movements or static movements, such as chatting, game user interface, or the shape of objects that change immediately, for example, numbers and letters. Let discontinuity map be the extracted map from the network where the discontinuous motion area of the input is. Thereafter, the discontinuity map can be used as a guide in which the network has to copy and paste from the previous frame $I_1$. We designed the discontinuity map $D \in [0, 1]^{M \times N}$ and modified the equation 2 as follows:

$$I_{cont} = V \odot T(I_1; \Theta_f) + (J - V) \odot T(I_2; \Theta_b)$$

(4)

$$I_{out} = D \odot I_1 + (J - D) \odot I_{cont}.$$  

(5)

For the target pixel $(i, j)$, $D(i, j) = 1$ implies that the pixel should be copied from $I_1$. The sub-network for discontinuity map extracted the discontinuous motion area from four input frames: the discontinuity map of one channel is presented as a result.

3.3. Figure-Text Mixing (FTM)

Previous studies have only applied flip augmentation with spatial and temporal axes. However, only this general augmentation is not sufficient to address various types of videos. We proposed a new data augmentation method for frame interpolation, called figure-text mixing (FTM), to handle general video frame interpolation. FTM consisted of two types of data augmentation methods: figure mixing (FM) and text mixing (TM). The discontinuity map, i.e.,
We added figures to the input frames to address the static objects in videos. The added figures had the same position and property on all input frames. We randomly applied the augmentation for the video sequence (see Figure 2). This technique can be the guideline where the discontinuous motion area is. This method also maintained the edge of an object from collapsing, even if the objects had continuous motion.

**Text Mixing.** There were many letters or sentences in videos, such as chatting and watermark. We put text on the video in the following four methods owing to the difference between the property of discontinuous and continuous motion areas: 1) the position of the text is static in the entire video, 2) where the text does not exist in the previous frame and appear in the future frame, 3) in the opposite of case 2), 4) the position of the text changes. We set the method of FTM in which the ground-truth frame and the discontinuity map follow the previous frame augmentation in text mixing. Therefore, we applied data augmentation, as given in Figure 2. The first row of Figure 2 is the case of 2), and the second row is the case of 4).

### 3.4. Network Architecture

We composed the proposed network based on [15]. Our network maintained the U-Net structure; we provided four frames as an input. After encoding the input frames, the eight sub-networks finally estimated the outputs \((W_{k,l}, \alpha_{k,l}, \beta_{k,l} \text{ for each frame and } V, D)\). We also used sigmoid activation for \(D\) as same as \(V\) to satisfy \(D \in [0, 1]^{M \times N}\). The network separates feature map into for combine output and for a discontinuity map before the sub-network stage and uses only two frames combined \(I_{\text{out}}\). Hence, the input of the sub-network that extracted the discontinuity map had 128 channels and the input of the other sub-network had 64 channels. The extracted discontinuity map was used in the final summation. More detailed architectures of the proposed network are described in Figure 1.

### 3.5. Objective Functions

**Loss Function.** First, we reduced the difference between the model output \(I_{\text{out}}\) and ground truth \(I_{\text{gt}}\). We used \(\ell_1\) norm for the loss as follows:

\[
L_1 = \|I_{\text{out}} - I_{\text{gt}}\|_1.
\]

We use the Charbonnier Function \(\Phi(x) = (x^2 + \epsilon^2)^{1/2}\) for optimizing \(\ell_1\) distance, where \(\epsilon = 0.001\) by following Liu et al. [16].

**Discontinuity map Supervision.** Difference previous algorithms, the proposed network has another result discontinuity map \(D\). We obtained the augmentation map of ground-truth frame \(D_{\text{gt}}\) by applying FTM in the training stage. Therefore, we provided discontinuity map supervision to the network to learn the location of discontinuous motion areas. We used two types of losses to apply supervision. Between \(\ell_1\) distance and binary cross entropy loss are used as follows:

\[
L_{D1} = \|D - D_{\text{gt}}\|_1
\]

\[
L_{Db} = - \sum_p \log P(D = D_{\text{gt}}),
\]

where \(p\) is the pixel location in \(D\).
Perceptual Loss. We also compare our network with and without perceptual loss. Previous studies [7, 13, 37] have demonstrated that perceptual loss is effective in producing more natural outputs. We apply the perceptual loss $L_f$ to our network with the feature from conv4 of ImageNet pretrained VGG16 network to confirm effectiveness.

$$L_f = \| F(I_{out}) - F(I_{gt}) \|_2$$

(9)

Therefore, the total loss for training as follows:

$$L_{total} = \lambda_1 L_1 + \lambda_f L_f + \lambda_D L_D,$$

(10)

where $\lambda_1 = \lambda_D = 1$ and $L_D$ can be $L_{D1}$ or $L_{Db}$.

4. Experiments

4.1. Experimental Settings

Training Dataset. We used Vimeo90K [33] dataset for training; it consisted of seven pairs of video frames. The septuplet dataset consisted of 91701 sequences with a fixed resolution of $448 \times 256$. We selected four out of seven frames at equal time intervals as inputs and the middle of the sequence as ground-truth. First, we randomly cropped $256 \times 256$ patches from the sequence for training. We also applied augmentation to a sequence that flipped horizontally, vertically, and temporally. After that, we finally added proposed data augmentation (FTM) to a video sequence based on the uniform probability.

Evaluation Setting. We composed a test dataset that had at least five frames for the evaluation of our network and previous method because the proposed network considered four frames at uniform time intervals as the input. We chose the video sequence, which had several frames, such as UCF101 [27] and the DAVIS test-dev dataset [24]. We also conducted a Game-graphic dataset for evaluating the discontinuous motion area, which consisted of game user interface videos. These test datasets contained approximately 70 frames for one sequence. However, the videos had various discontinuous motions and constant scenes in the Game-graphic dataset. Therefore, we only extracted five frames of the videos in the Game-graphic dataset containing discontinuous motion. We evaluated each algorithm by measuring PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity) [30] and LPIPS (Learned Perceptual Image Patch Similarity) [35] for all test datasets. The red numbers in all the tables in this paper represent the best performance, and the blue numbers represent the second-best performance.

4.2. Ablation Study

We analyzed the contributions of each component in terms of four keywords: data augmentation, number of input frames, discontinuity map supervision, perceptual loss.

Data Augmentation. We re-trained previous networks by applying FTM while fixing the other parameters to verify that FTM leads to obtain high performance for both continuous and discontinuous motion areas. We selected three networks for comparison: DVF [16], Sepconv [21] and AdaCoF [15]. The networks without symbol + are provided pre-trained networks by each author, with symbol + means which we train those networks by applying FTM in Table 2. As shown in Table 2, applying FTM for training leads to
improving the performance in both continuous and discontinuous motion areas. Especially, we confirmed the large gap for the Game-graphic dataset, i.e., discontinuous motion area could be handled by applying only proposed data augmentation.

To demonstrate that using both figure mixing (FM) and text mixing (TM) take high performance on dataset for discontinuous motion area, we conducted experiments for various types of our models in Table 3. Table 3 shows approximately similar results between our models on UCF101 [27] and DAVIS datasets [24] for continuous motion area. However, the result on Game-graphic dataset is better for TM than FM, and rather to apply both Figure and Text Mixing. This implies that proposed data augmentation leads to better performance for discontinuous motion areas without decreasing the performance on continuous motion areas. Therefore, we verified that FTM is an efficient data augmentation method for handling general video frame interpolation through two experiments.

**Number of input frames.** We increased the number of the input frames for effectively extracting discontinuity maps as we mentioned in Section 3.4. To verify increasing the number of input frames, we compared our network with re-trained AdaCoF [15] and our network used two frames as an input. In Table 1, the number 2 in parentheses represents the number of the input frames, $L_{D1}$ and $L_{Db}$ denote the supervision for the discontinuity map. The result shows that increasing the frames outperforms in the general motion area (see Table 1). In the continuous motion area especially in DAVIS dataset, the results of Ours-FTM-$L_{D1}$ obtained the highest performance for the three datasets. Additionally, we verified the difference between Ours-FTM (2) and Ours-FTM in Figure 3. In Figure 3, we can confirm the importance of the number of input frames, which increased input frames extract easily where discontinuous motion area is.

### Table 3. Result of ablation study on various type of our models.

| Method       | UCF101   | DAVIS   | Game-graphic |
|--------------|----------|---------|--------------|
|              | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| FM TM $L_{D1}$ $L_{Db}$ |      |       |       |      |       |       |      |       |       |
| ✓ ✓ | 33.531 | 0.9471 | 0.0369 | 27.279 | 0.8437 | 0.1368 | 28.675 | 0.9186 | 0.0819 |
| ✓ ✓ ✓ | 33.525 | 0.9473 | 0.0366 | 27.290 | 0.8446 | 0.1370 | 28.866 | 0.9205 | 0.0797 |
| ✓ ✓ ✓ | 35.569 | 0.9477 | 0.0364 | 27.212 | 0.8437 | 0.1334 | 29.151 | 0.9220 | 0.0776 |
| ✓ ✓ ✓ | 33.459 | 0.9466 | 0.0365 | 26.993 | 0.8347 | 0.1334 | 29.095 | 0.9256 | 0.0735 |
| ✓ ✓ ✓ | 33.532 | 0.9470 | 0.0368 | 27.233 | 0.8439 | 0.1371 | 29.066 | 0.9229 | 0.0756 |
| ✓ ✓ ✓ | 33.511 | 0.9471 | 0.0370 | 27.310 | 0.8458 | 0.1351 | 29.587 | 0.9232 | 0.0767 |
| ✓ ✓ ✓ | 33.502 | 0.9469 | 0.0369 | 27.193 | 0.8428 | 0.1393 | 29.349 | 0.9256 | 0.0735 |
| ✓ ✓ ✓ | 33.557 | 0.9477 | 0.0368 | 27.240 | 0.8449 | 0.1335 | 29.729 | 0.9280 | 0.0720 |
| ✓ ✓ ✓ | 33.606 | 0.9479 | 0.0360 | 27.199 | 0.8448 | 0.1355 | 29.698 | 0.9272 | 0.0720 |
| ✓ ✓ ✓ | 33.386 | 0.9459 | 0.0368 | 26.910 | 0.8343 | 0.1357 | 29.685 | 0.9227 | 0.0761 |

**Figure 4.** The result of discontinuity map on chatting example

**Discontinuity map supervision.** We applied two types of losses to supervise discontinuous motion areas introduced in Section 3.5: $L_{D1}$ and $L_{Db}$. We conducted three types of models with various types of applied FTM to understand the workings depending on the type of loss: no additional loss, $L_{D1}$, and $L_{Db}$. Results in Table 3 show the difference between applying $L_{D1}$ and $L_{Db}$. As shown in Figure 4, we obtained a clear discontinuity map for input frame when using $L_{Db}$, no discontinuity map for applying $L_{D1}$, and mixing with something without using loss. This implies that $L_{D1}$ indicates our network to handle the discontinuity map by only the using occlusion map, and $L_{Db}$ makes our network separate discontinuity map from occlusion map. In the case of Figure 4 (b), our network was trained by dividing occlusion and discontinuity map by itself. In short, applying the $L_{Db}$ used to obtain the discontinuity map shows that the model focuses too much on estimating the discontinuity map, resulting in slightly lower results on all datasets in Table 3. Therefore, no additional loss or application of $L_{D1}$ makes better performance for test datasets of general motion area for our models.

**Perceptual loss.** We added perceptual loss $L_f$ with $\lambda_f = 0.01$ to verify the influence of perceptual quality. Table 4 shows the influence of perceptual quality. The perceptual loss is deeply involved in terms of LPIPS [35] for all test datasets. Figure 5 results show the difference between with
Comparative algorithms for PSNR and SSIM by a high margin. In the case of other models, Ours-algorithms achieve state-of-art performance on three datasets.

We evaluated PSNR, SSIM [30], and LPIPS [35] of previous studies (see Table 4). Table 4 shows that our network showed sharp and clear results for large motion. Compared to previous algorithms using FTM and network, our network outperforms the previous algorithms using FTM and network.

### 4.3. Quantitative Results

We evaluated our network with several previous algorithms, such as DVF [16], SuperSlomo [12], SepConv [21], BMBC [23], AdaCoF [15], and Softsplat [19]. We compared three types of networks. One was the basic version which applied $L_2$. The second was a basic version with two input frames, and the third was the version that applied both $L_f$ and $L_{D1}$. We conducted an experiment for three datasets having different properties, as in Section 4.1. We evaluated PSNR, SSIM [30], and LPIPS [35] of previous studies (see Table 4). Table 4 shows that our networks achieve state-of-art performance on three datasets, containing both continuous and discontinuous motion areas. Our baseline model outperforms the previous algorithms for PSNR and SSIM by a high margin. In the case of other models, Ours-$FTM-L_f$ achieves the highest LPIPS on three datasets, and Ours-$FTM-L_{D1}$ (2) also outperforms than compared algorithms in most cases.

Especially, the margin between Ours-$FTM-L_{D1}$ and the other algorithms was over 1 dB for PSNR on Game-graphic datasets. This indicates that our model can effectively cover discontinuous motion areas compared to previous algorithms using FTM and network. As shown in Table 4, we can confirm that the gap on the DAVIS dataset [24] is larger than that on the UCF101 dataset [27]. Our method performs better on a dataset with large motion.

### 4.4. Qualitative Results

We focused on applying video frame interpolation for various types of videos, i.e., general video frame interpolation. Therefore, we conducted various types of videos to demonstrate the effectiveness of our models. We evaluated qualitative results on two types of datasets: continuous and discontinuous motion areas.

#### Continuous motion area

We selected the DAVIS test-dev dataset [24] for continuous motion area to compare, which is generally used for previous studies. We compared with AdaCoF [15], DVF [16], and Softsplat [19], as shown in Figure 6. The other algorithms showed the afterimage effect for the watermark (tractor example). On the contrary, our network showed the clear results owing to applying FTM and discontinuity mapping. Figure 6 shows that these examples have large motion in continuous domain (horsejump-stick, tennis-vest, girl_dog). Comparative algorithms showed that the structure collapses at all or does not maintain details for videos with large motion. Compared to this, our network showed sharp and clear results for large motion by maintaining the structure of the objects. These imply that our model is robust for large motion.

#### Discontinuous motion area

We conducted conduct two types of comparison to demonstrate the effectiveness for discontinuous motion areas: the chatting example in Figure 3 and, the graphic example in Figure 7. Figure 3 illustrates a typical example of discontinuous motion because chatting has to move momentarily. Previous studies did not consider discontinuous motion; therefore, the sentence was distorted and could not be recognized. In contrast, our net-

### Table 4. Quantitative evaluation on three datasets. The number (2) means our model that only takes two input frames.

| Algorithms    | UCF101 | DAVIS | Game-graphic |
|---------------|--------|-------|--------------|
|               | PSNR   | SSIM  | LPIPS        | PSNR   | SSIM  | LPIPS        | PSNR   | SSIM  | LPIPS        |
| DVF [16]      | 29.559 | 0.9265| 0.0491       | 25.447 | 0.8079| 0.1438       | 27.276 | 0.8945| 0.0951       |
| SuperSlomo [12]| 28.471 | 0.9116| 0.0626       | 25.710 | 0.8225| 0.1278       | 27.736 | 0.8899| 0.1041       |
| SepConv-$L_1$ [21]| 33.001 | 0.9439| 0.0368       | 26.543 | 0.8387| 0.1270       | 28.396 | 0.9037| 0.0961       |
| SepConv-$L_f$ [21]| 32.760 | 0.9400| 0.0348       | 26.267 | 0.8261| 0.1145       | 28.050 | 0.8929| 0.0902       |
| BMBC [23]     | 28.700 | 0.9288| 0.0552       | 24.989 | 0.8218| 0.1411       | 27.454 | 0.9042| 0.0942       |
| AdaCoF [15]   | 33.247 | 0.9453| 0.0366       | 26.377 | 0.8287| 0.1329       | 27.984 | 0.8960| 0.0990       |
| Softsplat-$L_1$ [19]| 33.035 | 0.9440| 0.0366       | 26.534 | 0.8387| 0.1269       | 28.426 | 0.9039| 0.0958       |
| Softsplat-$L_f$ [19]| 32.799 | 0.9401| 0.0347       | 26.256 | 0.8260| 0.1145       | 28.078 | 0.8932| 0.0901       |
| Ours-$FTM-L_{D1}$ (2) | 33.289 | 0.9462| 0.0370       | 26.808 | 0.8380| 0.1388       | 28.989 | 0.9185| 0.0808       |
| Ours-$FTM-L_f$ | 33.295 | 0.9445| 0.0327       | 26.601 | 0.8253| 0.1131       | 29.344 | 0.9228| 0.0607       |
| Ours-$FTM-L_{D1}$ | 33.606 | 0.9479| 0.0360       | 27.199 | 0.8448| 0.1355       | 29.698 | 0.9272| 0.0720       |

Figure 5. Visual comparison of adding perceptual loss on DAVIS test-dev dataset and without perceptual loss. Perceptual loss leads to the detailed and high perceptual quality result, as shown in Figure 5. We verify based on the results in Table 4 and Figure 5 that the perceptual loss can produce a detailed image, which cannot be produced using only $L_1$ loss.
work shows better results having sharp and clear edge of the letters. Figure 7 is a graphic image that contains both continuous and discontinuous motion areas. We selected two examples of this dataset for comparison: chatting with continuous motion background, appearing the object. The first row is the chatting example when the chatting goes up. The results of previous studies go up smoothly as the letters are broken; however, our result shows the clear letters with the same position with a ground-truth image. The second row represents the frame before the character was created. While existing algorithms produce an afterimage because they result in the next frame and median, the proposed network shows the result of copying the area from the previous frame for sudden scene transitions. Therefore, we confirm the large gap between our method and previous studies on the discontinuous motion domain (see Figures 3 and 7).

5. Conclusion

In this paper, we present the novel network for general video frame interpolation to address both continuous and discontinuous motion areas. We also propose a new data augmentation, named figure-text mixing (FTM), to handle discontinuous motion areas and to overcome large motion. The evaluation shows the state-of-the-art results for all kinds of datasets and indicators. These results imply that our network performs well for the general motion area without using extra datasets. We also demonstrated that the proposed architecture and data augmentation are effective solutions to handle general motion in various domains in ablation studies.
References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3703–3712, 2019.

[2] Wenbo Bao, Xiaoyun Zhang, Li Chen, Lianghui Ding, and Zhiyong Gao. High-order model and dynamic filtering for frame rate up-conversion. IEEE Transactions on Image Processing, 27(8):3813–3826, 2018.

[3] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In Advances in neural information processing systems, pages 730–738, 2016.

[4] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In Advances in neural information processing systems, pages 658–666, 2016.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[6] Tianyu Ding, Luming Liang, Zhuhui Zhu, and Ilya Zharkov. Cdfi: Compression-driven network design for frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8001–8011, 2021.

[7] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In Advances in neural information processing systems, pages 658–666, 2016.

[8] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepestereo: Learning to predict new views from the world’s imagery. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5515–5524, 2016.

[9] Damien Fourrure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. arXiv preprint arXiv:1707.07958, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[11] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Floconv 2.0: Evolution of optical flow estimation with deep networks. In IEEE conference on computer vision and pattern recognition (CVPR), volume 2, page 6, 2017.

[12] Huazhu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slowmo: High quality estimation of multiple intermediate frames for video interpolation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, pages 694–711. Springer, 2016.

[14] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. ACM Transactions on Graphics (TOG), 35(6):1–10, 2016.

[15] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5316–5325, 2020.

[16] Ziwei Liu, Raymond A. Yeh, Xiaou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[17] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In European Conference on Computer Vision, pages 434–450. Springer, 2016.

[18] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[19] 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.

[20] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[21] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[22] Genady Paikin, Yotam Ater, Roy Shaul, and Evgeny Soloveichik. Efi-net: Video frame interpolation from fusion of events and frames. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1291–1301, 2021.

[23] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbe: Bilateral motion estimation with bilateral cost volume for video interpolation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV, pages 109–125. Springer, 2020.

[24] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. arXiv:1704.00675, 2017.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. Unet: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
[26] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6587–6595, 2021. 1, 2

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012. 1, 5, 6, 7

[28] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 2

[29] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In European conference on computer vision, pages 402–419. Springer, 2020. 2

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004. 5, 7

[31] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 416–431, 2018. 1

[32] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. Advances in Neural Information Processing Systems, 32:1647–1656, 2019. 2

[33] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. International Journal of Computer Vision, 127(8):1106–1125, 2019. 5

[34] Zhiyang Yu, Yu Zhang, Deyuan Liu, Dongqing Zou, Xijun Chen, Yebin Liu, and Jimmy S. Ren. Training weakly supervised video frame interpolation with events. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 14589–14598, October 2021. 1

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595, 2018. 5, 6, 7

[36] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In European conference on computer vision, pages 286–301. Springer, 2016. 1

[37] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In European Conference on Computer Vision, pages 597–613. Springer, 2016. 5