DOI: 10.1111/cgf.15044
EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Sketch Video Synthesis

Yudian Zheng[1] Xiaodong Cun[2] Menghan Xia[2] Chi-Man Pun[3]

[1]Saarland University [2]Tencent AI Lab [3]University of Macau

http://sketchvideo.github.io

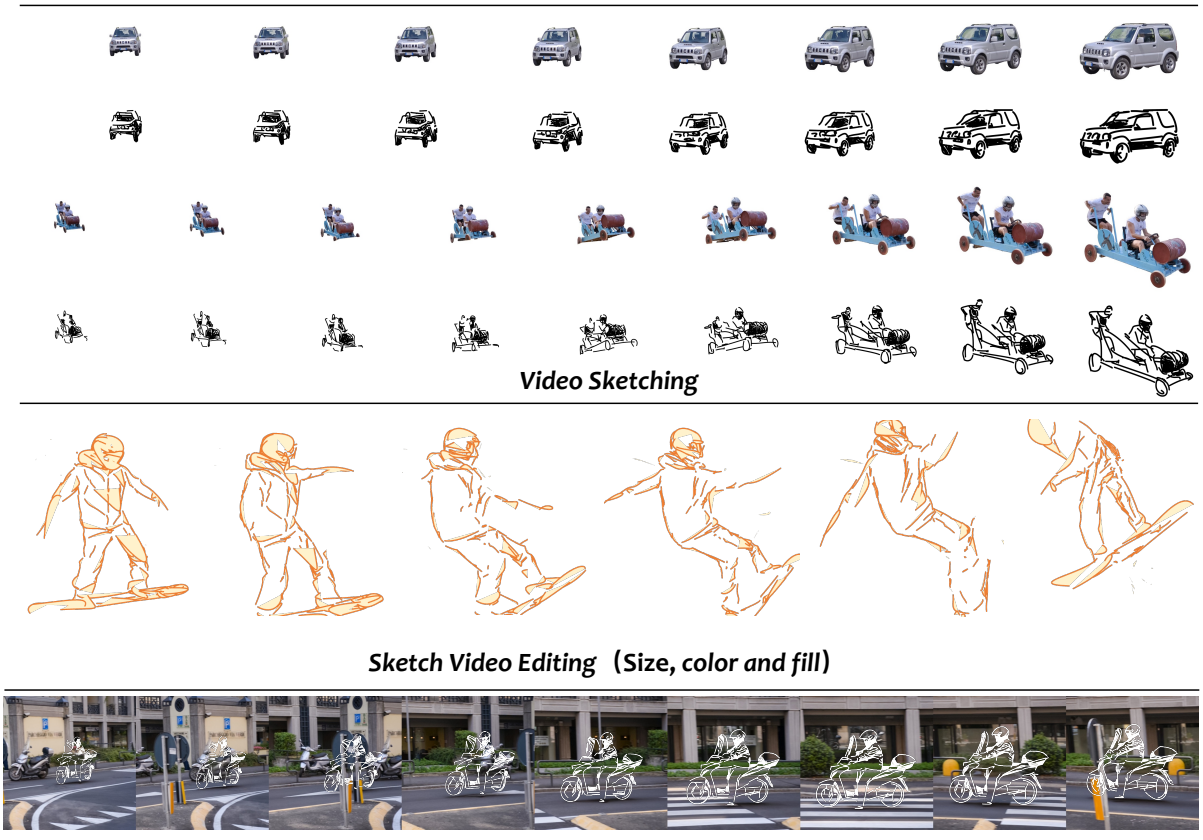

Figure 1: *Given an input video (with the foreground object), we introduce a novel method for sketching the video using Bézier Curves so that the video can be represented by scalable vector graphics (SVG). The generated sketch video maintains semantic alignment with the input and demonstrates temporal consistency. The flexibility of vector lines enables various rendering techniques, including resizing, color filling, and overlaying doodles on the original background images, allowing for the creation of diverse artistic effects.*

Abstract

Understanding semantic intricacies and high-level concepts is essential in image sketch generation, and this challenge becomes even more formidable when applied to the domain of videos. To address this, we propose a novel optimization-based framework for sketching videos represented by the frame-wise Bézier Curves. In detail, we first propose a cross-frame stroke initialization approach to warm up the location and the width of each curve. Then, we optimize the locations of these curves by utilizing a semantic loss based on CLIP features and a newly designed consistency loss using the self-decomposed 2D atlas network. Built upon these design elements, the resulting sketch video showcases notable visual abstraction and temporal coherence. Furthermore, by transforming a video into vector lines through the sketching process, our method unlocks applications in sketch-based video editing and video doodling, enabled through video composition.

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

1. Introduction

Freehand drawing is a widely adopted method for quickly prototyping ideas across various domains [GSH*19; XHY*20]. This approach embodies simplicity, abstraction, and adaptability, empowering individuals to effectively express their concepts. Furthermore, skilled artists can develop distinctive artistic styles through freehand drawing. While sketching is a common practice for images, often using formats like scalable vector graphics (SVG), there has been limited exploration of its application in the context of sketching videos. This uncharted territory makes the exploration of sketch videos an intriguing and meaningful endeavor.

While traditional approaches, such as edge detection methods [XT15; Can86] excel in rendering realistic sketches, they struggle to create more expressive and abstract representations due to their reliance on mathematical and geometric operations. In an attempt to incorporate semantic awareness, previous sketching methods have attempted to learn human-like sketches from a collected dataset in different levels of abstraction and styles [BSM*13; LLM*19; KP20] at the pixel level. While these data-driven methods imitate human sketches, the requirements and quality of the relevant datasets restrict the output. As introduced by recent image sketching works [VPB*22; VACS22], line drawings are defined using the control points of Bézier Curves and are optimized to represent the scene. These methods employ multi-scale deep perceptual losses [RKH*21] to bridge the gap between generated sketches and real scenes, bypassing the constraints of traditional datasets and yielding diverse results. We follow these frameworks to represent video sketches in vector format. However, if we simply apply image-based sketching [VPB*22] in a frame-wise manner without careful consideration, the strokes will converge into local minima rapidly. Additionally, the flickering issue of generated video is not easily resolved through conventional video deflickering algorithms [LXC20; LRZC23], especially when dealing with vector graphics.

To overcome the challenges mentioned above, we introduce an optimization-based framework aimed at generating sketch videos in vector format that exhibit both semantic alignment and temporal consistency. To achieve this goal, we leverage Neural Layered Atlas (NLA) [KOWD21] to our tasks for multiple purposes. NLA is first proposed for video editing, it maps each point in video to a uniform global UV map, so that it can guarantee the correspondences across frames and help for temporal point consistency. In detail, the process of generating a high-quality video sketch involves careful initialization and continuous optimization of the sketch video. We begin by carefully selecting the initial locations for candidate points, where an effective initialization is crucial for avoiding unfavorable local minima [VPB*22] and accurately conveying the video's semantics. To achieve this, our initialization approach utilizes semantic-aware edges, derived from salient maps obtained from the combination of CLIP features [RKH*21] and XDoG edge detection [WKO12]. Subsequently, these selected points are propagated to all frames and optimized to their initial positions using a pre-trained NLA. Then, we optimize the location of these points using several losses so that they can ensure both semantic alignment and temporal consistency. We transform the candidate points to Bézier Curves and utilize a differentiable rasterizer [LLMR20] to render them to the frame-wise images. For semantic abstraction, we utilize the pre-trained CLIP image encoder as a feature extractor and compute losses between the rendered one and the real video frame. For temporal consistency, we ensure consistency of vector points via the pre-trained NLA [KOWD21] so that they can be consistent from a global view. Based on these techniques, the proposed method can successfully generate the abstraction sketches of the specific given video.

In addition to streamlining the process of sketching videos, our approach paves the way for various video applications. For instance, it allows for the creation of colorful videos by applying drawing techniques to a single frame. Furthermore, our method introduces novel possibilities in video editing, such as substituting the original content by integrating sketches into the scenes. Additionally, our approach enables the generation of video doodles to enhance other video content.

The contribution of this paper can be summarized as:

- We first tackle the problem of generating scalable abstract videos via several Bézier curves.
- By utilizing the consistency of pre-trained video implicit representation [KOWD21], we propose a novel point initialization method and a temporal consistency loss for video sketching synthesis.
- Our method generates the animated vector sketch from the input video, enabling multiple new applications of video editing and doodles.

2. Related Work

2.1. Doodling and Abstraction

Doodling and abstraction are common forms of artistic expression that use few and sparse curves to depict objects or scenes abstractly. Early works in sketching, such as traditional edge detection methods [Can86; XT15], effectively describe the structure and semantics of images. While these methods produce clear and coherent sketch videos, they lean towards excessive realism, often neglecting the artistic preferences of viewers. Combining edge detection methods with line drawing video stylization [BBM*16] generates abstract images, but it often overlooks the holistic aspect of video frames and loses video consistency. This issue can be alleviated to some degrees by incorporating temporal noise control methods [NSC*11]. Recent data-driven methods primarily focus on generating human-like sketches through domain adaptation using carefully curated datasets [HE17; AMFM10]. These methods offer the ability to create various styles and levels of abstraction. However, the resulting style is often closely tied to the specific dataset characteristics, making it less suitable for unstructured and previously unseen data. Extending these methods to handle videos, which are typically supervised by image datasets, is also a challenging endeavor. Another method, such as video doodling [YBN*23], is used in sketch addition on video, but it is not satisfied with a global level of abstraction.

2.2. Vector Sketching Generation

There are some image-to-vectors methods [RGLM21; CDAT20; DYH*20] that can typically produce pixel-aligned results. How-

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

ever, sketching demands a higher level of abstraction and greater continuity between lines. Recent advancements in differential gradient algorithms, such as DiffVG [LLMR20], have made it feasible to optimize images and even SVG representations within the pre-trained CLIP [RKH*21] space. For example, CLIPDraw [FSW22] explores the potential of optimizing SVG images to generate drawings that closely align with text prompts, leveraging the guidance of CLIP. Similarly, Tian and Ha[TH22] have designed an evolutionary algorithm to abstract images using vector triangles.

Regarding closely related works, CLIPasso [VPB*22] specifically applies the technique of CLIPDraw [FSW22] to generate object sketches, and CLIPascene [VACS22] extends the optimization process to incorporate background elements. Furthermore, similar optimization methods have been employed in the domain of artistic fonts [IVH*23] with the diffusion model [RBL*21]. Text-based vector generation has attracted research attention, as evidenced by Wu *et al*. [WSML23], who propose a transformer-based method for generating icons from text auto-regressively. Different from the vectorization method for images, our approach focuses on generating sketch videos that need to keep temporal consistency, and directly using the image-based method will fall into local minimal due to initialization.

2.3. Rotoscope and Animation

Rotoscoping [Sab97], as a traditional animation technique, creates animated sequences by tracing over live-action footage frame by frame. This method yields a stylized video while preserving the motion trails and essential structure information from the source video. In response to diverse demands, various methods have been proposed, such as dealing with keyframes, preserving higher contrast regions, or tracking contours in the video sequence [WOG06; AHSS04]. The quality of the generated video often relies on user interaction [OH11; Aga02]. Traditional techniques serve as pre-processing steps that filter out fine-grained information, handling relatively simple effects and leaving more intricate work for the artist. Similar principles are applied in animation. Through the use of sparse tracking points, a sketch can be animated accordingly [SBF*18; SZL*23]. However, this method is region-specific and encounters challenges when dealing with complex videos and general motion, primarily focusing on extracting and adapting crucial motion curves. Recently, advancements in deep learning have played a pivotal role in contributing to the refined animation of sketches, enabling the creation of nuanced and high-quality outcomes [YYF*22].

2.4. Video Editing and Temporal Consistency

Video editing and stylization have a long history within the computer vision and graphics communities. Various attempts have been made to achieve stylization [FJL*16; JST*19]. However, these methods may face challenges in maintaining tracking consistency. Since frame-wise techniques can generate high-quality stylized images [GEB16; JAF16], it has become a common practice to employ neural networks for reducing temporal inconsistencies as a post-processing step [LXC20; BTS*15; LHW*18; LXOC22]. Nonetheless, it is important to note that style transfer techniques primarily rely on measuring perceptual distance [ZIE*18], which can result in imperfect stylization due to a lack of deep comprehension at the semantic level. Some recent works have shown improved consistency, but often within specific domains, such as portrait videos[FJS*17; YJLL22]. For local video editing, layer-atlas-based methods [KOWD21; BOF*22] present a promising approach by allowing video editing on a flattened texture map and generating results through color-wise mapping.

More recent approaches have explored video editing using diffusion models [RBL*21]. These models offer stronger priors for editing using text. *e.g*., Gen1 [ECA*23a] trains a conditional model for depth and text-guided video generation, allowing on-the-fly appearance editing of generated images. Several methods [WGW*22; QCZ*23; KMT*23; LZL*23] leverage pre-trained text-to-image diffusion models for zero-shot or one-shot video editing. Although current methods have shown promising results for image pixels, there is still a lack of techniques for generating vector video sketches.

3. Methods

Our approach primarily aims to generate the sketch representation of the objects in video. The main focus is usually on the foreground, presenting diverse motion trajectories. Despite our attempts at background sketching, the generated results often appear disorderly due to the inclusion of a sketchy background. The sketchy video is represented through multiple vector strokes. Each stroke is represented as the four points Bézier Curves, where we focus on maintaining both semantic accuracy and temporal consistency. To achieve, this goal, as shown in Figure 2, we first decompose a video into a 2D representation using layer atlas [KOWD21] (Sec. 3.1). Then, we introduce a new framework to optimize the location of these points in Sec. 3.2. Finally, we give some potential applications in Sec. 3.3.

3.1. Preliminary: Video Decomposition via Layer Atlas

Unlike previous video synthesis and editing tasks [WLZ*18; ECA*23a], our method focuses on optimizing the positions of discrete points within curves to ensure consistent behavior across frames. Therefore, using image-based frame deflickering methods [LXC20; LRZC23] directly becomes challenging. To address this, we leverage the trained representation of previous consistent video editing method, *i.e*., Neural Layered Atlas (NLA [KOWD21]), to maintain point consistency.

As shown in the left part of **Figure 2**, the Neural Layered Atlas (NLA) treats the video as a spatial-temporal 3D volume, where each 3D coordinate within the video is mapped onto the foreground (or background) 2D UV-maps through a Multi-Layer Perceptron (MLP). Additionally, an extra MLP is employed to assign color values to a given 2D UV location. To formalize this, let's consider a video pixel as $p = (x, y, t)$, where x and y are the coordinates of the pixel within the t-th frame. NLA generates the 2D UV maps for foreground and background individually as shown in **Figure 2**. It uses M_f and M_b to map 3D coordinate p to a 2D location in foreground UV-map as (u_f, v_f), and in background UV-map as (u_b, v_b) separately. The MLP M_α indicates the ownership of the

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

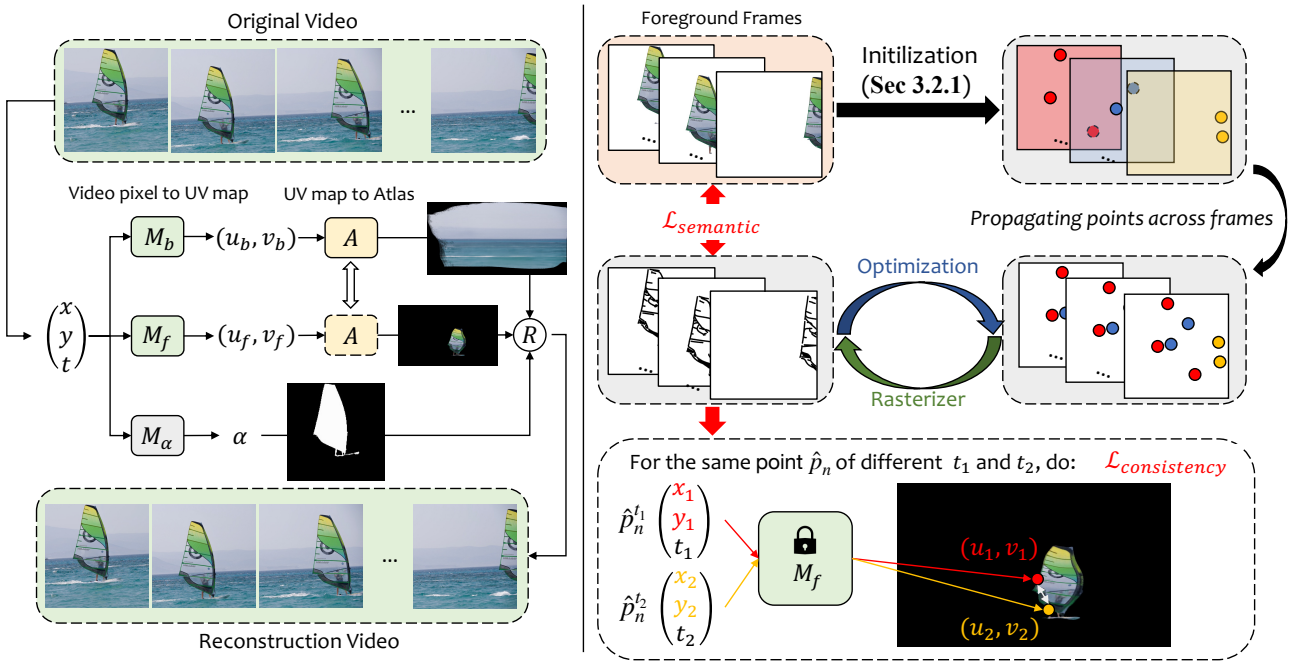

Figure 2: ***Pipeline.*** *Firstly, we train a layer atlas to decompose the video into the trained layer atlas. Then, we optimize the location of the generated frames via the proposed novel initialization methods and consistency loss.*

points (foreground or background) according to the motion prior and the pre-defined mask losses. Subsequently, a shared MLP A is trained to map the (u,v) to the corresponding color in RGB space, ensuring that the same pixel in the real world should have the same color.

This comprehensive process facilitates the video reconstruction. After training, the mapping MLPs M_f and M_b indicate the correspondence between points in the video and specific points within the holistic UV-map, aligning with our requirements for curve mapping consistency. Our method adheres to the original video decomposition approach of the layer atlas. For more specific information regarding training and loss functions, detailed insights can be found in [KOWD21].

3.2. Differentiable Optimization for Video Sketch

In this section, we give the details on how to generate a sketch video in our framework. Formally, given a real video $\{\mathcal{I}_1,...,\mathcal{I}_T\}$ contains T frames, we define the sketching video $\{\mathcal{S}_1,...,\mathcal{S}_T\}$ as a set of N strokes $\mathcal{S}_t=\{s_1,...,s_N\}$ for each frame. Each stroke s_i is defined as the two dimensional Bézier curves, where each curve is built via 4 control points $s_i=\{(x_i,y_i)^k\}_{k=1}^4$. We empirically use the same index i of stroke s_i across different sketches $\mathcal{S}_t$ to represent the same curve across frames and only optimize their positions. All other curve-related attributes are following the previous image sketching method [VPB*22]. We then use a differentiable rasterizer $\mathcal{R}$ from DiffVG [LLMR20] to transform the control points and attributes of Bézier Curves into SVG to represent the final sketch video. So the loss functions $\mathcal{L}$ can be optimized between the real video and the sketches representation. The overall process can be represented as:

$$\arg\min_{(X,Y)} \sum_{t=1}^{T} \mathcal{L}(\mathcal{R}(\mathcal{S}_t),\mathcal{I}_t), \tag{1}$$

where (X,Y) are all the coordinates of the control points $\bigcup_{i=1}^{N}\bigcup_{k=1}^{4}(x_i,y_i)^k$ in the whole sketch video.

Subsequently, we first introduce our method on how to obtain the initial stroke settings on video (Sec. 3.2.1). Then, we elaborate on the particulars of rendering points into curves and optimize the entire video (sec. 3.2.2).

3.2.1. Strokes Initialization

The objective function of our abstraction is highly non-convex [VPB*22] since the optimization loss is based on rendered views in pixel differences [LLMR20]. Thus, attempting to optimize the location directly from random initialization often leads to the process getting trapped in local extrema, both in the image and video abstraction (as in **Figure 3**). To overcome this challenge, CLIPasso [VPB*22] employs a saliency-guided initialization process, where strokes are sampled from the probability map. This map is generated by multiplying the saliency map via CLIP [RKH*21] feature with the image's edge map extracted using XDoG [WKO12], and then subjecting the result to softmax normalization. We argue this kind of initialization is hard to work in processing video since each frame has a variant attention map, where these varying initial points increase the difficulties of the point optimizations. Below, we give our solution step by step.

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

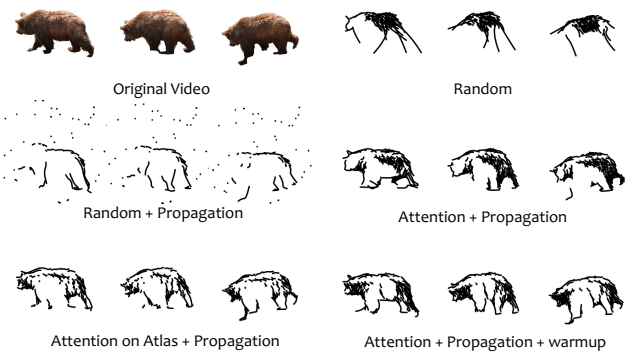

Figure 3: *The influence of different initialization methods.*

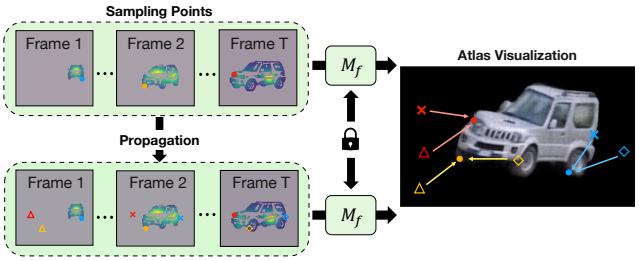

Figure 4: ***Point Position Warmup.*** *The initialized points are warmed up by optimizing the propagated points in different frames* { △, △, ×, ×, ◇, ◇ } *closer to the sampling points* { ●, ●, ● } *on the atlas (by mapping MLP M_f).*

Point Sampling. We first need to generate sparse key points across frames, which are then used to represent the entire video. These points should have higher semantic correlations. As shown in **Figure 4**, to build a sketch video with T frames, where each frame contains N strokes, we first apply salience-guided initialization process [VPB*22] to sample N candidate points in each frame based on their normalized attention maps individually. Then, in order to obtain the initial points on video, we consider temporal sampling, where we resample (randomly pick) N' points (N' equals N) from the set of all $N \times T$ candidate points to build a cross-frame point set $P = \{p_1, ..., p_N\}$. Here, p_n are 3D coordinates (x_n, y_n, t_n), where n indicates the index of control points. Considering both individual frames and the entire video, the more important parts are more likely to be sampled across the entire video.

Point Propagation. We assume that the points at the same index should be initialized at a similar position in different frames, which is very helpful in avoiding local minima (as in **Figure 3**). To achieve this, as shown in **Figure 4**, we paste sampled points into each frame with the corresponding index. So each point index has the same spatial 2D initialization as $\hat{p}_n^t = (x_n, y_n, t)$ in all video frames (the same x, y with different t). After the process of point propagation, the total number of points is represented as $N \times T$ and is denoted as $\mathcal{P}$.

Position Warmup. Since there can be offsets between the sampled points in different frames (*e.g.*, the same color markings in various frames in **Figure 4**), we employ an optimization-based approach to alleviate the differences between the propagated locations and the previously sampled positions on the UV maps by adjusting the propagated points only. This alignment is achieved through the use of the pretrained atlas mapping network M_f to optimize the control points across frames on the atlas:

$$\mathcal{L}_{warmup} = \sum_{t=1}^{T} \sum_{n=1}^{N} ||M_f(\hat{p}_n^t) - M_f(p_n)||_1, \tag{2}$$

In this situation, t represents the frame index, n signifies the control point index, p denotes the sampled points, and $\hat{p}$ signifies the propagated points.

In practice, we warm up 300 iterations. This strategy helps us to find the most suitable initialization points. As shown in **Figure 3**, random initialization (or solely attention-based methods [VPB*22] with similar results) yields poor performance due to the lack of correspondence between the same index points across the video. While propagating the points across different frames helps establish correspondence, it can still lead to a loss of focus on the object. Different attention strategies yield varying results, with frame-wise attention outperforming the attention map based on the atlas. This is because the atlas tend to exhibit distortions compared to the more realistic frames. Ultimately, the performance is further enhanced and becomes more accurate with the inclusion of a warm-up phase in the entire initialization process.

Curve Width Initialization. We also initiate the curve widths, considering that generated videos can involve significant motions and varying scales. We only consider the situation where there is only a single foreground object in the processed video. Since more accurate stroke thickness enhances the representation of contours, instead of solely relying on the mask's scale, we leverage the object's distortion to estimate the appropriate scale. In detail, as depicted in **Figure 5**, we randomly pair points in each frame to calculate the differences between points (here, we use all warmed-up points to create pairs). By computing the γ root of the weighted pair-wise differences on frame t (with γ equals to 3 as a measure of distortion in the 3D world), we derive a scale factor sc_t and curve width w_t:

$$sc_t = \sqrt[\gamma]{\frac{\sum_{n=1}^{N} D_n^t}{\max\limits_{1 \le i \le T} (\sum_{n=1}^{N} D_n^i)}}, \tag{3}$$

$$w_t = sc_t \times w, \tag{4}$$

where w is the default width. Consequently, more distant objects are depicted with finer lines to capture intricate details and scale, while closer objects are outlined with thicker lines. This strategy enhances the quality of abstraction, particularly in scenarios involving substantial object motion.

3.2.2. Curves Optimization

Following a well-executed stroke initialization for the video, our aim is to optimize the curves based on the positions of their points. This optimization needs to ensure that the resulting sketch video maintains not only semantic similarities with the input video but also consistency across frames.

Firstly, we follow image-based sketching methods [VPB*22] to

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

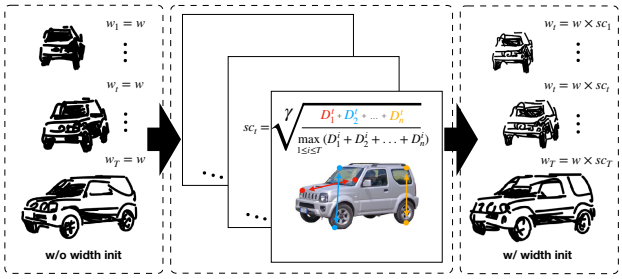

Figure 5: ***Curve Width Initialization.*** *The width w_t of strokes in t-th frame is scaled by the variable sc_t, with initialized width w. The variable γ controls the contrast ratio of scale (default value is 3).*

use the semantic-aware loss $\mathcal{L}_{semantic}$ to measure the differences between the generated points and the original video. $\mathcal{L}_{semantic}$ is based on the differences in multi-scale perception features extracted from the pre-trained CLIP image encoder [RKH*21]. Since CLIP is trained on a larger-scale dataset to align the text and image information through contrastive learning, the semantic information can be well-aligned. Formally, for the control points $\mathcal{S} = \{s_1, ..., s_N\}$ where $s_i = \{(x_i, y_i)^k\}_{k=1}^4$ on each vector frame, we optimize the positions of the control points based on the disparity between the generated vector sketch and the actual input image $\mathcal{I}$ via the l-th layer of pretrained CLIP Φ:

$$\mathcal{L}_{semantic} = \sum_{l\in[3,5,9]} \sum_{t=1}^{T} ||\Phi_l(\mathcal{R}(\mathcal{S}_t)) - \Phi_l(\mathcal{I}_t)||_1, \tag{5}$$

Here, T represents the total number of frames, and $\mathcal{R}$ is the differentiable rasterizer as introduced above.

We also investigate the impact of variants of the CLIP encoders, specifically, the ViT [DBK*20] based and ResNet101-based models [HZRS16]. The ResNet101-based CLIP model exhibits better performance with more local structures, while the ViT-based methods focus more on global features. Consequently, we default to using the ResNet101-based CLIP model, as depicted in **Figure 6**.

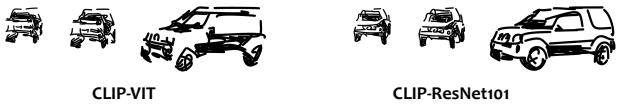

Figure 6: *The differences in the choice of different semantic losses.*

Besides, for our video sketching, we design a novel consistency loss to measure the consistency between the generated sketch frames The most naive approach is to maintain control over the corresponding offsets to guarantee temporal consistency via optical flow [TD20; WLZ*18]. While optical flows are typically dense, they might suffer from potential errors, *e.g.*, forward-backward consistency, and cumulative errors. Therefore, we employ the trained atlas network to obtain a panoramic view of the video. This allows each point in every frame of the video to be associated with a consistent global position on the atlas. By ensuring that related points and lines have consistent global positions, we can maintain the continuity of the video. Specifically, we utilize the pre-trained mapping network M_f from the layer atlas, as illustrated in **Figure 2**. For each control point $\hat{p}_n^{t_1}$ and its neighboring point $\hat{p}_n^{t_2}$ with the same index n, they are expected to be spatially closer in the atlas. Here, t_1 and t_2 are defined for sequentially neighboring frames. Then $\mathcal{L}_{consistency}$ can be written as:

$$\mathcal{L}_{consistency} = \sum_{\hat{p}'\in\mathcal{N}_t(\hat{p})} \sum_{\hat{p}\in\mathcal{P}} ||M_f(\hat{p}) - M_f(\hat{p}')||_1, \tag{6}$$

where the $\mathcal{N}_t$ means temporal neighborhood (neighboring frames). After optimization, point $\hat{p}_n$, with the same index n across frames, aim to have a similar location in the 2D UV-map and keeps consistency over frames.

Overall, the loss function can be written as:

$$\mathcal{L} = \omega_1 \mathcal{L}_{semantic} + \omega_2 \mathcal{L}_{consistency}, \tag{7}$$

where ω_1 and ω_2 are used to control the extent of semantics and consistency between sketches, respectively. We experimentally set $\omega_1 = 200.0$ and $\omega_2 = 3.0$.

3.3. Applications

SVG Editing. Because the generated video is in the form of SVG, the size can be edited losslessly. Our method also has the ability to change the colors of all lines or fill specific lines within the video to create richer details in the visual content. As shown in **Figure 1**, we can resize the canvas, paint all objects in orange, and fill the enclosed lines.

Video Editing and Doodling. Our method can also be employed for video editing and doodle creation. As shown in **Figure 1**, After generating the SVGs, we initially remove the foreground object from the video using inpainting techniques [SLM*21]. Following the restoration of the original video (background filling), we can seamlessly blend the sketch into the foreground. Additional video results are presented in the supplementary video.

4. Experiments

Our method is evaluated on DAVIS dataset [PPC*17], which provides foreground annotations for each frame. We also evaluate the proposed method on some self-collected datasets, utilizing the video matting method [LYSS21] to generate the foreground videos. In each case, we only used the first 50 frames in the video to generate the results for a fair comparison. Subsequently, we decompose the video into an atlas. For the optimization of sketching videos, we utilize the Adam optimizer with a learning rate of 1.0, following the approach of CLIPasso [VPB*22]. On average, optimizing a video sketch takes approximately 29 minutes, consuming around 19.5GB on a single NVIDIA GeForce RTX 3090 GPU.

4.1. Compare with State-of-the-Art Methods

As no previous video sketching methods exist, we employ some image sketching and edge detection techniques for comparison. For image abstraction, in particular, we compare with frame-by-frame CLIPasso [VPB*22]. Regarding edge detection, we assess against traditional edge detection techniques, namely Canny [Can86], as

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

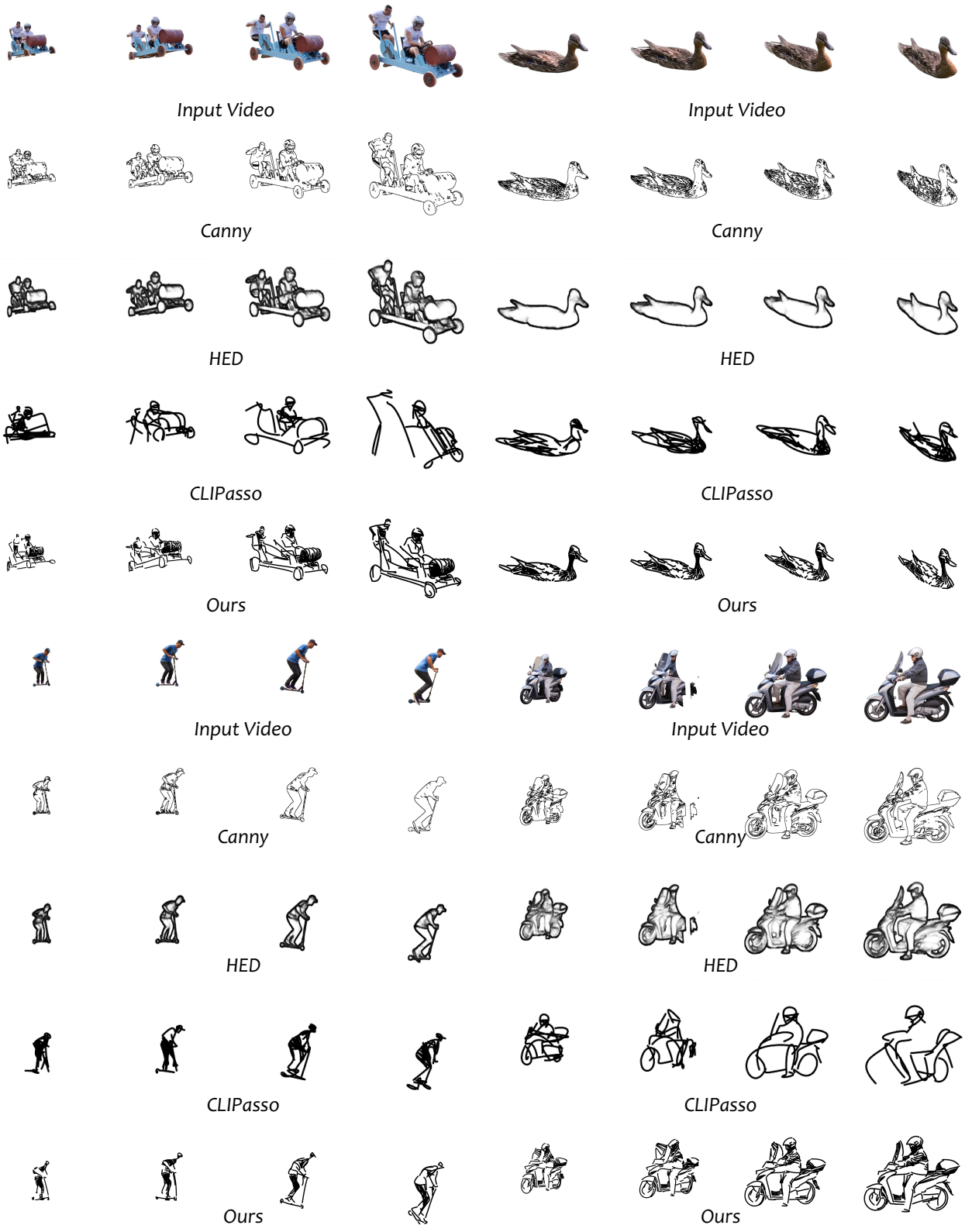

Figure 7: *Comparisons with our methods and the states-of-the-art methods frame-wise methods,* i.e.*, frame-wise CLIPasso [VPB*22] and edge detection methods (canny [Can86] and HED [XT15]) on different frames of the videos.*

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

	Temporal ↑	Semantic ↑	Abstraction ↑
Canny	0.975± 0.0038	0.807± 0.0095	0.298± 0.0071
HED	0.982± 0.0036	0.816± 0.0072	0.283± 0.0067
CLIPasso	0.949± 0.0033	0.801± 0.0074	0.299± 0.0069
Ours	**0.983**± 0.0029	**0.821**± 0.0086	**0.300**± 0.0070

Table 1: *The mean values of clip scores (± denotes standard error).*

	Temporal ↑	Abstraction ↑	Overall ↑
Canny	2.65± 0.055	2.58± 0.060	2.51± 0.062
HED	**2.67**± 0.063	2.30± 0.064	2.60± 0.062
CLIPasso	2.13± 0.069	2.41± 0.065	2.31± 0.064
Ours	2.58± 0.061	**2.74**± 0.062	**2.71**± 0.061

Table 2: *User study opinions on average (± denotes standard error).*

well as the deep learning-based method, HED [XT15]. As demonstrated in **Figure 7**, in comparison to CLIPasso, our approach exhibits improved temporal consistency while effectively preserving semantic information. In comparison with the edge detection methods, our proposed technique showcases superior semantic-aware abstraction. Further insights and video comparisons can be found in the supplementary video.

We conduct the quantitative evaluation using the trained CLIP model [RKH*21] as previous methods [ECA*23b; HHF*21]. The results are presented in Table 1. Specifically, we measure the temporal consistency of videos by calculating the cosine similarity between clip features of consecutive frames. This metric is also employed to assess frame-wise similarity with the original foreground video, indicating semantic coherence. In the case of Abstraction, we compute the cosine similarity with the text "a freehand drawing of <video name>" to gauge the likelihood of a freehand representation.

Recognizing the absence of universally accepted standard metrics for numerically evaluating sketching, we conduct user studies to assess the performance of the generated video. In detail, for each of the 12 clips, we provide generated videos created by different methods for comparison. We then invite 25 individuals to rank the resulting videos in terms of semantic alignment (Abstraction), temporal consistency (Temporal), and overall quality (Overall), respectively. To help users understand abstraction, we describe it as resembling a freehand drawing and aligning well with its meanings. Temporal consistency is explained as the stability of the image over time. In terms of overall quality, we define it as users' general preference. The order of the generated videos is shuffled, and participants are required to rank the results of four different

	Semantic ↑	Abstraction ↑
Random	0.712± 0.014	0.256± 0.005
Rand+Propag	0.722± 0.018	0.279± 0.012
Attn+Propag	0.788± 0.027	0.283± 0.009
Atlas+Propagn+Warmup	0.819± 0.015	0.298± 0.010
Attn+Propag+Warmup	**0.825**± 0.012	**0.300**± 0.012

Table 3: *Ablation on Point Initialization (± denotes standard error).*

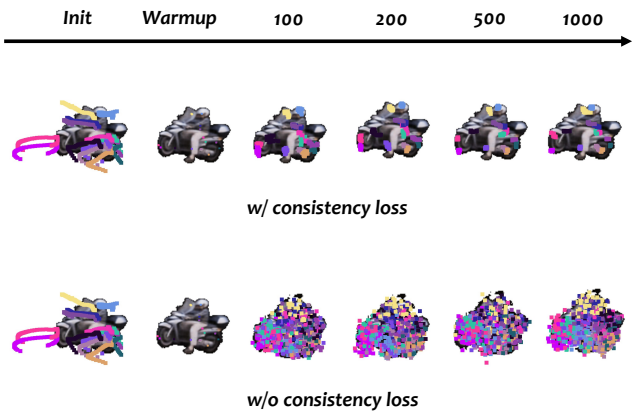

Figure 8: ***Point Visualization***. *To further support the effectiveness of the proposed consistency loss on the atlas, we visualize the location of the same curves (same color) on different frames during optimization. Kindly take note that the warmup points have been gathered, and it is recommended to view them with zoom in.*

methods from best to worst (with 4 being the best and 1 being the worst). The final score is determined based on these rankings. As presented in Table 2, our method receives more favorable feedback from users when compared to the baseline CLIPasso [VPB*22], demonstrating improvements across temporal consistency, semantic abstraction, and overall quality. Moreover, it's noteworthy that our approach achieves enhanced semantic results with scores on par with those of edge detection methods in terms of temporal consistency.

4.2. Ablation Studies

Point Initialization. We emphasize the importance of point initialization in sketch generation, as depicted in **Figure 3**. We propose a point initialization method that considers both local and global features. Here, we utilize CLIP scores [HHF*21], as explained in Section 4.1, to quantitatively evaluate different strategies. Specifically, 'random' involves randomly choosing points in each frame, 'propagation' maintains the same 2D initial location across frames, 'attention' samples points using the attention maps of each frame, 'atlas' uses the attention map from atlas, and 'warmup' keeps corresponding points across frames at the same location in the atlas. The evaluation results align with the visual outcomes in **Figure 3**, indicating that our approach outperforms other strategies in the Table 3.

Optimization Visualization. We introduce a novel consistency loss based on the trained atlas network. Here, we visualize the points on the global UV map (atlas) to gain a clearer understanding. As illustrated in **Figure 8**, points with the same index across multiple frames are represented in the same color. During the optimization process, the control points from different frames are appropriately positioned at the same points after the warming-up phase. Subsequently, we optimize these points using both the consistency loss and the semantic loss to ensure performance in terms of semantic alignment and coherence.

Consistency Weights. The novel consistency loss maintains tem-

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

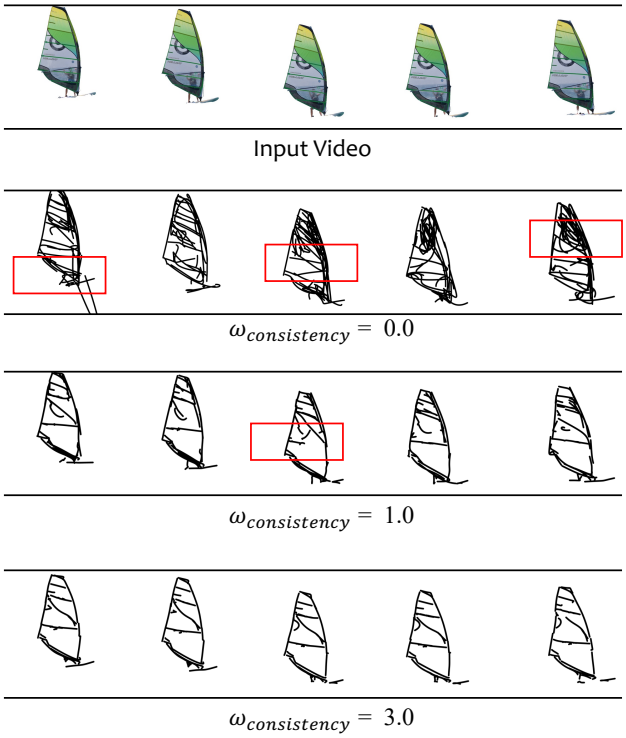

Figure 9: *The importance of the proposed consistency loss ($\omega_{consistency}$ means ω_2 in eq.7).*

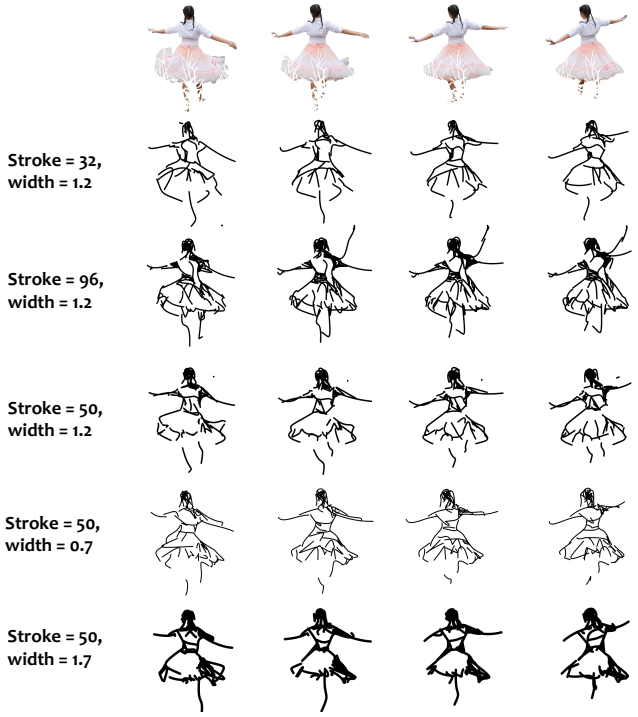

Figure 10: *The ablations on the stroke width and numbers.*

poral domain consistency through the trained atlas representation. In our ablation study, we investigate this loss using different values. As depicted in **Figure 9**, when the consistency loss is excluded (*i.e.*, $\omega_{consistency} = 0$), the generated sketch exhibits distinct representations across frames. As we increment this parameter, the optimized results demonstrate enhanced stability.

Strokes Number and Width. We also show the vector sketch frame using different numbers and widths of default strokes. Increasing the number of paths, as illustrated in **Figure 10**, leads to the generation of sketches that capture more intricate details. For instance, the details of the dress and its movements become more prominent and well-defined, which reduces the level of abstraction. Likewise, adjusting the stroke width yields similar effects. Decreasing the stroke width during optimization emphasizes finer details within the strokes rather than the overall structure, resulting in a less abstract representation.

4.3. Limitation

Our method excels at generating coherent and semantically rich videos, encompassing rigid objects, as exemplified by the snowboard sketch video in **Figure 1**. Nevertheless, our approach encounters challenges with complicated motion sequences. The limitations of the trained layer atlas [KOWD21] constrain the quality of the sketches. For instance, it faces challenges when representing the motion of non-rigid bodies. Additionally, during cases of self-occlusion, the generated abstract sketches may contain errors, often appearing as improper turns (as observed in the supplementary video). Furthermore, the generated sketches may exhibit some texture artifacts when the video undergoes significant motion or involves complex foreground elements. As depicted in **Figure 11**, our proposed method exhibits artifacts when the male face shakes violently, and the atlas struggles to accurately represent the woman in the image. While some of these issues can be mitigated by segmenting the video into smaller sections and employing more layers, the accuracy of segmentation and the computational cost associated with multi-layer approaches remain challenges.

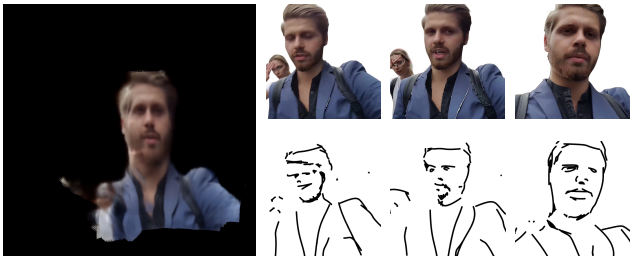

Figure 11: ***Limitations****. When the motion is large and the object is complex, the generated atlas may not be accurate, causing temporal inconsistency.*

4.4. Conclusion

We present an optimization-based approach for generating sketch videos that maintain both semantic and temporal consistency. Our method facilitates the creation of sketching videos with the appropriate level of abstraction. Essentially, it includes a novel initializa-

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

tion technique for acquiring well-initialized points and a distinctive consistency loss derived from self-supervised video decomposition. These innovations empower us to craft sketch videos using simple Bézier curves. Since the resulting videos are composed using Scalable Vector Graphics, our proposed methods offer versatile applications in video editing and doodling, accommodating various sizes while preserving intricate details.

Acknowledgement

This work was supported in part by the Science and Technology Development Fund, Macau SAR, under Grant 0087/2020/A2 and Grant 0141/2023/RIA2.

References

[Aga02] AGARWALA, ASEEM. "Snaketoonz: A semi-automatic approach to creating cel animation from video". *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. 2002, 139–ff 3.

[AHSS04] AGARWALA, ASEEM, HERTZMANN, AARON, SALESIN, DAVID H, and SEITZ, STEVEN M. "Keyframe-based tracking for rotoscoping and animation". *ACM Transactions on Graphics (ToG)* 23.3 (2004), 584–591 3.

[AMFM10] ARBELÁEZ, P, MAIRE, M, FOWLKES, C, and MALIK, J. "Contour Detection and Hierarchical Image Segmentation". en-US. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Aug. 2010), 898–916. DOI: 10.1109/tpami.2010.161. URL: http://dx.doi.org/10.1109/tpami.2010.161 2.

[BBM*16] BEN-ZVI, N., BENTO, J., MAHLER, MOSHE, et al. "Line-Drawing Video Stylization". *Computer Graphics Forum* 35 (2016). URL: https://api.semanticscholar.org/CorpusID:38616448 2.

[BOF*22] BAR-TAL, OMER, OFRI-AMAR, DOLEV, FRIDMAN, RAFAIL, et al. "Text2live: Text-driven layered image and video editing". *European Conference on Computer Vision*. Springer. 2022, 707–723 3.

[BSM*13] BERGER, ITAMAR, SHAMIR, ARIEL, MAHLER, MOSHE, et al. "Style and Abstraction in Portrait Sketching". *ACM Trans. Graph.* 32.4 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2461912.2461964. URL: https://doi.org/10.1145/2461912.2461964 2.

[BTS*15] BONNEEL, NICOLAS, TOMPKIN, JAMES, SUNKAVALLI, KALYAN, et al. "Blind Video Temporal Consistency". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2015)* 34.6 (2015) 3.

[Can86] CANNY, JOHN. "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), 679–698. DOI: 10.1109/TPAMI.1986.4767851 2, 6, 7.

[CDAT20] CARLIER, ALEXANDRE, DANELLJAN, MARTIN, ALAHI, ALEXANDRE, and TIMOFTE, RADU. "Deepsvg: A hierarchical generative network for vector graphics animation". *Advances in Neural Information Processing Systems* 33 (2020), 16351–16361 2.

[DBK*20] DOSOVITSKIY, ALEXEY, BEYER, LUCAS, KOLESNIKOV, ALEXANDER, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". *arXiv preprint arXiv:2010.11929* (2020) 6.

[DYH*20] DAS, AYAN, YANG, YONGXIN, HOSPEDALES, TIMOTHY, et al. "Bézier sketch: A generative model for scalable vector sketches". *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer. 2020, 632–647 2.

[ECA*23a] ESSER, PATRICK, CHIU, JOHNATHAN, ATIGHEHCHIAN, PARMIDA, et al. "Structure and content-guided video synthesis with diffusion models". *arXiv preprint arXiv:2302.03011* (2023) 3.

[ECA*23b] ESSER, PATRICK, CHIU, JOHNATHAN, ATIGHEHCHIAN, PARMIDA, et al. "Structure and content-guided video synthesis with diffusion models". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 7346–7356 8.

[FJL*16] FIŠER, JAKUB, JAMRIŠKA, ONDŘEJ, LUKÁČ, MICHAL, et al. "Stylit: illumination-guided example-based stylization of 3d renderings". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 1–11 3.

[FJS*17] FIŠER, JAKUB, JAMRIŠKA, ONDŘEJ, SIMONS, DAVID, et al. "Example-based synthesis of stylized facial animations". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1–11 3.

[FSW22] FRANS, KEVIN, SOROS, LISA, and WITKOWSKI, OLAF. "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders". *Advances in Neural Information Processing Systems* 35 (2022), 5207–5218 3.

[GEB16] GATYS, LEON A, ECKER, ALEXANDER S, and BETHGE, MATTHIAS. "Image style transfer using convolutional neural networks". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 2414–2423 3.

[GSH*19] GRYADITSKAYA, YULIA, SYPESTEYN, MARK, HOFTIJZER, JAN WILLEM, et al. "OpenSketch: a richly-annotated dataset of product design sketches". en. *ACM Transactions on Graphics* (Nov. 2019), 1–16. DOI: 10.1145/3355089.3356533. URL: http://dx.doi.org/10.1145/3355089.3356533 2.

[HE17] HA, DAVID and ECK, DOUGLAS. *A Neural Representation of Sketch Drawings*. en-US. Apr. 2017 2.

[HHF*21] HESSEL, JACK, HOLTZMAN, ARI, FORBES, MAXWELL, et al. "Clipscore: A reference-free evaluation metric for image captioning". *arXiv preprint arXiv:2104.08718* (2021) 8.

[HZRS16] HE, KAIMING, ZHANG, XIANGYU, REN, SHAOQING, and SUN, JIAN. "Deep residual learning for image recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778 6.

[IVH*23] ILUZ, SHIR, VINKER, YAEL, HERTZ, AMIR, et al. "Word-as-image for semantic typography". *arXiv preprint arXiv:2303.01818* (2023) 3.

[JAF16] JOHNSON, JUSTIN, ALAHI, ALEXANDRE, and FEI-FEI, LI. "Perceptual losses for real-time style transfer and super-resolution". *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, 694–711 3.

[JST*19] JAMRIŠKA, ONDŘEJ, SOCHOROVÁ, ŠÁRKA, TEXLER, ONDŘEJ, et al. "Stylizing Video by Example". *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3323006. URL: https://doi.org/10.1145/3306346.3323006 3.

[KMT*23] KHACHATRYAN, LEVON, MOVSISYAN, ANDRANIK, TADEVOSYAN, VAHRAM, et al. "Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators". *arXiv preprint arXiv:2303.13439* (2023) 3.

[KOWD21] KASTEN, YONI, OFRI, DOLEV, WANG, OLIVER, and DEKEL, TALI. "Layered neural atlases for consistent video editing". *ACM Transactions on Graphics (TOG)* 40.6 (2021), 1–12 2–4, 9.

[KP20] KAMPELMUHLER, MORITZ and PINZ, AXEL. "Synthesizing human-like sketches from natural images using a conditional convolutional decoder". *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, 3203–3211 2.

[LHW*18] LAI, WEI-SHENG, HUANG, JIA-BIN, WANG, OLIVER, et al. "Learning blind video temporal consistency". *Proceedings of the European conference on computer vision (ECCV)*. 2018, 170–185 3.

[LLM*19] LI, MENGTIAN, LIN, ZHE, MECH, RADOMIR, et al. "Photosketching: Inferring contour drawings from images". *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, 1403–1412 2.

© 2024 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

[LLMR20] LI, TZU-MAO, LUKÁČ, MICHAL, MICHAËL, GHARBI, and RAGAN-KELLEY, JONATHAN. "Differentiable Vector Graphics Rasterization for Editing and Learning". *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39.6 (2020), 193:1–193:15 2–4.

[LRZC23] LEI, CHENYANG, REN, XUANCHI, ZHANG, ZHAOXIANG, and CHEN, QIFENG. "Blind Video Deflickering by Neural Filtering with a Flawed Atlas". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023 2, 3.

[LXC20] LEI, CHENYANG, XING, YAZHOU, and CHEN, QIFENG. "Blind video temporal consistency via deep video prior". *Advances in Neural Information Processing Systems* 33 (2020), 1083–1093 2, 3.

[LXOC22] LEI, CHENYANG, XING, YAZHOU, OUYANG, HAO, and CHEN, QIFENG. "Deep video prior for video consistency and propagation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), 356–371 3.

[LYSS21] LIN, SHANCHUAN, YANG, LINJIE, SALEEMI, IMRAN, and SENGUPTA, SOUMYADIP. "Robust High-Resolution Video Matting with Temporal Guidance". (2021). arXiv: 2108.11515 [cs.CV] 6.

[LZL*23] LIU, SHAOTENG, ZHANG, YUECHEN, LI, WENBO, et al. "Video-P2P: Video Editing with Cross-attention Control". *arXiv:2303.04761* (2023) 3.

[NSC*11] NORIS, GIOACCHINO, SỲKORA, DANIEL, COROS, STELIAN, et al. "Temporal noise control for sketchy animation". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. 2011, 93–98 2.

[OH11] O'DONOVAN, PETER and HERTZMANN, AARON. "Anipaint: Interactive painterly animation from video". *IEEE transactions on visualization and computer graphics* 18.3 (2011), 475–487 3.

[PPC*17] PONT-TUSET, JORDI, PERAZZI, FEDERICO, CAELLES, SERGI, et al. "The 2017 davis challenge on video object segmentation". *arXiv preprint arXiv:1704.00675* (2017) 6.

[QCZ*23] QI, CHENYANG, CUN, XIAODONG, ZHANG, YONG, et al. "FateZero: Fusing Attentions for Zero-shot Text-based Video Editing". *arXiv:2303.09535* (2023) 3.

[RBL*21] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV] 3.

[RGLM21] REDDY, PRADYUMNA, GHARBI, MICHAEL, LUKAC, MICHAL, and MITRA, NILOY J. "Im2vec: Synthesizing vector graphics without vector supervision". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 7342–7351 2.

[RKH*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. "Learning transferable visual models from natural language supervision". *International conference on machine learning*. PMLR. 2021, 8748–8763 2–4, 6, 8.

[Sab97] SABISTON, BOB. *Commercial systems: Bob Sabiston's Rotoshop*. Brian Whited's system for Disney's Paperman. 1997. URL: https://www.flatblackfilms.com/Flat_Black_Films/Rotoshop.html 3.

[SBF*18] SU, QINGKUN, BAI, XUE, FU, HONGBO, et al. "Live sketch: Video-driven dynamic deformation of static drawings". *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, 1–12 3.

[SLM*21] SUVOROV, ROMAN, LOGACHEVA, ELIZAVETA, MASHIKHIN, ANTON, et al. "Resolution-robust Large Mask Inpainting with Fourier Convolutions". *arXiv preprint arXiv:2109.07161* (2021) 6.

[SZL*23] SMITH, HARRISON JESSE, ZHENG, QINGYUAN, LI, YIFEI, et al. "A Method for Animating Children's Drawings of the Human Figure". *ACM Transactions on Graphics* 42.3 (2023), 1–15 3.

[TD20] TEED, ZACHARY and DENG, JIA. "Raft: Recurrent all-pairs field transforms for optical flow". *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, 402–419 6.

[TH22] TIAN, YINGTAO and HA, DAVID. "Modern Evolution Strategies for Creativity: Fitting Concrete Images and Abstract Concepts". en-US. *Artificial Intelligence in Music, Sound, Art and Design,Lecture Notes in Computer Science*. Apr. 2022, 275–291. DOI: 10.1007/978-3-031-03789-4_18. URL: http://dx.doi.org/10.1007/978-3-031-03789-4_18 3.

[VACS22] VINKER, YAEL, ALALUF, YUVAL, COHEN-OR, DANIEL, and SHAMIR, ARIEL. "CLIPascene: Scene Sketching with Different Types and Levels of Abstraction". en-US. (Nov. 2022) 2, 3.

[VPB*22] VINKER, YAEL, PAJOUHESHGAR, EHSAN, BO, JESSICAY., et al. "CLIPasso: Semantically-Aware Object Sketching". en-US. (2022) 2–8.

[WGW*22] WU, JAY ZHANGJIE, GE, YIXIAO, WANG, XINTAO, et al. "Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation". *arXiv preprint arXiv:2212.11565* (2022) 3.

[WKO12] WINNEMÖLLER, HOLGER, KYPRIANIDIS, JAN ERIC, and OLSEN, SVEN C. "XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization". *Computers & Graphics* 36.6 (2012), 740–753 2, 4.

[WLZ*18] WANG, TING-CHUN, LIU, MING-YU, ZHU, JUN-YAN, et al. "Video-to-Video Synthesis". *Advances in Neural Information Processing Systems (NeurIPS)*. 2018 3, 6.

[WOG06] WINNEMÖLLER, HOLGER, OLSEN, SVEN C, and GOOCH, BRUCE. "Real-time video abstraction". *ACM Transactions On Graphics (TOG)* 25.3 (2006), 1221–1226 3.

[WSML23] WU, RONGHUAN, SU, WANCHAO, MA, KEDE, and LIAO, JING. "IconShop: Text-Based Vector Icon Synthesis with Autoregressive Transformers". *arXiv preprint arXiv:2304.14400* (2023) 3.

[XHY*20] XU, PENG, HOSPEDALES, TIMOTHYM., YIN, QIYUE, et al. "Deep Learning for Free-Hand Sketch: A Survey and A Toolbox". en-US. *arXiv: Computer Vision and Pattern Recognition* (Jan. 2020) 2.

[XT15] XIE, SAINING and TU, ZHUOWEN. "Holistically-nested edge detection". *Proceedings of the IEEE international conference on computer vision*. 2015, 1395–1403 2, 7, 8.

[YBN*23] YU, EMILIE, BLACKBURN-MATZEN, KEVIN, NGUYEN, CUONG, et al. "VideoDoodles: Hand-Drawn Animations on Videos with Scene-Aware Canvases". *ACM Transactions on Graphics (TOG)* 42 (2023), 1–12. URL: https://api.semanticscholar.org/CorpusID:259359061 2.

[YJLL22] YANG, SHUAI, JIANG, LIMING, LIU, ZIWEI, and LOY, CHEN CHANGE. "VToonify: Controllable High-Resolution Portrait Video Style Transfer". *ACM Transactions on Graphics (TOG)* 41.6 (2022), 1–15. DOI: 10.1145/3550454.3555437 3.

[YYF*22] YI, RAN, YE, ZIPENG, FAN, RUOYU, et al. "Animating portrait line drawings from a single face photo and a speech signal". *ACM SIGGRAPH 2022 Conference Proceedings*. 2022, 1–8 3.

[ZIE*18] ZHANG, RICHARD, ISOLA, PHILLIP, EFROS, ALEXEI A, et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". *CVPR*. 2018 3.

© 2024 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.