Computational In-betweening for Line Drawings in Animation

(Supplemetary)

Anonymous Author(s)

1 DATA AND PRE-PROCESSING

We generated 240 videos in Blender [Community 2018], each with 242 frames and rendered at 24 frames per second for a total of 58,080 frames. These videos depict an object moving in space, spinning, and scaling up and down. Half of these videos contain a cube, which is a simpler 3D form, while the other half shows the Blender primitive of a monkey's head (called Suzanne), a more complex form. We only rendered the forms' edges to best match our intended domain, and also ran the difference of Gaussians (DoG) algorithm on each image to binarize our data into foreground segmentation masks. From each video, we took successive triplets of frames as our training data. We did this at two granularities: 1 intermediate frame (e.g., network inputs are frames 000 and 002 of a video, and the network output tries to match frame 001) and 3 intermediate frames (e.g., network inputs are frames 000 and 004 of a video, and the network output tries to match frame 002). Finally, we cropped our training images to 512×512 squares, both to augment our data and to make training more efficient.

As for the James Baxter's animation sequence of line drawings, called a "pencil test" by traditional animators, we took the video from his YouTube channel [Baxter 2016] and broke the video down into 24 individual frames per second. We once again ran the DoG algorithm on the frames, divided them into triplets at the aforementioned granularities, and cropped the training images into 512×512 squares to prepare these images as data.

2 TRAINING DETAILS

Our network was trained end-to-end. Our in-between generator was a UNet, with an underlying architecture of a ResNet34 [Yakubovskiy 2020]. It was pre-trained on ImageNet segmentation, and was then fine-tuned on our data. We used the Adam optimizer to train both our generator and our discriminator [Kingma and Ba 2015]. For the generator, we used an initial learning rate of 10^{-4}, which changed for subsequent epochs with cosine annealing [Loshchilov and Hutter 2017]. We used learning rate 10^{-5} to train the discriminator. For both network optimizers, we set $\beta_1 = 0.5$, $\beta_2 = 0.999$. We also weighted our cycle-consistency-inspired loss term [Zhu et al. 2017], since its magnitude was overwhelming that of our other loss terms. Therefore, we multiplied the raw cycle-consistency loss by 0.1. This resulted in the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{GAN}} + 0.1\mathcal{L}_{\text{cycle}}$$

3 ADDITIONAL RESULTS

There are no good standard metrics to evaluate the generated in-between frames. Standard metrics used in the generative modeling literature are not well suited for line drawing domain, as they encourage generating blurry "smudges" to cover more area as opposed to crisp lines. One future work of our research is to explore metrics suitable for line drawings. In addition, we will explore user studies with traditional animators, who work with raster line drawings.

Table 1. Preliminary results on cubes (Blender).

Method	PSNR ↑	SSIM ↑
DAIN	26.89	0.98
RIFE	**39.02**	**0.99**
Ours	28.93	0.99

Table 2. Preliminary results on Suzanne (Blender).

Method	PSNR ↑	SSIM ↑
DAIN	22.88	0.96
RIFE	**34.22**	**0.99**
Ours	25.23	0.98

Table 3. Preliminary results on pencil test.

Method	PSNR ↑	SSIM ↑
DAIN	18.68	0.89
RIFE	**20.69**	**0.96**
Ours	17.29	0.93

Even with these drawbacks, we report the standard generative modeling metrics here in Tables 1, 2, and 3 for completeness. For each dataset, we show comparisons to two state-of-the-art video interpolation methods in peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), both metrics that evaluate the quality of generated images (though again, not necessarily good metrics for line drawings). The best-performing method for each dataset is bolded and the second-best method is shown in blue. Note that although our metrics are usually an improvement on DAIN [Bao et al. 2019], we are usually outperformed by RIFE [Huang et al. 2022]. However, our qualitative results often show a different picture.

For the Blender-generated datasets, our results look just as clean as those generated by RIFE, though our PSNR and SSIM scores are usually lower. Furthermore, we see Figure 1 that for the pencil test, RIFE often returns blurry images, especially at motion boundaries. Note in particular the clusters of purple and green pixels near the left character's legs and hands in the fifth and seventh rows of Figure 1, where there is the most motion. In contrast, our framework generates crisper lines in our in-betweens, though they are sometimes grainy. This shows that the metrics alone do not reflect all aspects of the qualitative results, as they are well-performing for images with smudges, but not those that are grainy. This disparity between our metrics and the qualitative results is especially apparent when considering that DAIN outperforms our framework on the pencil test dataset in PSNR, but all DAIN in-betweens have many visual artifacts and are generally noisy, as well as not properly deforming shapes to actually depict characters moving "in between" the start and end frames.

REFERENCES

W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang. 2019. Depth-Aware Video Frame Interpolation. In *CVPR*.

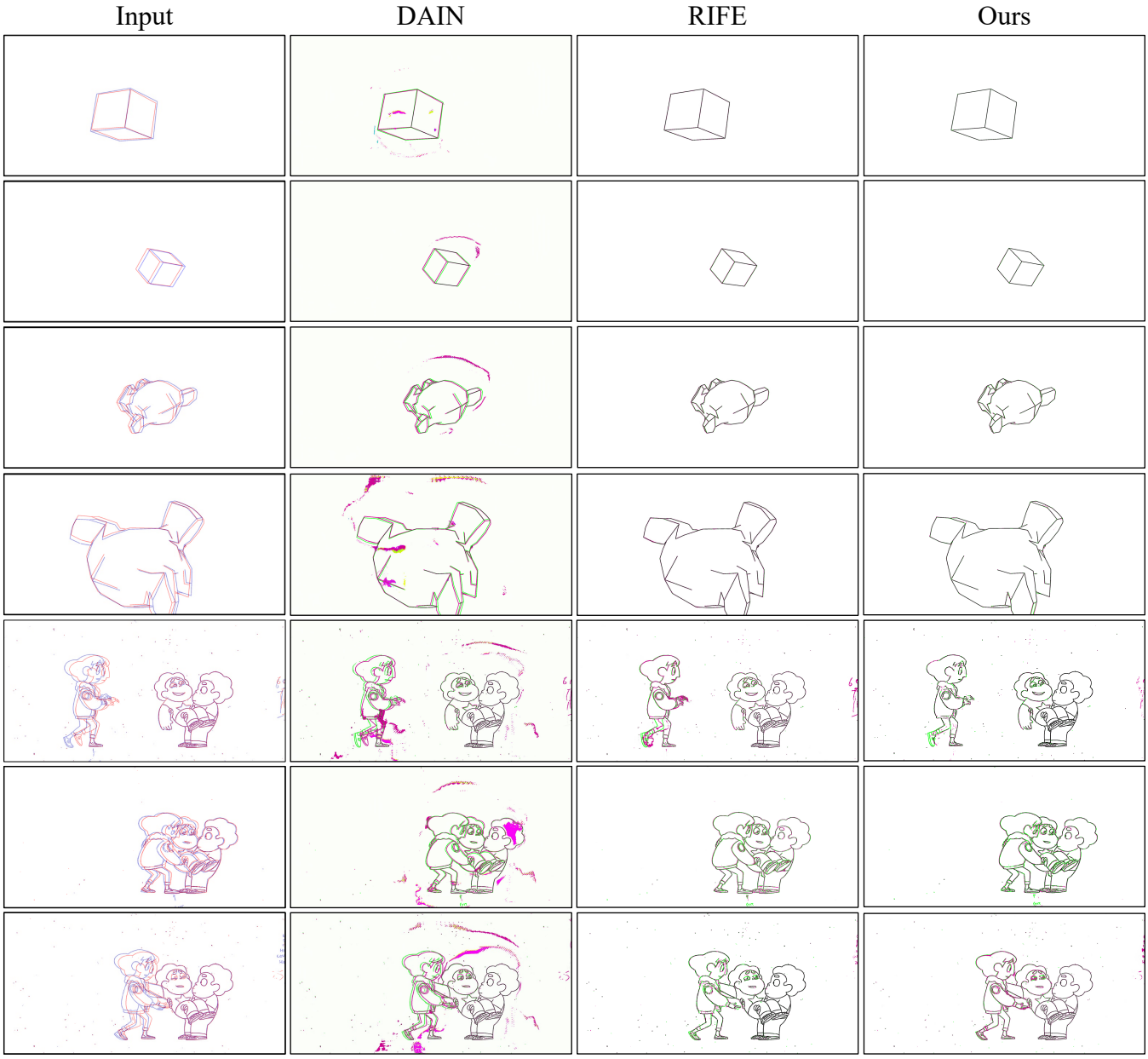

Figure 1. **Additional preliminary results**. The inputs to our framework are depicted in the leftmost column; the starting frame is shown in red, while the ending frame is overlaid in blue. The second column shows results on DAIN, the third shows results on RIFE, and the last shows our framework's results. Each row shows a different example from our various datasets: Blender-generated cubes, Blender-generated monkey heads (Suzanne), and James Baxter's pencil test. For each image in the three rightmost columns, **black** pixels depict true positives (correctly drawn in-between line), **purple** pixels are false positives (drawn line not present in ground truth), and **green** pixels are false negatives (lines missed by the generator). Original images ©James Baxter.

J. Baxter. 2016. James Baxter's YouTube Channel. https://tinyurl.com/fdjwj5mj

B. O. Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou. 2022. Real-Time Intermediate Flow Estimation for Video Frame Interpolation. In *ECCV*.

D. P. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun (Eds.). http://arxiv.org/abs/1412.6980

I. Loshchilov and F. Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Skq89Scxx

P. Yakubovskiy. 2020. Segmentation Models Pytorch. https://tinyurl.com/4stf98fb.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*.