Improving the Perceptual Quality of 2D Animation Interpolation, Supplementary

Shuhong Chen[1] and Matthias Zwicker[1]

University of Maryland, College Park, MD 20742, USA
{`shuhong,zwicker`}`@cs.umd.edu`

Abstract. In this supplementary file, we provide our user study responses, video results, additional experiment breakdowns, a discussion on tolerance to displacement, a discussion of computational cost, and a full list of data sources. Please unzip the supplementary before viewing any of the files. We release the full codebase at this github link[1].

Keywords: animation, video frame interpolation

1 User Study Results

Samples from our user study, as well as all user responses, can be found in `./user_study/index.html`. Please open this in a browser supporting WEBP animations. For each of the K=323 animation triplet pairs reviewed by our N=5 participants, we specify which generation was ours vs. AnimeInterp [8], provide the users' responses to each queried criteria, and additionally show comparative metrics (with an asterisk next to the better metric).

Due to file size limitations, we only store the first several triplets as shown to the participants, though we provide responses and statistics for all 323 triplets. The full user study can be found at the github linked in the abstract.

2 Slow-motion Video Examples

In the `./slomo_video/` folder, we provide several full-length cuts to compare our system with AnimeInterp [8] in context. Each video is from a held-out show that played no role in model training (see Tab. 5). The interpolation was set to 3x the native 24fps, played back at 12fps (effectively 6x slowdown); we compare the 24fps input, our best model, and the original AnimeInterp. As both compared models are based on forward splatting, it is trivial to interpolate at arbitrary rates simply by changing the interval parameter.

We compress with H264 to fit within the supplementary size limit, but use constqp=18 which does not introduce any noticeable compression artifacts. It is recommended to view the clips on a player that can loop and change playback speed (e.g. VLC).

---

[1] https://github.com/ShuhongChen/eisai-anime-interpolator

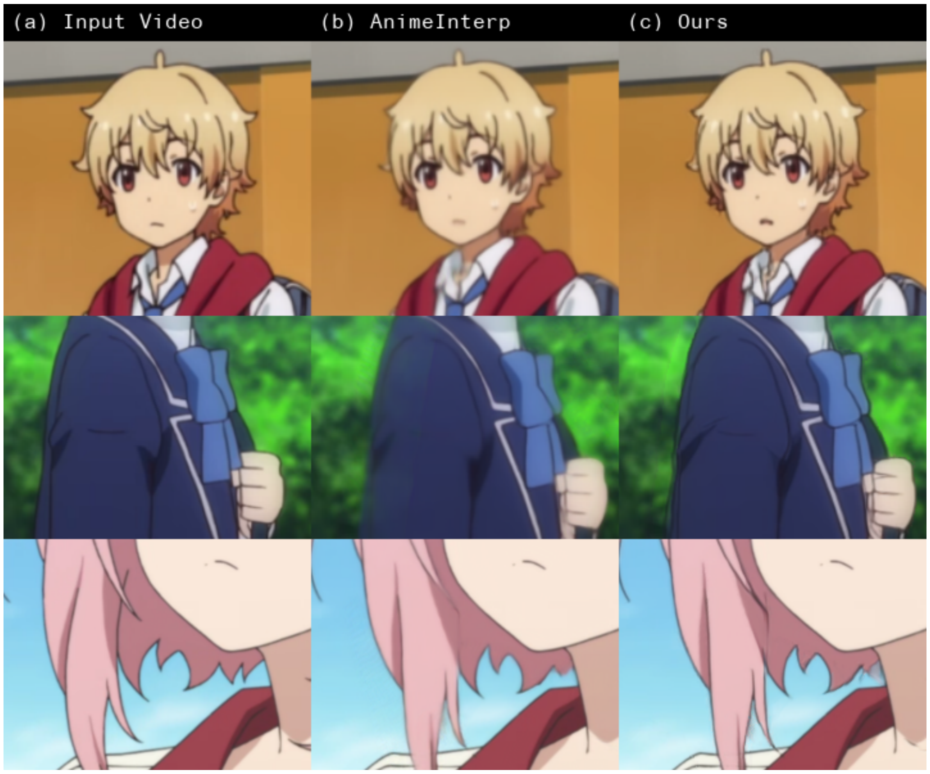

Fig. 7. Video results. These are screenshots taken from our full-cut video examples, please play the mp4 videos to see them in motion. AnimeInterp [8] blurs lines and causes ghosting artifacts, which cause noticeable flickering and line inconsistencies when played across time.

Importantly, we remove duplicate frames using our trained regression model. For long periods without movement, we continue interpolating at 5/24-ths of a second before the next movement, under the assumption that animators would not draw continuous movements at wider intervals. Had the duplicates not been removed, processed slo-mos would exhibit stop-and-go behaviors within a supposedly fluid motion. While vanilla AnimeInterp did not have deduplication[2], we process AnimeInterp outputs here with deduplication for fairer comparison.

We may observe similar trends across all examples. As expected, the original AnimeInterp introduces blurs that often undesirably erase lines from moving objects (see `konobi_s01e007__000282` and hair in `sakura_quest_s01e003__000082`). AnimeInterp's lack of an inpainting mechanism causes noticeable ghosting artifacts, causing flickers on the edges of moving objects (see edges of the blazer in

---

[2] AnimeInterp's demo: https://youtu.be/2bbujT-ZXr8?t=184, change playback speed to 0.25 to observe "stop-and-go" artifacts

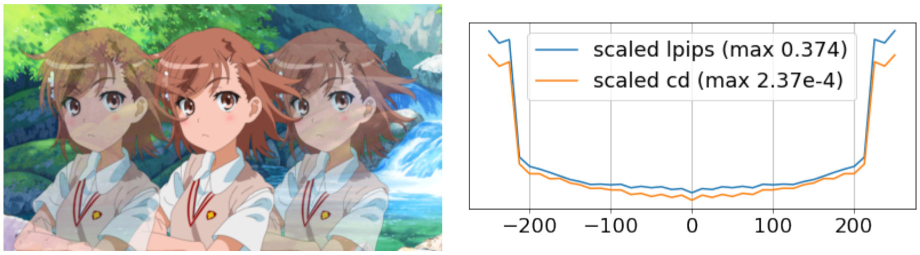

Fig. 8. Synthetic example showing pixel displacement (x-axis) vs. performance. Left: visualization of inputs at ±250px overlaid with center target. Image width: 960px. Artists: hariken, k.k.[3]

`konobi_s01e012__000339`). We point out several of these areas in Fig. 7. Though our model is still imperfect, we are demonstrably more robust to these pitfalls.

3 Tolerance to Displacement

We briefly investigate the effects of displacement magnitude on interpolation performance. Tolerance to pixel displacement is heavily dependent on the flow estimator, and it is difficult to disentangle performance with respect to displacement magnitude. We provide a synthetic example in Fig. 8 that demonstrates reasonable performance up to roughly ±220px across a 960px-width image, which is considered very large even in the animation domain.

4 Additional Experiment Breakdown

In Tab. 4, we provide additional breakdown of our experimental results by AnimeInterp annotations [8]. RoI metrics were calculated by cropping to the annotation and resizing the maximum dimension to 540px. We see that our full model still has the best LPIPS in all cases. However, similarly to the results from the main paper, chamfer distance sometimes behaves out-of-line with LPIPS. In the hard category, our model is significantly better on LPIPS than other models, but second place on CD; this is coupled with the same model (AnimeInterp ft.) also tying our chamfer distance in the RoI category.

We also briefly discuss PSNR/SSIM training. Our proposed methods were designed to improve perceptual quality, which we have shown does not always correlate with lower PSNR/SSIM values. Modules like SSL infilling and DTM preserve details/lines, but are not necessarily helpful when the loss encourages blurring. Nonetheless for sake of comparison, we report that training our system solely on L_1 loss (with the same hyperparameters as our best LPIPS/CD model) achieves 29.56/95.7 PSNR/SSIM. This performance is comparable to that of

---

[3] hariken: https://danbooru.donmai.us/posts/5378938
k.k.: https://danbooru.donmai.us/posts/789765

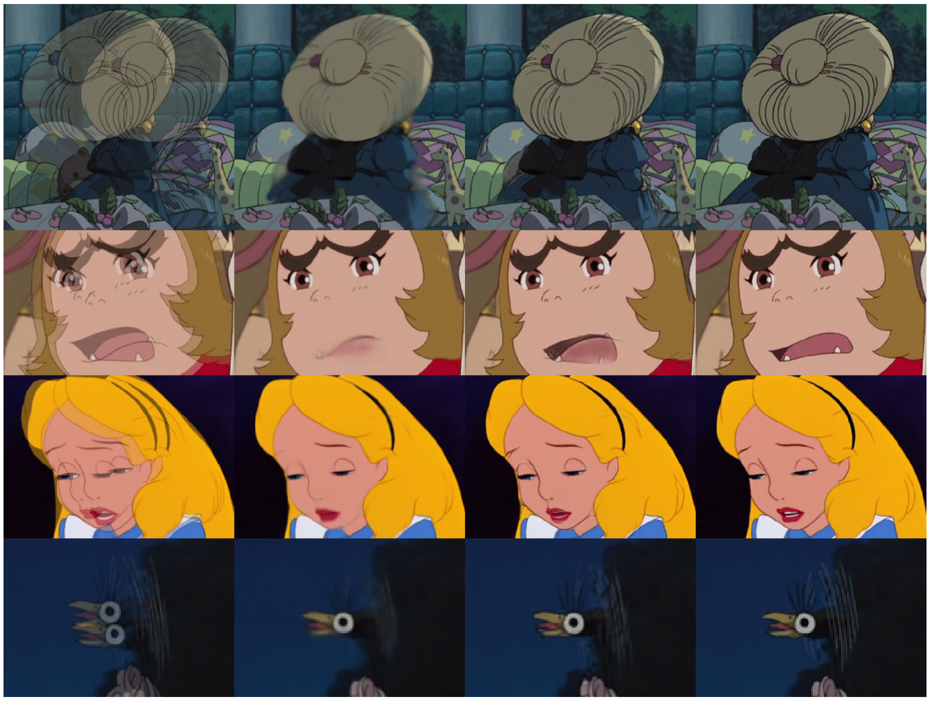

Fig. 9. More results, which were shown in the rebuttal. From left: inputs, AnimeInterp, ours, target. We preserve lines instead of blurring, and are better at retaining semantically meaningful shapes.

AnimeInterp 29.68/95.8, and still exceeds Softsplat [4] 29.34/95.7 (the runner-up model reported in AnimeInterp Tab. 1 [8]).

5 Computational Cost & SGM

While AnimeInterp's Segment-Guided Matching (SGM) [8] module improves performance in general, it is very computationally expensive. In our experiments, calculating a single bidirectional SGM flow took on average 118.4s, and thus must be precomputed for any training or inference to complete in reasonable time. This also severely limits the application of systems built on SGM, as a 1-second animation at 12fps would take around 24 minutes to process at inference. Though the given implementation could be better optimized, the core algorithm runs in quadratic time with respect to the typically high number of segmented regions in the input images, and additionally requires the computation of VGG [7] features.

As the contributions of our main paper do not focus on improving flow or inference speed, we used the same flows as AnimeInterp for our experiments (computed with SGM); we provide a brief tradeoff comparison here for completeness.

We found that removing SGM shortened our inference time from 2min to 0.45s, an over 260x speedup; at the same time, LPIPS increases from 3.49e-2 to 3.53e-2, and CD increases from 4.35e-5 to 4.40e-5. As this performance decrease is not very large, and as our non-SGM model still outperforms AnimeInterp ft. (LPIPS 3.76e-2, CD 4.51e-5), one could consider removing SGM for time-sensitive inference applications.

6 Data Sources

We use the existing frame triplets provided by AnimeInterp's ATD12k dataset [8], available on their github[4] under the MIT License. In addition, we use frames sourced from the videos listed in Tab. 5. These sources were purchased as blu-ray discs, ripped using MakeMKV[5], and re-encoded with FFmpeg[6]. To maintain a reasonable space-quality tradeoff and ensure compatibility with later parts of the pipeline, we encode with hevc_nvenc, at 24fps, constqp 23, medium preset. We then use NVIDIA DALI[7] and Kornia [6] to filter duplicate frames based on a linear regression of L_2 LAB color difference trained on a small hand-labeled dataset, removing roughly 49.17% of frames; the coefficients will be released with the code. The remaining frames are interpreted as triplets, and filtered with our proposed RRLD metric at threshold 0.3 using FlowNet2 flows [3], resulting in 543.6k triplets. We then limit to one triplet per cut, detected by pretrained TransNet v2 [9], coming to 49.7k triplets. The 9k triplets randomly selected from these for training are provided in `./frame_indices.txt`. The full pipeline for extracting these data is available at the github linked in the abstract.

References

1. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019)
2. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Rife: Real-time intermediate flow estimation for video frame interpolation. arXiv preprint arXiv:2011.06294 (2020)
3. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
4. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5437–5446 (2020)
5. Park, J., Lee, C., Kim, C.S.: Asymmetric bilateral motion estimation for video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14539–14548 (2021)

---

[4] https://github.com/lisiyao21/AnimeInterp

[5] https://www.makemkv.com/

[6] https://www.ffmpeg.org/

[7] https://github.com/NVIDIA/DALI

6. Riba, E., Mishkin, D., Shi, J., Ponsa, D., Moreno-Noguer, F., Bradski, G.: A survey on kornia: an open source differentiable computer vision library for pytorch (2020)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Siyao, L., Zhao, S., Yu, W., Sun, W., Metaxas, D., Loy, C.C., Liu, Z.: Deep animation video interpolation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6587–6595 (2021)
9. Souček, T., Lokoč, J.: Transnet v2: An effective deep network architecture for fast shot transition detection. arXiv preprint arXiv:2008.04838 (2020)

	RoI		Easy		Med		Hard	
Model	LPIPS	CD	LPIPS	CD	LPIPS	CD	LPIPS	CD
DAIN [1]	8.53	18.0	2.19	2.29	4.13	4.55	8.85	10.32
DAIN ft.	7.70	*16.5*	*1.89*	*2.20*	3.62	4.19	7.87	9.31
RIFE [2]	8.36	19.0	2.21	2.41	3.96	4.81	8.14	10.56
RIFE ft.	7.93	17.6	2.29	3.01	3.82	4.88	7.43	9.38
ABME [5]	9.92	27.2	2.84	2.53	4.87	5.07	10.75	16.30
ABME ft.	7.42	16.9	2.11	2.21	3.56	4.15	7.88	9.79
AnimeInterp [8]	9.10	20.1	2.88	2.65	4.60	4.69	8.60	10.62
AnimeInterp ft.	*6.81*	14.8	1.99	2.33	*3.43*	*4.08*	*6.60*	8.07
Ours	6.46	14.8	1.86	2.14	3.20	3.91	6.23	*8.14*

Table 4. Additional breakdown of results by AnimeInterp annotations [8], an extension of Tab. 1 from the main paper. Models from prior work are fine-tuned on LPIPS for fairer comparison. Best values are underlined, and runner-ups are italicized. "CD" stands for chamfer distance; LPIPS and CD are multiplied by factors of 1e2 and 1e5 for readability, respectively.

MAL	Title	Year	Studio	Eps.	Min/ep.	Dur.
512	Majo no Takkyuubin	1989	Studio Ghibli	1	103	103
849	Suzumiya Haruhi no Yuuutsu **	2006	Kyoto Animation	14	24	336
4382	Suzumiya Haruhi no Yuuutsu (2009) **	(2009)	Kyoto Animation	14	24	336
4224	Toradora!	2008	J.C.Staff	25	24	600
11553	Toradora!: Bentou no Gokui	(2011)	J.C.Staff	1	24	24
8557	Shinryaku! Ika Musume	2010	Diomedéa	12	24	288
10378	Shinryaku!? Ika Musume	(2011)	Diomedéa	12	24	288
13267	Shinryaku!! Ika Musume	(2012)	Diomedéa	3	24	72
9989	Ano Hi Mita Hana no Namae wo Bokutachi wa Mada Shiranai. *	2011	A-1 Pictures	11	24	264
15809	Hataraku Maou-sama!	2013	White Fox	13	24	312
20541	Mikakunin de Shinkoukei	2014	Doga Kobo	12	24	288
25835	Shirobako	2014	P.A. Works	24	24	576
24765	Gakkougurashi! ***	2015	Lerche	12	24	288
31953	New Game!	2016	Doga Kobo	12	24	288
34914	New Game!!	(2017)	Doga Kobo	12	24	288
31952	Kono Bijutsubu ni wa Mondai ga Aru! *	2016	feel.	12	24	288
34494	Sakura Quest *	2017	P.A. Works	25	24	600
39071	Machikado Mazoku	2019	J.C.Staff	12	24	288
37890	Oshi ga Budoukan Ittekuretara Shinu	2020	8bit	12	24	288
	Totals			239		5815

Table 5. Video sources. We take from 14 franchises, some with sequels/OVAs premiering in parenthesized years. Duration is in minutes. More information can be found for each show at `https://myanimelist.net/anime/MAL_ID`. (*) Anohana, Konobi, and Sakura Quest were set aside for qualitative validation purposes only (for example the full-length cuts in this supplementary), and played no role in model training/testing. (**) The Haruhi episodes in `./frame_indices.txt` follow the blu-ray order, all labeled as "season zero"; it was too complicated to reshuffle to broadcast order. (***) Note that out-of-context frames may be enough to spoil Gakkougurashi; we very strongly recommend watching the first episode blind before seeing any other related content.