Thin-Plate Spline-based Interpolation for Animation Line Inbetweening

Tianyi Zhu[1], **Wei Shang**[1], **Dongwei Ren**[2, ✉]

[1]Faculty of Computing, Harbin Institute of Technology
[2]Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University
ztianyi@stu.hit.edu.cn, csweishang@gmail.com, rendw@tju.edu.cn

Abstract

Animation line inbetweening is a crucial step in animation production aimed at enhancing animation fluidity by predicting intermediate line arts between two key frames. However, existing methods face challenges in effectively addressing sparse pixels and significant motion in line art key frames. In literature, Chamfer Distance (CD) is commonly adopted for evaluating inbetweening performance. Despite achieving favorable CD values, existing methods often generate interpolated frames with line disconnections, especially for scenarios involving large motion. To address this issue, we propose a simple yet effective interpolation method for animation line inbetweening, where thin-plate spline-based transformation is employed to estimate coarse motion more accurately by modeling the keypoint correspondence between two key frames, particularly for large motion scenarios. Building upon the coarse estimation, a motion refine module is employed to further enhance motion details before final frame interpolation using a simple UNet model. Furthermore, to more accurately assess the performance of animation line inbetweening, we refine the CD metric and introduce a new metric named Weighted Chamfer Distance, which demonstrates a higher consistency with visual perception quality. Additionally, we incorporate Earth Mover's Distance and conduct user study to provide a more comprehensive evaluation. Our method outperforms existing approaches by delivering high-quality interpolation results with enhanced fluidity.

Code — https://github.com/Tian-one/tps-inbetween

Introduction

2D animation plays a pivotal role in various domains, ranging from entertainment to education. With the advancement of generative models based on large language models, it may be feasible that 2D animation can be directly generated through video generation techniques (Guo et al. 2024; Xing et al. 2024). Despite involving conditions (Zhang et al. 2024; Mou et al. 2024; Zhang, Rao, and Agrawala 2023), generating animation videos using given key frames provided by animators remains non-trivial. Therefore, it is still crucial to assist animators in completing animation videos by providing several key frames. As shown in Fig. 1, the animation creation process involves several distinct steps: (i) Creating key animation frames by animators, (ii) Inbetweening, responsible for interpolating intermediate frames, and (iii) Colorizing and compositing the line arts. In previous studies, inbetweening and colorization have been sometimes used interchangeably. For example, in (Siyao et al. 2021; Chen and Zwicker 2022), inbetweening is addressed through video interpolation after colorizing and compositing key frames. Although automatic colorization techniques (Zhang et al. 2021; Li et al. 2022) offer certain advantages, they compromise the visual quality during the interpolation process, and present challenges for animators in terms of reprocessing post-colorized frames.

Therefore, this work devotes attention to animation line inbetweening, *i.e.*, frame interpolation on line arts of key frames. This task presents challenges due to the presence of substantial motion amplitudes and sparse pixel distribution, rendering existing video interpolation methods unsuitable (Kalluri et al. 2023; Huang et al. 2022). To overcome the large motion of line arts, in (Siyao et al. 2023), a graph-based approach is employed to vectorize and reposition vertices of the line art for generating intermediate graphs. While graph-based representation facilitates cleaner lines, it also introduces artifacts and often results in discontinuous lines that affect interpolation outcomes. For raster images, inaccuracies in vectorization methods can also impact the performance of this approach.

In this work, we propose a novel raster image-based interpolation pipeline for animation line inbetweening, as shown in Fig. 4. We first introduce the thin-plate spline (TPS) transformation (Bookstein 1989) to model coarse motion, by which the intermediate poses particularly for large motion can be more accurately obtained by using keypoints correspondence priors. Building upon the coarse estimation, a simple motion refine module is proposed, consisting of an optical flow network for refining coarse motion and a UNet model for synthesizing final frames. Additionally, we introduce an improved metric dubbed Weighted Chamfer Distance (WCD), which exhibits a closer alignment with human visual perception compared to the original CD metric (Chen and Zwicker 2022; Siyao et al. 2023).

Extensive experiments are conducted on benchmark datasets, and our method is compared against state-of-the-art techniques in video interpolation (Huang et al. 2022; Zhang et al. 2023; Li et al. 2023), animation interpolation (Chen

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

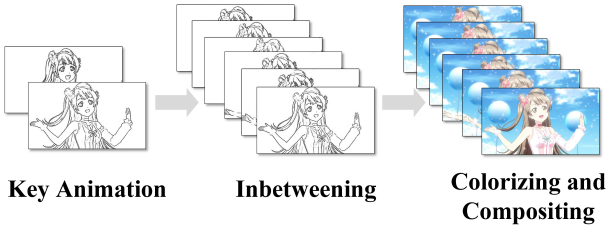


Figure 1: Common workflow of animation production, including creating key animation, inbetweening, colorizing and compositing.

and Zwicker 2022), and line inbetweening (Siyao et al. 2023). The evaluation metrics include CD and WCD scores, as well as introducing Earth Mover's Distance (EMD) and user study into consideration. Our approach outperforms existing methods by producing high-quality interpolation results with enhanced fluidity for all three interpolation gaps, *i.e.*, 1, 5, and 9. Our contributions can be summarized as:

- We propose a simple but effective framework for animation line inbetweening, which includes a coarse motion estimation module for handling large motions, and a motion refine module for refining motion details and generating final frames.
- The thin-plate spline-based interpolation module is proposed for better modeling the keypoint correspondence to handle large motion and line disconnections.
- We present an improved metric WCD and introduce the EMD metric for evaluating inbetweening effect. Extensive experiments demonstrate that our approach outperforms existing methods by generating high-quality interpolation results with enhanced fluidity.

Related Work

Video Frame Interpolation. In recent years, numerous methods have been proposed to address the challenge of video frame interpolation (Jiang et al. 2018; Shang et al. 2023; Sim, Oh, and Kim 2021; Reda et al. 2022; Kalluri et al. 2023; Yoo, Lee, and Jung 2023; Wu et al. 2023; Hu et al. 2024). Some methods attempted to extract features from concatenated input frames to learn kernels (Niklaus, Mai, and Liu 2017; Lee et al. 2020) or directly synthesize intermediate frames (Kalluri et al. 2023). The most recent state-of-the-art techniques (Li et al. 2023; Zhang et al. 2023; Wu et al. 2024) predominantly leverage optical flow. They predict the optical flow between the intermediate frames and the two input frames, utilizing warping to obtain the intermediate frames. Liu *et al.* (2024) introduce sparse global matching to refine optical flow for large motion. RIFE (Huang et al. 2022) adopts a privileged distillation scheme to directly approximate the intermediate flows for real-time frame interpolation. Zhang *et al.* (2023) proposed a hybrid CNN and Transformer (Vaswani et al. 2017) architecture to extract motion and appearance information. Li *et al.* (2023) introduced bidirectional correlation volumes to obtain fine-grained flow fields from updated coarse bilateral flows. While these methods have demonstrated superior performance in video frame interpolation within real-world scenarios, their application to the line art inbetweening often encounters challenges. The sparse pixel distribution in line art images and substantial inter-frame motion frequently result in undesired effects, such as blurring and artifacts.

Line Art Inbetweening. Previous works on line art inbetweening were often conducted under strict conditions (Yang 2017) or employed methods similar to video frame interpolation (Shen et al. 2023; Narita, Hirakawa, and Aizawa 2019). However, these approaches consistently faced challenges in handling large motion. Recently, Siyao *et al.* (2023) attempted to address line art inbetweening by converting line art into the graph, then repositioning vertex to synthesize an intermediate graph. Although this method considers the case of large motion, several issues persist: 1) For raster images, the accuracy of vectorization significantly impacts the interpolation results. 2) The correspondence of the vectorized graph vertices is crucial when training, which is challenging to obtain on wild animation. 3) Discontinuous lines sometimes appear due to the mask of vertices. Compared with them, our model is more flexible based on raster images, and the two-stage pipeline design can allow manual fine-tuning to achieve higher quality results.

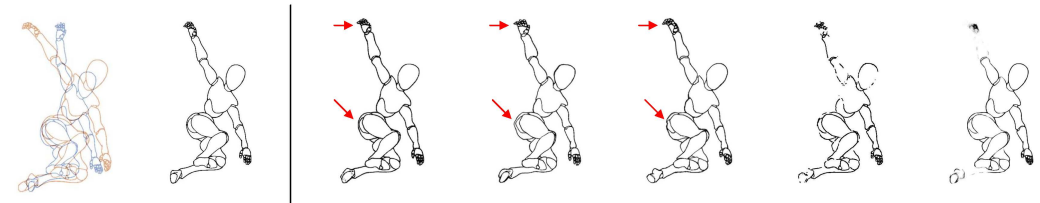
Video Generation. There are many researches based on diffusion models (Rombach et al. 2022) obtained remarkable results on video generation. Some approaches leverage text or a combination of text and images (Singer et al. 2023; Zeng et al. 2023; Guo et al. 2024) to generate video. While some research is dedicated to exploring how to generate videos with control (Zhang et al. 2024; Mou et al. 2024; Zhang, Rao, and Agrawala 2023), producing stable and smooth video sequences remains challenging, particularly when dealing with the large motion. Moreover, the transference of conventional video generation techniques to the domain of line art inbetweening is further complicated due to the domain gap. For animation, plot arrangement, the density of the rhythm, and the quality of inbetween frames are paramount. Consequently, our focus is on assisting animators with creating inbetween drawings rather than directly generating the final animation.

Proposed Method

In this section, we first discuss the limitations of the existing evaluation metrics and methods (Siyao et al. 2023; Shen et al. 2023), and then propose a simple yet effective pipeline to solve large motion of inbetweening for raster line art images.

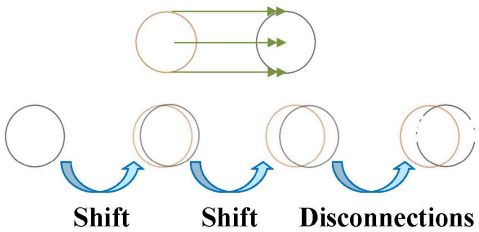
Limitations in Existing Methods and Metric

To obtain better inbetween quality, a key challenge is overcoming exaggerated and non-linear large motions of line art. Previous works on real-world video frame interpolation (Sim, Oh, and Kim 2021; Reda et al. 2022) attempted to directly use optical flow and multi-scale strategy to handle large motion. However, these approaches always failed on line art inbetweening because of the sparse pixels of the line art, which makes it difficult to obtain accurate optical flow and context information.



	Ours	TPS-forward	TPS-backward	AnimeInbet	AMT
CD (↓)	5.54	8.29	8.55	5.11	6.75
WCD (↓)	7.50	8.67	8.65	8.10	8.39

Figure 2: Visual comparison and corresponding metrics. The left side showcases the overlapped input frames along with the ground truth, while the right side displays the interpolation results and their evaluation metrics of inbetweening methods. TPS-forward and TPS-backward refer to the results obtained only using the flow estimated by the TPS module to warp the input's first and last frames, respectively. WCD is an enhanced metric that we proposed, which is better aligned with human visual perception, as detailed in Eq. (15).



CD (↓)	0	3.34	6.28	5.29

Figure 3: A simple evaluation case of orange and black circular objects. In the presence of line disconnections, the CD metric is misleading.

Previous works on line art inbetweening (Siyao et al. 2023; Narita, Hirakawa, and Aizawa 2019) have mainly adopted Chamfer Distance (CD) (Chen and Zwicker 2022) to evaluate the quality of the results. CD metirc is defined as

$$CD\left(\boldsymbol{b}_0, \boldsymbol{b}_1\right) = \frac{1}{2HWD} \sum \left(\boldsymbol{b}_0 \odot \mathcal{D}(\boldsymbol{b}_1) + \boldsymbol{b}_1 \odot \mathcal{D}(\boldsymbol{b}_0)\right), \tag{1}$$

where $\boldsymbol{b}_0$ and $\boldsymbol{b}_1$ are binary images with effective pixels is 1, $\mathcal{D}$ is the distance transform, and $\boldsymbol{b}_0 \odot \mathcal{D}(\boldsymbol{b}_1)$ denotes the distance map, wherein each effective pixel in $\boldsymbol{b}_0$ is labeled with the distance to the nearest effective pixel in $\boldsymbol{b}_1$, $\odot$ is element-wise multiplication, H, W, $D = \sqrt{H^2 + W^2}$ are height, width, and diameter, respectively.

Although these methods achieve favorable CD scores, we still observe significant issues with line disconnections as shown in Fig. 2. This raises the question: can CD truly reflect the quality of line art inbetweening? Let's consider a straightforward motion scenario: a circle is translated from left to right, as shown in Fig. 3. When the predicted circle aligns with the ground truth, the value of CD is 0. As the predicted results shift gradually, the CD value increases, which is expected. However, if random lines are erased from the circle, the CD value decreases, which means the quality is better. This is obviously unreasonable. The main reason for the error in quality evaluation is that CD simply averages the sum of $\boldsymbol{b}_0 \odot \mathcal{D}(\boldsymbol{b}_1)$ and $\boldsymbol{b}_1 \odot \mathcal{D}(\boldsymbol{b}_0)$, without considering the count of effective pixels.

To overcome the limitations of the existing metric, we introduce the perceptually aligned WCD metric and augment our evaluation with the EMD metric. Details are provided in the evaluation metrics section. To address the limitations of existing methods, we propose an effective two-stage approach for 2D hand-drawn animation inbetweening, mimicking animators' practice of anchoring motion with key features. Initially, we employ keypoint matching to detect local feature correspondences, followed by TPS for motion modeling, providing an initial estimate capable of handling large motions as shown in Fig. 2. TPS's plane deformation preserves line continuity. This module yields aligned frames that are subsequently refined by a simple network for precise motion tuning.

Thin-Plate Spline-based Inbetweening Method

Given two input raster line art images $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$, where $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{R}^{H \times W}$, and H and W are the height and the width, respectively. The goal of our method is to interpolate an intermediate raster line art image $\boldsymbol{y}_t$ with $\boldsymbol{y}_t \in \mathbb{R}^{H \times W}$ at arbitrary time $t \in (0, 1)$. The overall pipeline of our method is illustrated in Fig. 4.

TPS-based Coarse Motion Estimation. Due to nonlinear large motion in line art inbetweening, we employ a coarse motion estimation approach based on TPS to predict the motion instead of directly utilizing optical flow. First, we feed $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ into a pre-trained keypoints matching model to extract keypoints and establish the correspondence of each keypoint $\{(x_0^i, y_0^i), (x_1^i, y_1^i)\}_{i=1}^n$, where (x_0^i, y_0^i) and (x_1^i, y_1^i) represent coordinates of the i-th valid keypoint pair in $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$, respectively. In the following, we use p_0^i and p_1^i to represent (x_0^i, y_0^i) and (x_1^i, y_1^i) for simplicity. We in this paper apply pre-trained keypoints matching model and freeze it during training. To utilize the correspondence of keypoint pairs $\{p_0^i, p_1^i\}_{i=1}^n$ for predicting the rough motion between line art images, we draw inspiration from the thin-plate spline (TPS) transformation (Bookstein 1989). TPS

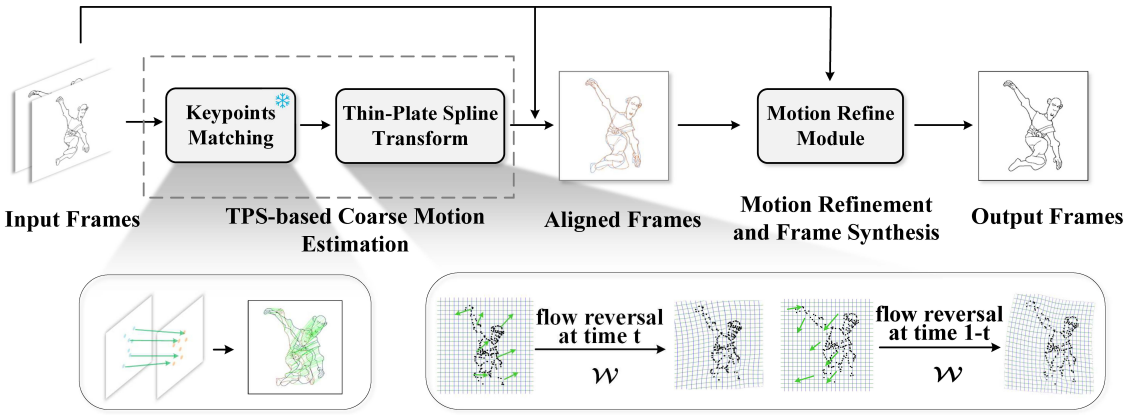


Figure 4: The overall pipeline of our approach begins with the initial estimation of coarse motion using a TPS-based transformation, which models point correspondence between key frames. Subsequently, we employ a simple motion refine module consists an optical flow network for estimating fine motion and a simple UNet model for synthesizing the inbetweening results.

transformation addresses the problem of interpolating surfaces over scattered data and modeling shape change as deformation. It utilizes thin-plate splines to interpolate surfaces based on a fixed set of points in the plane. Specifically, given a set of corresponding keypoint pairs$\{p_0^i, p_1^i\}_{i=1}^n$, we can directly use the TPS transformation function (Bookstein 1989) to model the motion between the two sets of keypoints, represented as

$$\mathcal{F}_{0\to1}(p_1) = \boldsymbol{A}\begin{bmatrix} p_1^{\mathrm{T}} \\ 1 \end{bmatrix} + \sum_{i=1}^{n} w_i \mathcal{U}\left(\left\|p_0^i - p_1\right\|_2\right), \tag{2}$$

where $p_1 \in \{p_1^i\}_{i=1}^n$ is an arbitrary keypoint coordinate of $\boldsymbol{y}_1$, $\boldsymbol{A} \in \mathbb{R}^{2\times3}$ means affine transformation matrix on linear space, n is the number of keypoints, $w_i \in \mathbb{R}^{2\times1}$ is the weight of the i-th keypoint of p_0^i, and $\mathcal{U}(k) = k^2\log(k)$ is the thin-plate spline kernel, which represents a mapping to displacement between the control point p_0^i and the final point p_1. The energy function $\mathcal{E}$ of TPS formula is minimized by

$$\mathcal{E} = \iint_{\mathbb{R}^2}\left(\left(\frac{\partial^2\mathcal{F}}{\partial^2 x^2}\right)^2 + 2\left(\frac{\partial^2\mathcal{F}}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\mathcal{F}}{\partial^2 y^2}\right)^2\right)dxdy. \tag{3}$$

To solve this optimization problem, we employ a linear system to compute the values of $\boldsymbol{A}$ and $\boldsymbol{W}$:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix}\begin{bmatrix} \boldsymbol{W}^{\mathrm{T}} \\ \boldsymbol{A}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{o} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{K}_{ij} = \mathcal{U}\left(\left\|p_0^i - p_0^j\right\|_2\right)$, $\boldsymbol{K} \in \mathbb{R}^{n\times n}$, i-th row of $\boldsymbol{P}$ is $[p_0^i, 1]$, $\boldsymbol{v} \in \mathbb{R}^{n\times2}$ is a column vector composed of $\{p_1^i\}_{i=1}^n$, $\boldsymbol{O} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{o} \in \mathbb{R}^{3\times2}$ are zero matrices.

After obtaining the affine matrix $\boldsymbol{A}$ and weight matrix $\boldsymbol{W}$ in TPS, we initialize the coordinate grid $\boldsymbol{g} \in \mathbb{R}^{H\times W\times2}$ and apply it to Eq. (2) to obtain the coarse motion map $\boldsymbol{m}_{0\to1}$ by

$$\boldsymbol{m}_{0\to1} = \mathcal{F}_{0\to1}(\boldsymbol{g}). \tag{5}$$

Similarly, we can also obtain the coarse motion flow $\boldsymbol{m}_{1\to0}$ from $\boldsymbol{y}_1$ to $\boldsymbol{y}_0$.

Now we get the coarse motion between two frames, and our goal is to interpolate intermediate frame $\boldsymbol{y}_t$ at time $t \in (0,1)$. Since $\boldsymbol{y}_t$ is unavailable, we cannot directly compute the motion $\boldsymbol{m}_{t\to0}$ and $\boldsymbol{m}_{t\to1}$. Drawing inspiration from (Jiang et al. 2018), we utilize linear models based on the assumption that the optical flow field is locally smooth to approximately compute the intermediate motion flow by

$$\begin{aligned} \boldsymbol{m}_{t\to0} = t\boldsymbol{m}_{1\to0} &= t((1-t)\boldsymbol{m}_{1\to0} + t\boldsymbol{m}_{1\to0}) \\ &= -(1-t)t\boldsymbol{m}_{0\to1} + t^2\boldsymbol{m}_{1\to0}. \end{aligned} \tag{6}$$

Similarly, we can obtain the motion $\boldsymbol{m}_{t\to1}$.

The coarsely aligned frames are obtained through backward warping operation $\mathcal{W}$:

$$\tilde{\boldsymbol{y}}_{0\to t} = \mathcal{W}\left(\boldsymbol{y}_0, \boldsymbol{m}_{t\to0}\right), \tilde{\boldsymbol{y}}_{1\to t} = \mathcal{W}\left(\boldsymbol{y}_1, \boldsymbol{m}_{t\to1}\right). \tag{7}$$

Motion Refinement Module After performing the coarse alignment operation, we have reduced the motion between the two frames. At this stage, we feed the coarsely aligned frames into optical flow methods to further refine the motion. We employ a lightweight and efficient module IFBlock (Huang et al. 2022) to iteratively compute the residual motion flows $\Delta\boldsymbol{m}_{t\to1}$, $\Delta\boldsymbol{m}_{t\to0}$ and fusion map $\boldsymbol{f}$:

$$\begin{aligned} &\Delta\boldsymbol{m}_{t\to1}^{\alpha}, \Delta\boldsymbol{m}_{t\to0}^{\alpha}, \boldsymbol{f}^{\alpha} = \\ &\mathrm{IFBlock}_{\alpha}\left(\tilde{\boldsymbol{y}}_{0\to t}, \tilde{\boldsymbol{y}}_{1\to t}, \Delta\boldsymbol{m}_{t\to1}^{\alpha-1}, \Delta\boldsymbol{m}_{t\to0}^{\alpha-1}, \boldsymbol{f}^{\alpha-1}\right), \end{aligned} \tag{8}$$

where $\mathrm{IFBlock}_{\alpha}$ is the α-th IFBlock. When α is equal to 1, we directly pass $\tilde{\boldsymbol{y}}_{0\to t}$ and $\tilde{\boldsymbol{y}}_{1\to t}$ into the IFblock. The total number of IFBlocks is 3 in our method, and we use the output in the last iteration as the residual motion flows and fusion map. Then we add the residual motion flows to coarse motion flow yielding refined motion flows $\boldsymbol{s}_{t\to0}$ and $\boldsymbol{s}_{t\to1}$. We can obtain the refinedly aligned frames $\hat{\boldsymbol{y}}_{0\to t}$ and $\hat{\boldsymbol{y}}_{1\to t}$ with refined motion flows.

$$\hat{\boldsymbol{y}}_{0\to t} = \mathcal{W}\left(\boldsymbol{y}_0, \boldsymbol{s}_{t\to0}\right), \hat{\boldsymbol{y}}_{1\to t} = \mathcal{W}\left(\boldsymbol{y}_1, \boldsymbol{s}_{t\to1}\right). \tag{9}$$

By combining the two refinedly aligned frames, weighted by time t, we can obtain the initial inbetweening result

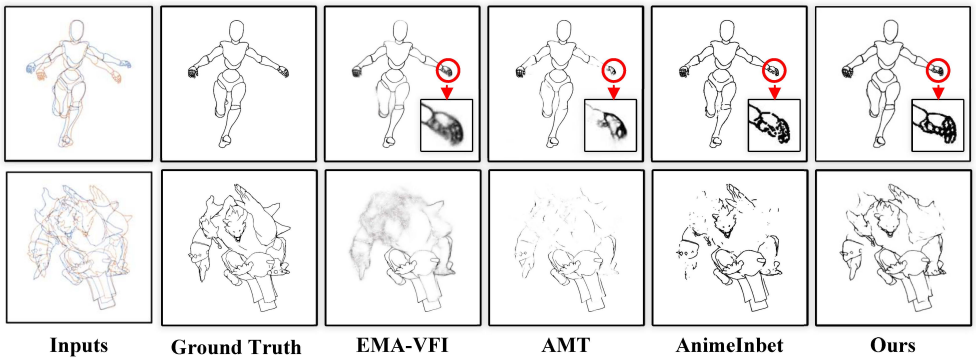


Figure 5: Inbetweening results comparison on the MixiamoLine240 dataset.

$\hat{\boldsymbol{y}}_t$. Moreover, we utilize a multi-scale feature alignment approach where, for each scale, we warp the features extracted from $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ to obtain multi-scale aligned features $\{\boldsymbol{y}_0^l\}_{l=1}^L$ and $\{\boldsymbol{y}_1^l\}_{l=1}^L$. And we set $L = 4$ in our experiments.

Finally, we utilize a simple UNet (Ronneberger, Fischer, and Brox 2015) for predicting the residual inbetweening, whose input is the concatenation of $\boldsymbol{y}_0, \boldsymbol{y}_1, \hat{\boldsymbol{y}}_{0\to t}, \hat{\boldsymbol{y}}_{1\to t}, \boldsymbol{s}_{t\to 0}, \boldsymbol{s}_{t\to 1}, \boldsymbol{f}, \{\boldsymbol{y}_0^l\}_{l=1}^L, \{\boldsymbol{y}_1^l\}_{l=1}^L$. The UNet architecture consists of $(L+1)$ scales, from 0 to L. At the 0-th scale, the concatenation of input is first processed for feature extraction. For subsequent scales, features in l-th scale of UNet are concatenated with the l-th warped features $\boldsymbol{y}_0^l$ and $\boldsymbol{y}_1^l$ and then fed into the l-th encoder layer. Finally, we add the residual inbetweening result $\Delta\hat{\boldsymbol{y}}_t$ to the initial inbetweening result $\hat{\boldsymbol{y}}_t$ to obtain final interpolation result $\boldsymbol{y}_t$.

Loss Functions

Most of the video frame interpolation works employ $\mathcal{L}_1$ loss to optimize PSNR (Chen and Zwicker 2022; Zhang et al. 2023). However, for line art, pixel-wise measurement is not suitable as slight misalignment can result in significant pixel-wise differences. Thus we use distance transform $\mathcal{D}$ to convert a line art frame into a distance transform map. The implementation involves using a differentiable convolutional distance transform layer (Pham, Dovletov, and Pauli 2021). By minimizing the difference between the distance transform map of predicted frames and the ground truth, we ensure the similarity of the line structure. The loss is represented as

$$\mathcal{L}_{dt} = \frac{1}{N}\sum_t \left\|\mathcal{D}(\boldsymbol{y}_t) - \mathcal{D}(\boldsymbol{y}_t^{gt})\right\|, \tag{10}$$

where N is the interpolated frame number between two key frames.

We noticed that better inbetweening results usually yield a sum of effective pixels roughly equivalent to that of the ground truth. Thus, we introduce a loss $\mathcal{L}_{cnt}$ to constrain the sum of effective pixels

$$\mathcal{L}_{cnt} = \frac{1}{N}\sum_t \left\|\sum \mathrm{ReLU}(\boldsymbol{y}_t - \eta) - \sum \mathrm{ReLU}(\boldsymbol{y}_t^{gt} - \eta)\right\|, \tag{11}$$

where η is the threshold to delimit the effective pixels.

To make the resulting line art clearer, we add binarization loss (Xie et al. 2021) $\mathcal{L}_{bi}$ to encourage the network to generate black or white pixels

$$\mathcal{L}_{bi} = \frac{1}{N}\sum_t \left\||\boldsymbol{y}_t - 0.5| - 0.5\right\|_2. \tag{12}$$

Further, we adopt LPIPS (Zhang et al. 2018) loss $\mathcal{L}_{lpips}$ to improve the perceptual quality of line art. The total loss $\mathcal{L}$ is defined as

$$\mathcal{L} = \mathcal{L}_{dt} + \lambda_{cnt}\mathcal{L}_{cnt} + \lambda_{bi}\mathcal{L}_{bi} + \lambda_{lpips}\mathcal{L}_{lpips}, \tag{13}$$

where λ_{cnt}, λ_{bi}, λ_{lpips} are hyperparameters.

Experiments

Implementation Details

We implemented our model in PyTorch (Paszke et al. 2019). We apply GlueStick (Pautrat et al. 2023) as our keypoints matching model. The training and testing were conducted on MixiamoLine240 dataset (Siyao et al. 2023), a line art dataset with ground truth geometrization and vertex matching labels. It's worth noting that during training, we utilized only raster images from the dataset. We set the frame gap $N = 5$ during training, and tested on the test set with the gaps $N = 1, 5, 9$ respectively. For augmentation, we applied random temporal flipping. Our model was trained at the resolution of 512×512, and tested at the original resolution 720×720. We employed the Adam (Kingma and Ba 2015) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ at a learning rate of 1×10^{-4} for 50 epochs. The training and testing were performed on an NVIDIA RTX A6000 GPU. The hyperparameters λ_{lpips}, λ_{cnt}, λ_{bi} and η were set as 5, 5, 1×10^{-3} and 0.9, respectively.

Evaluation Metrics

PSNR and SSIM (Wang et al. 2004) are common evaluation metrics in video interpolation tasks but they often fall short when applied to animation videos (Chen and Zwicker 2022). Previous works have adopted CD as the evaluation metric. However, as discussed in the previous section, it has certain limitations. While it can reflect the alignment with the ground truth, it is affected by discontinuous lines.

Method	$N = 1$ (validation / test)		$N = 5$ (validation / test)		$N = 9$ (validation / test)	
EMA-VFI	3.01/8.62/5.84	3.70/8.59/6.16	15.21/12.04/11.1	17.47/12.00/11.31	28.44/14.94/13.95	31.30/14.86/14.05
RIFE	3.43/8.28/5.48	2.93/8.39/3.81	14.14/11.27/8.37	15.82/11.15/9.63	25.70/13.67/11.61	28.27/13.59/11.77
AMT	2.43/7.33/2.46	2.84/7.31/2.34	14.34/9.58/3.83	16.65/10.13/4.07	22.39/11.62/5.02	25.77/12.21/5.24
PerVFI	2.88/7.91/2.98	3.32/7.79/2.78	11.90/9.23/3.29	12.62/9.05/3.37	19.80/10.20/3.42	21.38/10.09/3.65
EISAI	3.58/8.91/6.26	4.02/8.56/5.86	13.06/9.80/6.29	14.46/9.57/5.80	22.41/10.69/6.27	24.46/10.54/5.79
AnimeInbet	2.51/7.54/3.02	3.07/7.93/3.74	9.33/8.53/3.51	**10.74**/9.84/4.91	16.08/10.01/4.36	**17.76**/11.54/5.74
Ours	**2.34/7.32/1.01**	**2.77/7.28/2.24**	**9.24/7.99/1.34**	10.79/**8.39/2.88**	**15.89/8.85/1.69**	18.09/**9.48/3.37**

Table 1: Quantitative comparison (CD↓ / WCD↓ / EMD↓) with state-of-the-art methods on different frame gaps. The first place and runner-up are highlighted in bold and underlined, respectively. CD is scaled by 10^5, WCD by 10^4, EMD by 10^3.

To address this issue, we introduce an improved metric based on CD for evaluating line art visual quality, called WCD, aiming for better alignment with human perception. Based on the observation that when the line art image maintains a similar overall structure to the ground truth and the lines remain continuous, the effective pixels should be approximately equal to those in the ground truth, we introduce a weighted factor $\mathcal{H}(\boldsymbol{b}_0, \boldsymbol{b}_1) \in [0.5, 1)$ derived from the difference between the effective pixels of the two line art images. Furthermore, we apply a non-linear softplus function mapping $\mathcal{G}(\cdot)$ to the values obtained from the distance transform, imposing a greater penalty on values with larger distance. WCD can be represented as

$$\mathcal{H}(\boldsymbol{b}_0, \boldsymbol{b}_1) = \text{Sigmoid}\left(\frac{||\boldsymbol{b}_0|_n - |\boldsymbol{b}_1|_n|}{\min\left(|\boldsymbol{b}_0|_n, |\boldsymbol{b}_1|_n\right)}\right), \tag{14}$$

$$\begin{aligned} WCD\left(\boldsymbol{b}_0, \boldsymbol{b}_1\right) = \frac{\mathcal{H}(\boldsymbol{b}_0, \boldsymbol{b}_1)}{HWD} \sum (&\mathcal{G}\left(\boldsymbol{b}_0 \odot \mathcal{D}(\boldsymbol{b}_1)\right) \\ &+ \mathcal{G}\left(\boldsymbol{b}_1 \odot \mathcal{D}(\boldsymbol{b}_0)\right)), \end{aligned} \tag{15}$$

where $|\cdot|_n$ means the number of effective pixels.

As distance map only considers the nearest neighbour of a point, it falls short in adequately representing the distribution of these points. Therefore, we advocate for the adoption of EMD (Nguyen et al. 2021; Achlioptas et al. 2018), a metric commonly used in 3D point cloud representation, to assist in the assessment of line art quality. Considering the potential computational cost of EMD calculation (Pele and Werman 2009), we compute EMD separately for the horizontal and vertical directions and then average the results.

Comparison with State-of-the-arts

We categorized the methods for comparison into three groups. For real-word video frame interpolation, we choose RIFE (Huang et al. 2022), EMA-VFI (Zhang et al. 2023), AMT (Li et al. 2023) and PerVFI (Wu et al. 2024). For animation interpolation and line art inbetweening, we choose EISAI (Chen and Zwicker 2022) and AnimeInbet (Siyao et al. 2023), respectively. The AnimeInbet is a vector-based method, while the others are raster image-based methods. We finetune these methods with the frame gap $N = 5$. We tested each model with frame gaps of 1, 5, and 9. For each gap N, evaluation metrics are based on the average of these generated N frames. Predicted raster images are binarized by setting pixels smaller than 0.95 to 0.

As shown in Tab. 1, with the frame gap of 1, the method of video frame interpolation based on optical flow (Li et al. 2023) can also achieves commendable performance. However, as the motion gradually increases, the performance of these interpolation models based on real-world videos significantly degrades. Though previous work focused on line art inbetweening (Siyao et al. 2023) has demonstrated some potential in handling large motions, our method outperforms on both test and validation sets.

In Fig. 2, a line art inbetweening result and corresponding evaluation metrics of our method are compared with other approaches. It can be observed that AnimeInbet achieved the highest score in CD, despite the presence of noticeable missing lines. Although our method not exhibit the highest performance in CD, it attains the best result in WCD that is more aligned with human visual perception. More inbetweening results are illustrated in Fig. 5. For most cases, EMA-VFI tends to blur, and AMT causes large areas of lines to disappear, AnimeInterp often generates discontinuous lines, while our method produces more continuous and clear lines.

Additionally, due to the simplicity of our network architecture, our method exhibits advantages in inference time and parameters compared to other methods, as detailed in the supplementary file. Notably, although the generative model-based method ToonCrafter (Xing et al. 2024) has shown potential in inbetweening, it performed extremely unstably on test data, where the interpolated frames frequently turned out to be identical to either the first or last frame.

To get the color animation frame, it seems to be a solution to colorize first and then interpolate the frame. However, existing interpolation methods on color frames also suffer from line artifacts. Moreover, modifying linearts is easier than modifying colored frames, making a line inbetweening algorithm can greatly facilitate animators. We compare the line art frame interpolation followed by coloring and colorizing first followed by color frame interpolation. Our method is still competitive, detailed in the supplementary.

User Study

User study can be conducted by using pairwise comparison scaling methods such as the Thurstone's Case V model (Maydeu-Olivares 2004), or by having participants select the best option. To keep consistent with existing animation interpolation methods (Chen and Zwicker 2022), we choose the

	Ours	AnimeInbet	EMA-VFI
$N = 1$	57.50%	31.50%	11.00%
$N = 5$	75.17%	16.33%	8.50%
$N = 9$	69.50%	16.50%	14.00%
All	**70.50%**	19.40%	10.10%

Table 2: User study on evaluating inbetweening performance of competing methods.

	CD	WCD	EMD
$N = 1$	55.07%	60.67%	52.25%
$N = 5$	57.56%	77.41%	66.57%
$N = 9$	28.66%	68.90%	59.15%
All	51.74%	**72.50%**	61.73%

Table 3: User study on the consistency of visual perception and evaluation metrics.

latter approach in our user study. Twenty participants were recruited, and each was presented with 50 sets of images randomly selected from the results of EMA-VFI, AnimeInbet, and our method. As shown in Tab. 2, most participants preferred the results of our method, especially for large motion scenarios. The user study also demonstrates the validity of our proposed metric WCD compared with CD in Tab. 3. When the frame gap is small, the differences between the different metrics are not obvious. However, when the frame gap is 9, only approximately 29% of the participants preferred the result with better CD, contrasting with 69% and 59% of participants who preferred results with WCD and EMD, respectively.

Ablation Study

We conduct ablations to verify the effectiveness of certain modules and losses, evaluating the ablation study on a validation set with a frame gap of 5. Initially, we verified the effectiveness of the TPS-based coarse motion estimation module that is the core of our approach. As shown in Tab. 4, "w/o MRM" refers to the removal of the entire Motion Refine Module (MRM), while "w/o IFBlock" refers to the removal of the IFBlock, which is a component of MRM employed to estimate the motion between two aligned frames. It is observed that employing only coarse motion yields satisfactory results in Fig. 6, though the evaluation metrics are not as favorable due to the misalignment with the ground truth. The absence of IFBlock will result in a decrease in the model's performance as demonstrated in Tab. 4. The impact on metrics is not substantial, given that the preceding coarse motion already provides enough information.

In Tab. 5, we analyze the effectiveness of different loss terms during training. The distance transform loss $\mathcal{L}_{dt}$ has a more pronounced influence on CD, while $\mathcal{L}_{cnt}$ plays a more critical role in optimizing WCD and EMD. The binarization loss $\mathcal{L}_{bi}$ mainly influences the visual performance by enhancing line clarity, with its impact on the evaluation metrics being comparatively marginal.

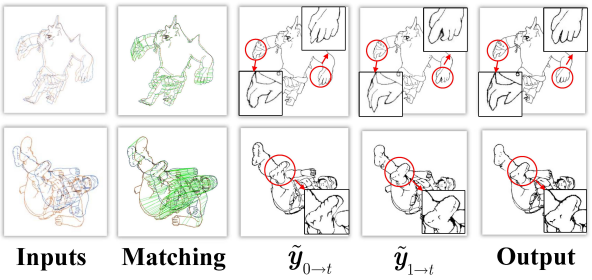


Figure 6: Visual comparison of ablation study on modules, where TPS module can well predict poses in intermediate frames, while MRM bring benefits to detail enhancement.

	CD	WCD	EMD
w/o MRM	11.800	8.850	4.233
w/o IFBlock	9.314	8.015	2.625
full model	**9.235**	**7.994**	**1.343**

Table 4: Ablation study on modules.

$\mathcal{L}_{bi}$	$\mathcal{L}_{dt}$	$\mathcal{L}_{cnt}$	CD	WCD	EMD
✗	✓	✗	9.236	8.592	3.750
✓	✓	✗	9.330	8.777	3.732
✓	✗	✓	9.991	8.112	1.996
✓	✓	✓	**9.235**	**7.994**	**1.343**

Table 5: Ablation study on loss functions.

Limitations

While our approach excels in modeling motion from coarse to fine through keypoint correspondence, challenges emerge when the keypoint matching model produces significant errors or misses in its estimation. These errors tend to propagate through subsequent motion estimations, potentially leading to artifacts, especially in regions with substantial occlusion. These matching errors can be rectified through manual fine-tuning. Given the two-stage nature of our approach, animators can effortlessly rectify false matches and add some crucial matches that might have gone undetected.

Conclusion

In this work, we propose a novel framework for addressing the animation line inbetweening problem based on raster images. By leveraging thin-plate spline-based transformation, more accurate estimates of coarse motion can be obtained by modeling point correspondence between key frames, particularly in scenarios involving significant motion amplitudes. Subsequently, a motion refine module is adopted to enhance motion details and synthesize final frames. To mitigate the inherent shortcomings of CD, we introduce an improved metric called WCD that aligns more closely with human visual perception. Our quantitative and qualitative evaluations demonstrate the superiority of our method over existing approaches, highlighting its potential impact in relevant fields.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62172127, U22B2035), and the Natural Science Foundation of Heilongjiang Province (YQ2022F004).

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning representations and generative models for 3D point clouds. In *ICML*, 40–49.

Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE TPAMI*, 11(6): 567–585.

Chen, S.; and Zwicker, M. 2022. Improving the perceptual quality of 2D animation interpolation. In *ECCV*, 271–287.

Guo, Y.; Yang, C.; Rao, A.; Liang, Z.; Wang, Y.; Qiao, Y.; Agrawala, M.; Lin, D.; and Dai, B. 2024. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. In *ICLR*.

Hu, M.; Jiang, K.; Zhong, Z.; Wang, Z.; and Zheng, Y. 2024. IQ-VFI: Implicit Quadratic Motion Estimation for Video Frame Interpolation. In *CVPR*, 6410–6419.

Huang, Z.; Zhang, T.; Heng, W.; Shi, B.; and Zhou, S. 2022. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 624–642.

Jiang, H.; Sun, D.; Jampani, V.; Yang, M.-H.; Learned-Miller, E.; and Kautz, J. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 9000–9008.

Kalluri, T.; Pathak, D.; Chandraker, M.; and Tran, D. 2023. Flavr: Flow-agnostic video representations for fast frame interpolation. In *WACV*, 2071–2082.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Lee, H.; Kim, T.; Chung, T.-y.; Pak, D.; Ban, Y.; and Lee, S. 2020. Adacof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, 5316–5325.

Li, Z.; Geng, Z.; Kang, Z.; Chen, W.; and Yang, Y. 2022. Eliminating gradient conflict in reference-based line-art colorization. In *ECCV*, 579–596.

Li, Z.; Zhu, Z.-L.; Han, L.-H.; Hou, Q.; Guo, C.-L.; and Cheng, M.-M. 2023. AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation. In *CVPR*, 9801–9810.

Liu, C.; Zhang, G.; Zhao, R.; and Wang, L. 2024. Sparse Global Matching for Video Frame Interpolation with Large Motion. In *CVPR*, 19125–19134.

Maydeu-Olivares, A. 2004. *Thurstone's Case V Model: A Structural Equations Modeling Perspective*, 41–67. Dordrecht: Springer Netherlands. ISBN 978-1-4020-1958-6.

Mou, C.; Wang, X.; Xie, L.; Wu, Y.; Zhang, J.; Qi, Z.; and Shan, Y. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, 4296–4304.

Narita, R.; Hirakawa, K.; and Aizawa, K. 2019. Optical flow based line drawing frame interpolation using distance transform to support inbetweenings. In *ICIP*, 4200–4204.

Nguyen, T.; Pham, Q.-H.; Le, T.; Pham, T.; Ho, N.; and Hua, B.-S. 2021. Point-set distances for learning representations of 3D point clouds. In *ICCV*, 10478–10487.

Niklaus, S.; Mai, L.; and Liu, F. 2017. Video frame interpolation via adaptive separable convolution. In *ICCV*, 261–270.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32.

Pautrat, R.; Suárez, I.; Yu, Y.; Pollefeys, M.; and Larsson, V. 2023. Gluestick: Robust image matching by sticking points and lines together. In *ICCV*, 9706–9716.

Pele, O.; and Werman, M. 2009. Fast and robust earth mover's distances. In *ICCV*, 460–467.

Pham, D. D.; Dovletov, G.; and Pauli, J. 2021. A differentiable convolutional distance transform layer for improved image segmentation. In *GCPR*, 432–444.

Reda, F.; Kontkanen, J.; Tabellion, E.; Sun, D.; Pantofaru, C.; and Curless, B. 2022. Film: Frame interpolation for large motion. In *ECCV*, 250–266.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241.

Shang, W.; Ren, D.; Yang, Y.; Zhang, H.; Ma, K.; and Zuo, W. 2023. Joint Video Multi-Frame Interpolation and Deblurring under Unknown Exposure Time. In *CVPR*, 13935–13944.

Shen, J.; Hu, K.; Bao, W.; Chen, C. W.; and Wang, Z. 2023. Bridging the Gap: Fine-to-Coarse Sketch Interpolation Network for High-Quality Animation Sketch Inbetweening. *arXiv preprint arXiv:2308.13273*.

Sim, H.; Oh, J.; and Kim, M. 2021. Xvfi: extreme video frame interpolation. In *ICCV*, 14489–14498.

Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; Parikh, D.; Gupta, S.; and Taigman, Y. 2023. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *ICLR*.

Siyao, L.; Gu, T.; Xiao, W.; Ding, H.; Liu, Z.; and Loy, C. C. 2023. Deep Geometrized Cartoon Line Inbetweening. In *ICCV*, 7291–7300.

Siyao, L.; Zhao, S.; Yu, W.; Sun, W.; Metaxas, D.; Loy, C. C.; and Liu, Z. 2021. Deep animation video interpolation in the wild. In *CVPR*, 6587–6595.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4): 600–612.

Wu, G.; Tao, X.; Li, C.; Wang, W.; Liu, X.; and Zheng, Q. 2024. Perception-Oriented Video Frame Interpolation via Asymmetric Blending. In *CVPR*, 2753–2762.

Wu, H.; Zhang, X.; Xie, W.; Zhang, Y.; and Wang, Y. 2023. Boost Video Frame Interpolation via Motion Adaptation. In *BMVC*.

Xie, M.; Xia, M.; Liu, X.; Li, C.; and Wong, T.-T. 2021. Seamless manga inpainting with semantics awareness. *ACM TOG*, 40(4): 1–11.

Xing, J.; Liu, H.; Xia, M.; Zhang, Y.; Wang, X.; Shan, Y.; and Wong, T.-T. 2024. ToonCrafter: Generative Cartoon Interpolation. *arXiv preprint arXiv:2405.17933*.

Yang, W. 2017. Context-aware computer aided inbetweening. *IEEE TVCG*, 24(2): 1049–1062.

Yoo, J.-S.; Lee, H.; and Jung, S.-W. 2023. Video object segmentation-aware video frame interpolation. In *ICCV*, 12322–12333.

Zeng, Y.; Wei, G.; Zheng, J.; Zou, J.; Wei, Y.; Zhang, Y.; and Li, H. 2023. Make pixels dance: High-dynamic video generation. *arXiv preprint arXiv:2311.10982*.

Zhang, G.; Zhu, Y.; Wang, H.; Chen, Y.; Wu, G.; and Wang, L. 2023. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, 5682–5692.

Zhang, L.; Li, C.; Simo-Serra, E.; Ji, Y.; Wong, T.-T.; and Liu, C. 2021. User-guided line art flat filling with split filling mechanism. In *CVPR*, 9889–9898.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*, 3836–3847.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 586–595.

Zhang, Y.; Wei, Y.; Jiang, D.; ZHANG, X.; Zuo, W.; and Tian, Q. 2024. ControlVideo: Training-free Controllable Text-to-video Generation. In *ICLR*.