Deep Animation Video Interpolation in the Wild Supplementary File

Li Siyao[1*] Shiyu Zhao[1,2*] Weijiang Yu[3] Wenxiu Sun[1,4]
Dimitris Metaxas[2] Chen Change Loy[5] Ziwei Liu[5]
[1]SenseTime Research and Tetras.AI [2]Rutgers University [3]Sun Yat-sen University
[4]Shanghai AI Laboratory [5]S-Lab, Nanyang Technological University
lisiyao1@sensetime.com sz553@rutgers.edu weijiangyu8@gmail.com
sunwenxiu@sensetime.com dnm@cs.rutgers.edu {ccloy, ziwei.liu}@ntu.edu.sg

Abstract

In this supplementary file, we provide some details of the dataset **ATD-12K** *and illustrate network architectures of the proposed method* **AnimeInterp**. *Moreover, with the motion tags of ATD-12K, we benchmark the performance of various methods on different motion categories. A demo video, which contains several clips of interpolation results generated by* **AnimeInterp** *and other state-of-the-art methods, is attached to further demonstrate the effectiveness of our method on animation video interpolation.*

1. Details of ATD-12K

Data sources (Section 3.1 in the main paper). Triplets of ATD-12K are collected from 30 cartoon movies made by diversified producers. The training set is collected from 22 movies, and the test set is collected from the rest 8. The detailed movie names are listed as follows:

- **Movies for the training set:** *Brother Bear; Treasure Planet; Atlantis: The Lost Empire; Lilo & Stitch; The Emperor's New Groove; Tarzan; Mulan; The Hunchback of Notre Dame; Pocahontas; Beauty and the Beast; Aladdin; The Little Mermaid; The Adventures of Ichabod and Mr. Toad; Princess Mononoke; Howl's Moving Castle; The Wind Rises; Your Name; Time Traveller: The Girl Who Leapt Through Time; The Disappearance of Haruhi Suzumiya; Sword Art Online The Movie Ordinal Scale; Cinderella; A Silent Voice.*
- **Movies for the test set:** *The Lion King; The Jungle Book; Peter Pan; Alice in Wonderland; Spirited Away; Children who Chase Lost Voices; The Boy and The Beast; K-ON!*

Difficulty level (Section 3.2). We divided triplets of the ATD-2K into three levels, *i.e.*, "Easy", "Medium", and "Hard", which indicate the difficulty to generate the middle image $I_{1/2}$ of a triplet with I_0 and I_1. Specifically, the difficulty levels are labeled based on the average magnitude of motion and the the ratio of the occlusion area in the input frame. First, we estimate the optical flow $f_{0\to1}$ of I_0 and I_1. Then, we compute the mean $\bar{f}_{0\to1}$ and standard deviation $\sigma_{f_{0\to1}}$ of the magnitude of $f_{0\to1}$. Next, we use $f_{0\to1}$ to splat a tensor (denoted as $\mathbf{1}$), which has the same size of I_0 and is filled with 1. The splatting result is notated as $\mathbf{1}_{0\to1}$. In $1_{0\to1}$, pixels with values smaller than 0.05 is regarded to be occluded, and the occlusion area rate $O_{f_{0\to1}}$ is computed as the ratio of the number of the occluded pixels and the area of the whole image. Based on calculated $\bar{f}_{0\to1}$, $\sigma_{f_{0\to1}}$, and $O_{f_{0\to1}}$, we define the difficulty levels according to the rules shown in Table 1. Note that we take not only the mean magnitude but also the inner variation of motion into judgement to reflect the complexity more objectively.

Table 1: **Definition of difficulty levels.**

	Range of $O_{f_{0\to1}}$		
	$[0, 0.05)$	$[0.05, 0.2)$	$[0.2, \infty)$
$\bar{f}_{0\to1} > 10, \sigma_{f_{0\to1}} > 10$	Middle	Hard	Hard
$\bar{f}_{0\to1} > 10, \sigma_{f_{0\to1}} \leq 10$	Easy	Middle	Hard
$\bar{f}_{0\to1} \leq 10$	Easy	Easy	Middle

2. Details of AnimeInterp

Contour Extraction and Color Piece Segmentation (Section 4.1). Unlike edges of real image, the contours of a cartoon are explicitly drawn with non-negligible width. If one use a regular edge detection algorithms (*e.g.* Canny detector) on anime image, two seperative lines will be produced for both sides of the contour, which will yield wrong segmentation in the next step. Instead, we adopt convolution on the image with a 5×5 Laplacian kernel to filter out a consistent

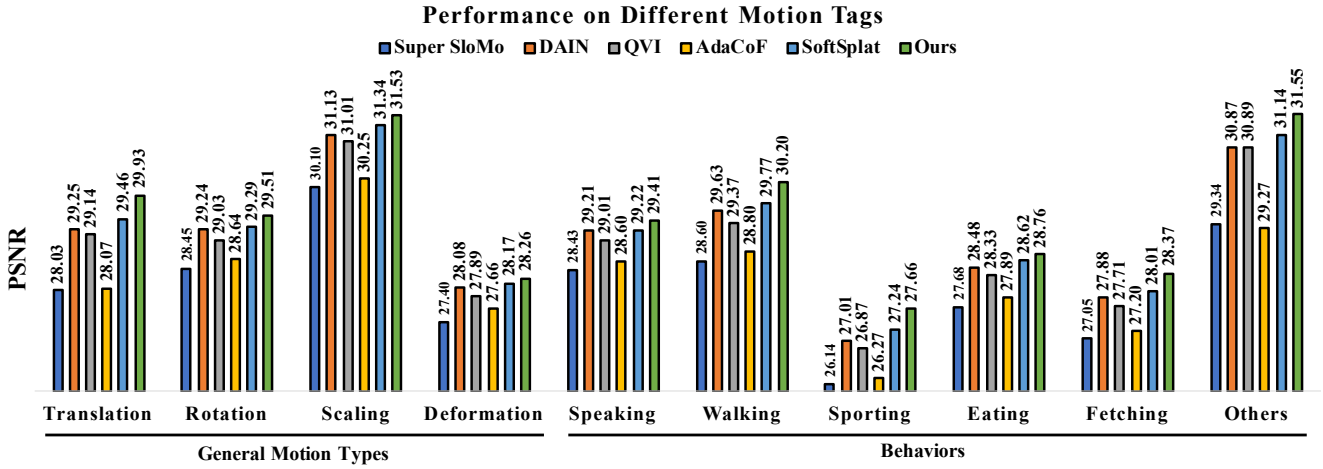

Figure 1: **Evaluations on different motion tags of ATD-12K.** AnimeInterp achieves the leading performance on all tags.

boundary. We then clip out the negative values of the result and refine it via the double-thresholding algorithm [1] to get the final contour. As Laplacian operator is very sensitive to noise, bilateral filtering [3] is performed in advance to avoid the potential artifacts of input images. Based on the extracted contour, we segment input frame into a bunch of color pieces. To realize this, we adopt the "trapped-ball" filling algorithm proposed in [4], where each area enclosed by the contour is split as a single piece.

Network Structures of RFR Module (Section 4.2). In RFR, a feature extraction network (FeatureNet) transforms the input frames I_0 and I_1 into deep features $\mathcal{F}_0$ and $\mathcal{F}_1$, respectively, and a three-layer CNN is adopted to predict confidence maps to suppress the unreliable values in the coarse flows $f_{0\to1}$ and $f_{1\to0}$. The detailed structures of the FeatureNet and the three-layer CNN are listed in Table 2.

Then, a series of residues $\{\Delta f_{0\to1}^{(t)}\}$ are learnt via a convolutional GRU [2]:

$$\begin{aligned} x^{(t)} &= [f_{0\to1}^{(t)}, \mathcal{F}_0, corr(\mathcal{F}_0, \mathcal{F}_{1\to0}^{(t)})], \\ z^{(t)} &= \sigma(\text{Conv}([h^{(t-1)}, x^{(t)}])), \\ r^{(t)} &= \sigma(\text{Conv}([h^{(t-1)}, x^{(t)}])), \\ \widetilde{h}^{(t)} &= \sigma(\text{Conv}([r^{(t)} \odot h^{(t-1)}, x^{(t)}])), \\ h^{(t)} &= (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot \widetilde{h}^{(t)}, \\ \Delta f_{0\to1}^{(t)} &= \text{Conv}(h^{(t)}), \end{aligned} \tag{1}$$

where Conv represents 3×3 convolutions, σ denotes the ReLU. For two $N \times H \times W$ features $\mathcal{F}_0$ and $\mathcal{F}_1$ (N, H, W are the channel numbers, the height and the width of the tensors), the $corr$ operation computes a normalized correlation

Table 2: **Architectures of "FeatureNet" and "3-layer CNN" in RFR module. (Section 4.2)**. The arguments in Conv represent the input channel number, the output channel number, the kernel size, and the convolution stride in turn. RB denotes Residual Block where the input is downsampled to the same size of the learned residue. σ represents ReLU.

	Layer name(s)
FeatureNet	Conv(3, 64, 7, 2), σ
	RB[Conv(64, 64, 3, 1), σ, Conv(64, 64, 3, 1), σ]
	RB[Conv(64, 64, 3, 1), σ, Conv(64, 64, 3, 1), σ]
	RB[Conv(64, 96, 3, 2), σ, Conv(96, 96, 3, 1), σ]
	RB[Conv(96, 96, 3, 1), σ, Conv(96, 96, 3, 1), σ]
	RB[Conv(96, 128, 3, 2), σ, Conv(128, 128, 3, 1), σ]
	RB[Conv(128, 128, 3, 1), σ, Conv(128, 128, 3, 1), σ]
	Conv(128, 256, 1, 1)
3-layer	Conv(6, 64, 5, 1), σ
	Conv(64, 32, 3, 1), σ
	Conv(32, 1, 3, 1), σ

c between those two tensors as

$$c(u \cdot \Delta + v, i, j) = \sum_{n=0}^{N-1} \mathcal{F}_0(n, i, j) \cdot \mathcal{F}_1(n, i+u, j+v)/N, \tag{2}$$

where Δ is the shift size and $u, v \in [-\frac{\Delta-1}{2}, +\frac{\Delta-1}{2}]$.

3. More Experimental Results

Performance on different motion tags (Section 5.3). In supplement to the experimental results of the main paper, the performance of various methods on different motion categories is shown in Figure 1. In general, the proposed AnimeInterp achieves the highest PSNR score on each motion

Figure 2: **Visual comparisons with state-of-the-art methods.** Our proposed method generates more complete objects in the interpolation results.

category. For specific motion categories, *e.g.*, "Walking", AnimeInterp can improve over 0.4dB than state-of-the-art methods. The influence of different motion categories on the performance of video interpolation can be further analyzed in the future studies.

More visual comparisons (Section 5.1). More visual comparisons between our proposed method and the state-of-the-art are shown in Figure 2. A video demo is also attached in the supplementary file to show the effectiveness of the proposed method.

References

[1] John Canny. A computational approach to edge detection. *IEEE TPAMI*, (6):679–698, 1986.

[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.

[3] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[4] Song-Hai Zhang, Tao Chen, Yi-Fei Zhang, Shi-Min Hu, and Ralph R. Martin. Vectorizing cartoon animations. *IEEE TVCG*, 15(4):618–629, 2009.