SUPPLEMENTARY MATERIAL FOR ENHANCED DEEP ANIMATION VIDEO INTERPOLATION

Wang Shen[⋆], *Cheng Ming*[⋆], *Wenbo Bao*[†], *Guangtao Zhai*[⋆], *Li Chen*[⋆], *Zhiyong Gao*[⋆]

[⋆] Shanghai Jiao Tong University [†] PointSpread Technology Inc.

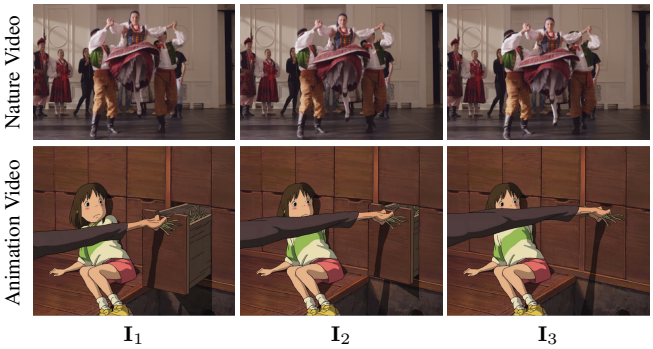

Fig. 1. Illustration of consecutive frames of nature video and animation video.

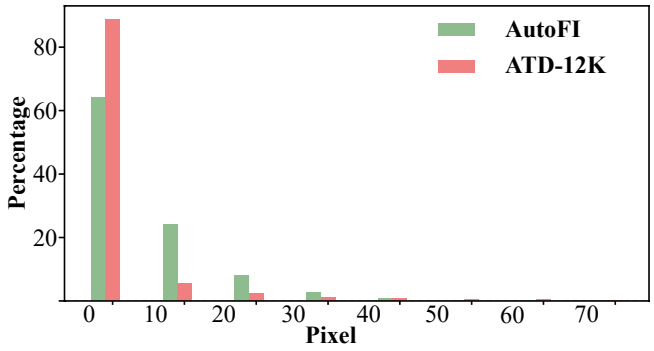

Fig. 2. Characteristic of dataset. Histogram of datasets AutoFI and ATD-12K [1].

1. AUTOFI TRAINING SET

Most pixels of a natural frame move linearly, except for a few that are nonlinear due to nonlinear motion, such as rotation. In the animation industry, animation frames are produced from hand drawing. Thus, it can not ensure linear motion between consecutive animation frames. As shown in Figure 1, the middle animation frame I_2 does not fit the linear assumption of traditional motion model at timestamp $t = 0.5$. Using those nonlinear animation data to train frame interpolation networks will mislead the synthesis procedure, resulting in blurry interpolated frames. Thus, it is unsuitable for generating a training set directly from cartoon videos.

We show triple samples of the training set by AutoFI in Figure 4. We then present the statistics of the constructed training set. We compare the motion statistics of the ATD-12K [1] and the proposed AutoFI training set. We estimate the optical flow between consecutive frames of the input pair of the two datasets. We compute the mean displacements of each optical flow. We show the histograms in Figure 2. The results show that the AutoFI has a higher percentage of large and diverse displacement ($>$ 10 pixels) than the ATD-12K dataset. Besides, we show more visual results of AutoFI in Figure 8.

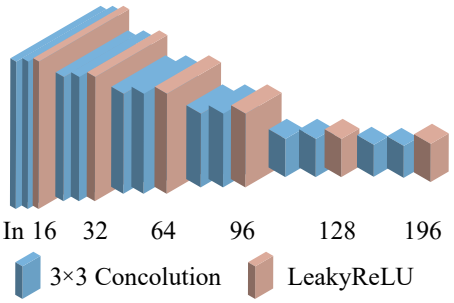

Fig. 3. Network architecture of feature extractor.

2. NETWORK DETAILS OF SKTFI

In this section, we introduce the network details of the SktFI module, as shown in Figure. 4 in the main manuscript. We use feature extractor for $\mathbf{I}_1$ and $\mathbf{I}_3$ in Figure 3, which consists 3×3 convolutional layer and LeakyReLU layer with negative slope equals to 0.1. We connect a pre-processing network for the middle feature extractor to address multiple frames input for the middle feature extractor. The architecture of the pre-processing network is shown in Table 1, which includes several convolutional layers and residual blocks. We use a U-net-like network as the frame synthesis module, which is adopted from previous work [2].

Table 1. Detailed configuration of pre-processing network.

	Input	Output	Kernel size	#Input Channels	#Output Channels	Stride	Activation	Output size
in	——	RGBs	——	——	10	——	——	$H \times W$
Conv2d	RGBs	ResB_2	7×7	10	64	1	LeakyReLU	$H \times W$
$\text{ResB}_n, n = [2, 4]$	ResB_n_in	ResB_n_conv1	3 × 3	10	64	1	LeakyReLU	$H \times W$
	ResB_n_conv1	ResB_n_out	3 × 3	64	64	1	LeakyReLU	$H \times W$
Conv2d	ResB_4_out	Out	3×3	64	3	1	——	$H \times W$

Fig. 4. Triplet samples of training set by AutoFI.

3. SKETCH GENERATION PIPELINE

The proposed SktFI utilizes one user-input sketch, a pair of keyframes, and an initial interpolated frame by frame interpolation algorithms to generate one output frame, as shown in Figure. 4 of the main manuscript. In this section, we first introduce the pipeline of sketch generation for training the SktFI. In order to conduct supervised learning, we create synthetic sketch $\mathbf{S}_2$ from the ground-truth intermediate frame $\mathbf{I}_2$. It is well acknowledged that deep neural networks tend to overfit the training samples and may generalize poorly to frames that slightly deviate from the training data. Therefore, it is significant to synthesize synthetic sketches as close to the hand-drawn one as possible. We first extract contour maps using the holistically-nested contour detection (HED) [3]. HED provides multi-level contour maps estimations. We choose the second level of its estimation because we empirically find that this level of output presents well visual resemblance to real drawn sketches, and it maintains high contour completeness, as shown in Figure 5.

We further remove short contours by performing morphological operations to remove small holes in the sketch. In addition, we fit splines for the contour maps using Potrace [4] and smooth the curvature by manipulating control points as suggested in [5], as shown in Figure 5. Such curve smoothing is essential for improving the generalization since it allows better tolerance to the potentially inaccurate sketch simplification and helps the network learn the synthesis based on rough contour locations.

4. ROBUSTNESS OF SKTFI

User Sketch versus Synthetic Sketch. During training, we generate synthetic sketch using the intermediate ground truth frame, as described in Section 3. During inference, the input sketch could be drawn by users. Since the sketches drawn by artists are rough and casual, developing a generic approach to process them directly is challenging. Therefore, a sketch simplification or cleanup is required as a pre-processing procedure, removing superfluous sketches and leaving a clean line drawing to characterize the motion. During inference, we adopt existing sketch simplification algorithms [6, 7] which are robust in producing reasonable sketch simplification for unseen styles by leveraging unsupervised data during train-

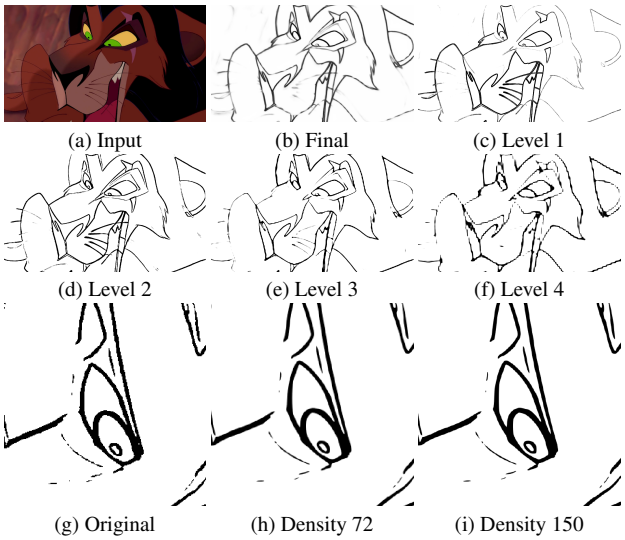

Fig. 5. Different level of HED outputs.

ing. We use simplified sketches as the network input in our work and focus more on video synthesis. We will use the term sketches to refer to the simplified ones with a clean line drawing style unless otherwise specified.

As shown in Figure 6, $\mathbf{S}_2$ is the synthesized sketch using the method described in Section 3, $\mathbf{S}_2$ *-user* is user-provided sketch using sketch simplification algorithms [6, 7] with pencil mode, $\mathbf{S}_2$ *-simp* is simplified sketch using sketch simplification algorithms [6, 7] with GAN mode. We find the results of SktFI using $\mathbf{S}_2$ and SktFI using $\mathbf{S}_2$ *-simp* show similar visual presentations in Figure 6. Both of them restores clear boundaries and vivid details of cartoon frames.

Besides, we show more visual results of SktFI in Figure 8.

5. REFERENCES

[1] 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*, 2021, pp. 6587–6595.

[2] Avinash Paliwal and Nima Khademi Kalantari, "Deep slow motion video reconstruction with hybrid imaging system," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1557–1569, 2020.

[3] Saining Xie and Zhuowen Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[4] potrace, "Potrace," 2021.

[5] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker, "Faceshop: Deep sketch-based face image editing," *ACM Transactions on Graphics*, vol. 37, pp. 99, 2018.

[6] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa, "Learning to simplify: Fully convolutional networks for rough sketch cleanup," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, 2016.

[7] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa, "Mastering sketching: Adversarial augmentation for structured prediction," *ACM TOG*, vol. 37, no. 1, pp. 1–13, 2018.

[8] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim, "Bmbc: 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 16*. Springer, 2020, pp. 109–125.

[9] 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*, 2020, pp. 5316–5325.

[10] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712.

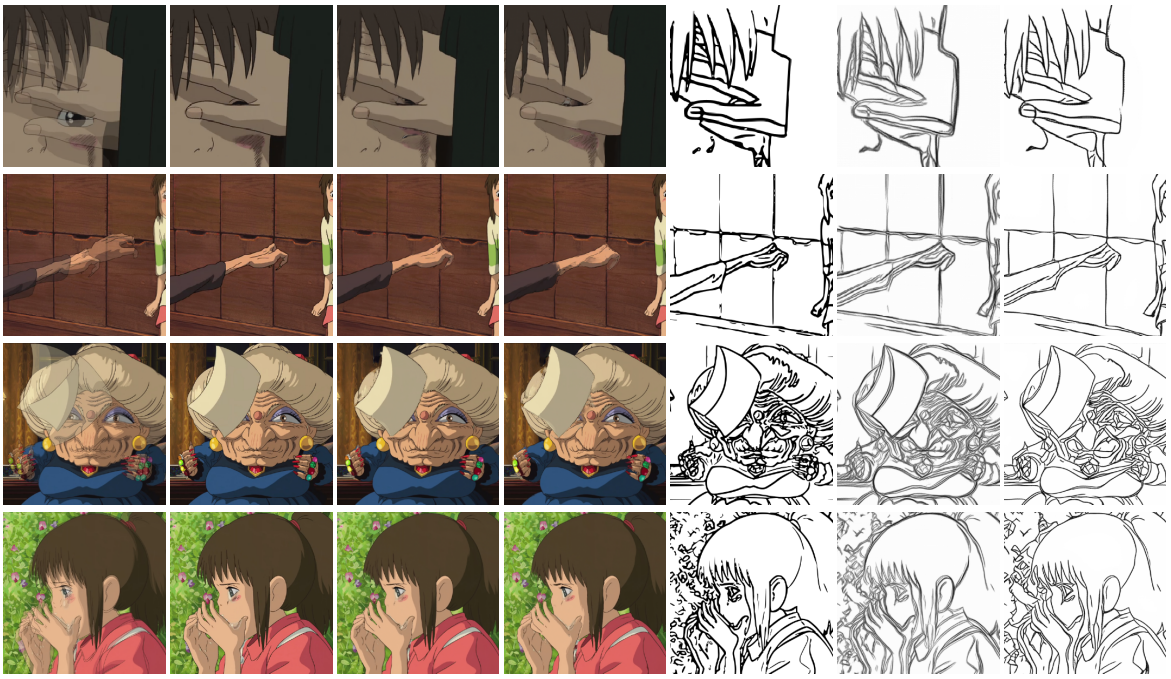

Fig. 6. Analysis of SktFI synthesis.

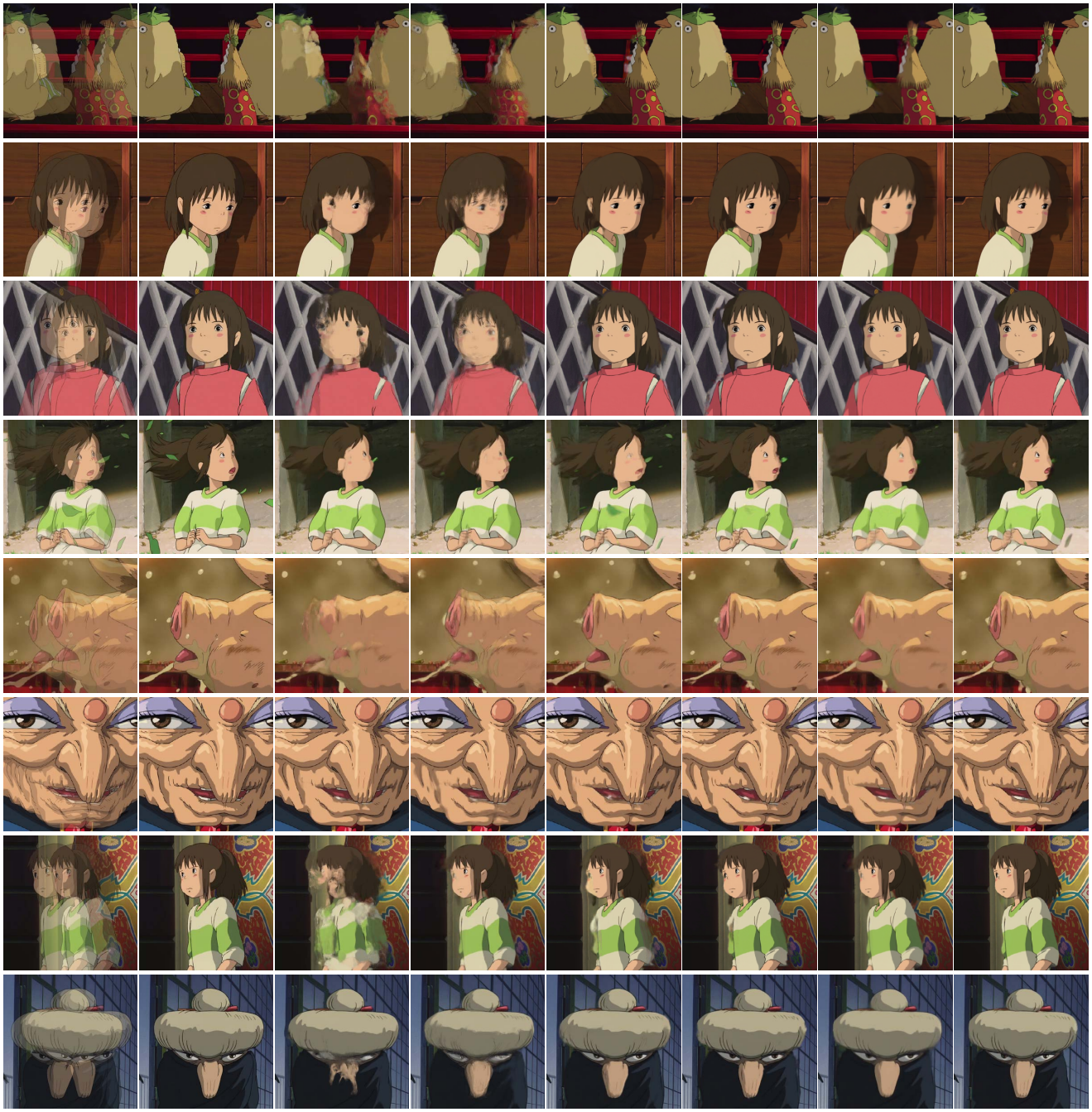

Fig. 7. Visual comparisons of AutoFI. The AutoFI synthesizes sharper and higher-quality interpolated frames.

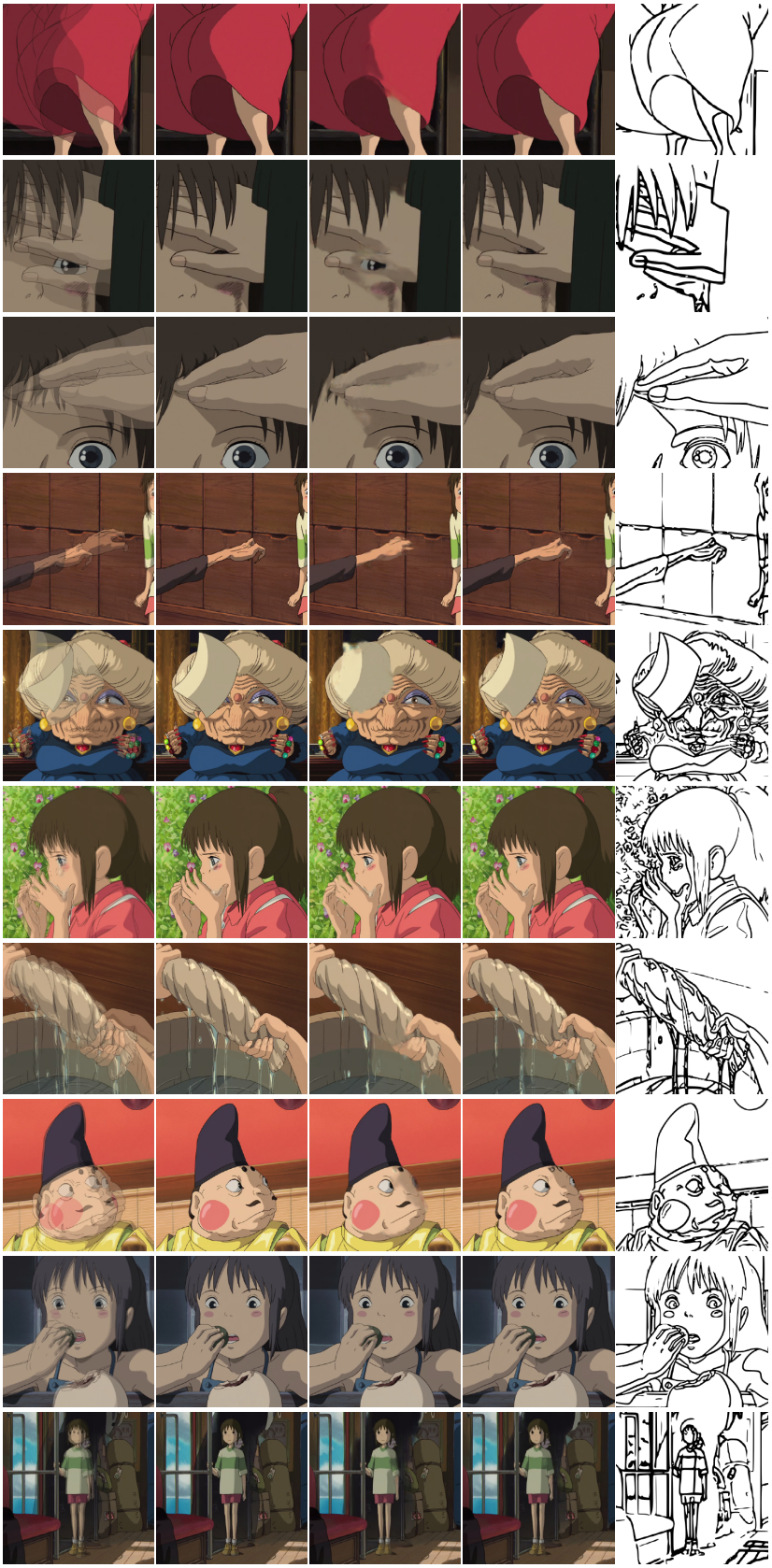

Fig. 8. Visual comparisons of AutoFI. The AutoFI synthesizes sharper and higher-quality interpolated frames.