VectorTalker: SVG Talking Face Generation with Progressive Vectorisation

Hao Hu[1] Xuan Wang[2] Jingxiang Sun[3] Yanbo Fan[2] Yu Guo[1] Caigui Jiang[1]

[1]Xi'an Jiaotong University [2]Ant Group [3]Tsinghua University

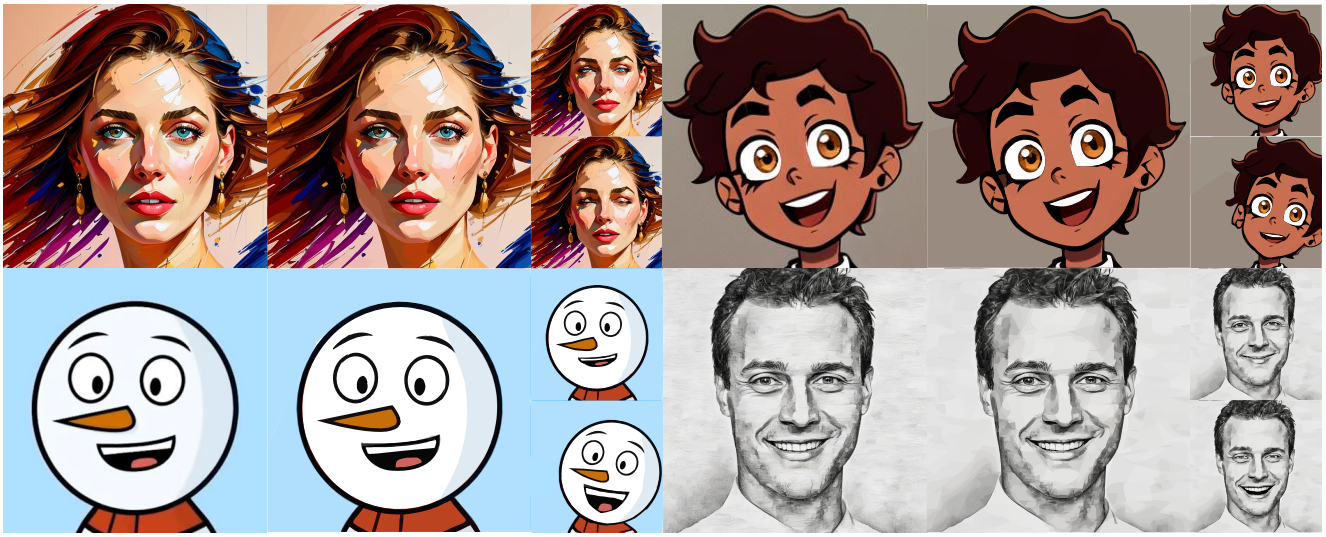

Figure 1. Image vectorization and SVG animation results. We can faithfully reconstruct SVG for different styles of raster portraits and provide vivid animation effects.

Abstract

High-fidelity and efficient audio-driven talking head generation has been a key research topic in computer graphics and computer vision. In this work, we study vector image based audio-driven talking head generation. Compared with directly animating the raster image that most widely used in existing works, vector image enjoys its excellent scalability being used for many applications. There are two main challenges for vector image based talking head generation: the high-quality vector image reconstruction w.r.t. the source portrait image and the vivid animation w.r.t. the audio signal. To address these, we propose a novel scalable vector graphic reconstruction and animation method, dubbed VectorTalker. Specifically, for the high-fidelity reconstruction, VectorTalker hierarchically reconstructs the vector image in a coarse-to-fine manner. For the vivid audio-driven facial animation, we propose to use facial landmarks as intermediate motion representation and propose an efficient landmark-driven vector image deformation module. Our approach can handle various styles of portrait images within a unified framework, including Japanese manga, cartoon, and photorealistic images. We conduct extensive quantitative and qualitative evaluations and the experimental results demonstrate the superiority of VectorTalker in both vector graphic reconstruction and audio-driven animation.

1. Introduction

We study one-shot audio-driven talking head generation [1, 3–5, 15–17, 19, 22, 23, 30], which aims to animate a single portrait image with speech audio. In recent years, the applications of audio-driven facial animation has become ubiquitous in various fields, such as digital human creation, video conferences, and game industry, etc. For these applications, the high visual quality of the animated faces and the ability of scalability are very important. State-of-the-art

realizations conduct facial animation in the input raster image space, for example, by learning a warping field on the raster image according to the audio signal. However, because of the complex structure of the facial area, learning a reasonable warping field for each pixel or latent feature from speech remains challenging and the animated results suffer from distortions and blurs. Besides, the visual quality of the facial animation is restricted by the training resolutions. For instance, the scaled-up operation of the animated raster images will lead to blur or distortion artifacts.

In this work, we study the high-fidelity facial animation in the context of vector graphcis. Unlike raster images that are composed of individual pixels, vector images are composed of mathematical primitives such as lines, curves, and geometric shapes in a resolution-independent fashion. That is, the vector images can be scaled up or down without loss of visual quality. Such scalability property makes vector images ideal for various output media and resolution sizes, which is highly appealing to talking head applications. Besides, vector images possess the characteristic of editability. The primitives of vector images are defined by mathematical equations that are easy to edit, modify, and adjust, without compromising image quality. This makes it possible for high-quality facial animations. Inspired by these promising properties, we study vector graphics based talking head animation. Given a source raster image and an audio clip, we propose to first reconstruct the vector image *w.r.t.* the source image, and then perform facial animation in the vector image space. To the best of our knowledge, we are the first to explore audio-driven facial animation in the vector image space. There are two main challenges: 1) the high-quality vector image reconstruction and 2) the vivid facial animation according to the input audio information.

We propose a novel scalable vector graphic reconstruction and animation method, dubbed *VectorTalker*, that addresses the above challenges. Firstly, given a raster image, we construct the vector image using path primitives composed of L segments of cubic Bezier curves. The shape and color of each primitive are optimized based on the reconstruction error *w.r.t.* the raster image. For a high-fidelity reconstruction, we propose a novel progressive vectorization algorithm. Specifically, we perform l_0 regularized image smoothing on the input raster image with N-1 levels of smooth strengths which yields smoothed images as targets and then perform the differentiable vectorization progressively. In each level we add paths initialized as circles under guidance of the reconstruction error between the optimized paths and the smoothed image from last level. To facilitate subsequent SVG animation, we implement a semantic hierarchical design to reconstruct vector images of the background layer, foreground layer and local layer respectively. Secondly, to enable vivid audio-driven animation, we use facial landmarks as intermediate motion representation and perform Delaunay triangulation on the vector image based on facial landmarks. We then predict the landmark deformation according to the audio information and deform each path primitives of the vector image accordingly. As the Bezier curve may crosses multiple triangulations, a simple unified transformation may leads to distortion results, We propose to split the curves at the intersection points of the triangulation for more accurate animation. We conduct extensive experiments by considering various categories of facial portraits and several state-of-the-art methods. Experimental results demonstrate the effectiveness of VectorTalker and the necessity of the proposed coarse-to-fine vectorization and vivid animation designs.

The contributions of this paper can be summarized as follows:

- We propose the first method to consider vector graphics based one-shot audio-driven talking head generation.
- We propose a novel progressive vectorization algorithm that obtains superior vector image reconstruction.
- We propose an efficient audio-driven module for vivid vector-image animation.
- We conduct extensive experiments and demonstrate the excellent performance of the proposed method.

2. Related Work

Image Vectorization. Unlike raster images composed of pixels, vector graphics are composed of mathematical descriptions of geometric shapes. Most traditional vectorization methods [8, 13, 24] begin by segmenting the images into patches and then fit Bezier curves at the boundaries. To achieve smoother gradient colors, diffusion curves [11, 25] and gradient meshes [7, 18] are used to describe the colors. In recent years, The boom of deep learning has promoted research on differentiable rendering of vector graphics so that raster-based algorithms can be used for vector generation. DiffVG [9] applies anti-aliasing to smooth vector graphics scene discontinuities and makes it differentiable. Im2Vec [12] trains an encoder-decoder architecture from a raster dataset to predict a set of ordered closed vector paths. LIVE [10] initializes component-wise path and optimizes the vector graph in a layer-wise manner and attemps to maintain the topological relationship of SVG, but it is only available for simple images and hard to handle complex textures. [2] work uses parameterization to describe facial attributes and transfer raster image to vector avatar by learning the mapping between both modalities, but it can only generate a single-style result.

Audio-driven Portrait Talking. Most previous [1, 3–5, 14–17, 19–23, 28–31] works mainly focus on raster portrait talking. [19] train a RNN to learn mapping from audio features to mouth movements based on pre-built 3D facial models. AD-NeRF [4] trains two neural radiation

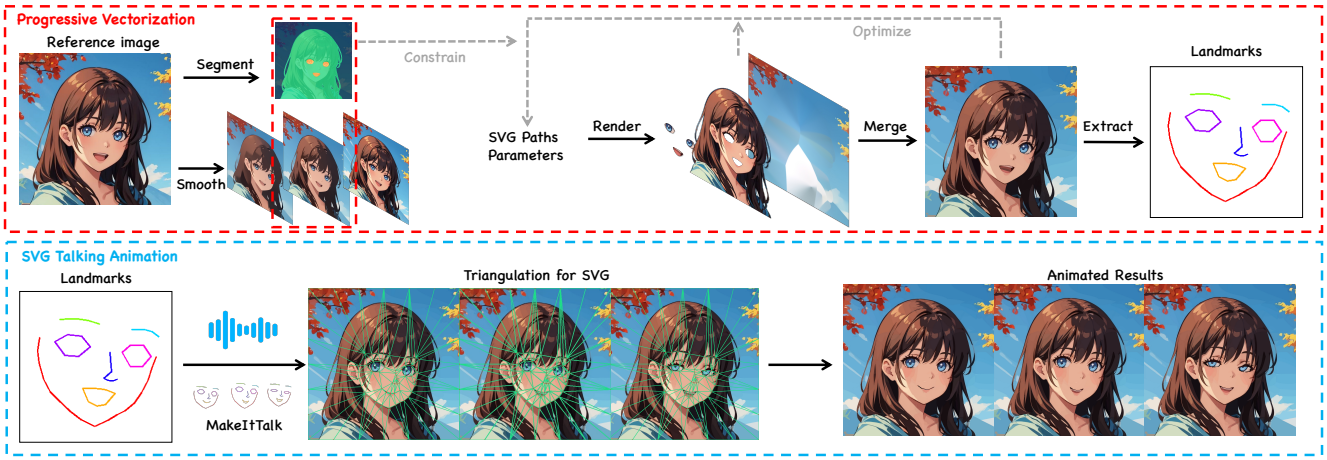

Figure 2. Pipeline of our method. Given an input raster portrait image, our method first segment the image to get semantic mask (back, front, and local) and smooth the image to obtain different levels of smoothed representations as target image. Then we perform the differentiable vectorization to reconstruct three SVG layers constrained by semantic mask progressively and merge them to get final SVG result. For SVG talking animation, we extract landmarks and predict new ones from an audio clip to warp SVG paths by affine transformation.

Figure 3. Image smoothing results. From left to right are the original image, light smoothed and heavy smoothed.

fields by features extracted from audio to render more detailed results. SadTalker [28] proposed ExpNet and PoseVAE to predict 3D motion coefficients from audio, which are used as intermediate representations to generate videos. MakeItTalk [31] uses the facial landmark as the intermediate representation and maps pixels across frames via triangulation to make non-photorealistic portrait talk. To the best of our knowledge, there has been no prior work attempting to animate SVG portrait to talk. We are the first to apply talking generation methods used for raster to vector images.

Method	MSE↓	LPIPS↓	PSNR↑	SSIM↑
DiffVG	0.00213	0.290	27.799	0.857
LIVE	0.00194	0.269	28.291	0.864
Ours	**0.00131**	**0.258**	**29.835**	**0.886**

Table 1. Quantitative evaluation using MSE, LPIPS, PNSR and SSIM on benchmark with different styles of portraits. Our method outperforms in all metrics.

3. Method

We aim to solve the problem of one-shot talking portrait generation in vector graphics which consist of multiple parametric paths as shown in Fig. 2. In this section, we expatriate the details of our proposed method, VectorTalker. The contents are organized as follows. First, the pipeline of reconstructing the SVG image from input raster images is described in Sec. 3.1. Then we present how to animate the reconstructed SVG portraits in Sec. 3.2.

Notations. Let's start by defining the notations and concepts used in vector graphics and our approach. An SVG image is made up of graphic primitives or paths. In this method, we use a specific path $\mathbf{P}_i \in \mathcal{P}$ containing a series of third-order Bezier curves $\mathbf{p}_{i,j}$ which are closed. Each curve consists of four control points $\mathbf{c}^*_{i,j}$. Then we denote the SVG image as a path set $\mathcal{P}$, the raster image as $\mathbf{I}_*$, and the differentiable rasterization as the function $rast(\cdot)$. In optimization, all the paths are initialized as circles with a defined radius r. We represent the stack of smoothing levels as $\mathcal{S} = \{\mathbf{S}^l | l = 1, ..., N\}$. L_f, L_m and L_b are used to refer to the foreground, middle and background layers obtained by the foreground mask $\mathbf{M}_{fore}$ and the local mask $\mathbf{M}_{local}$ in the semantic layering. To create the animation, we extract a few facial key points, denoted by $\mathbf{k}_m$, from the input portrait. We then perform triangulation on these points to produce multiple triangles, represented by F_t. Using the offsets of the corresponding key points, we derive an affine transformation M_t that is used to animate the paths in the image plane.

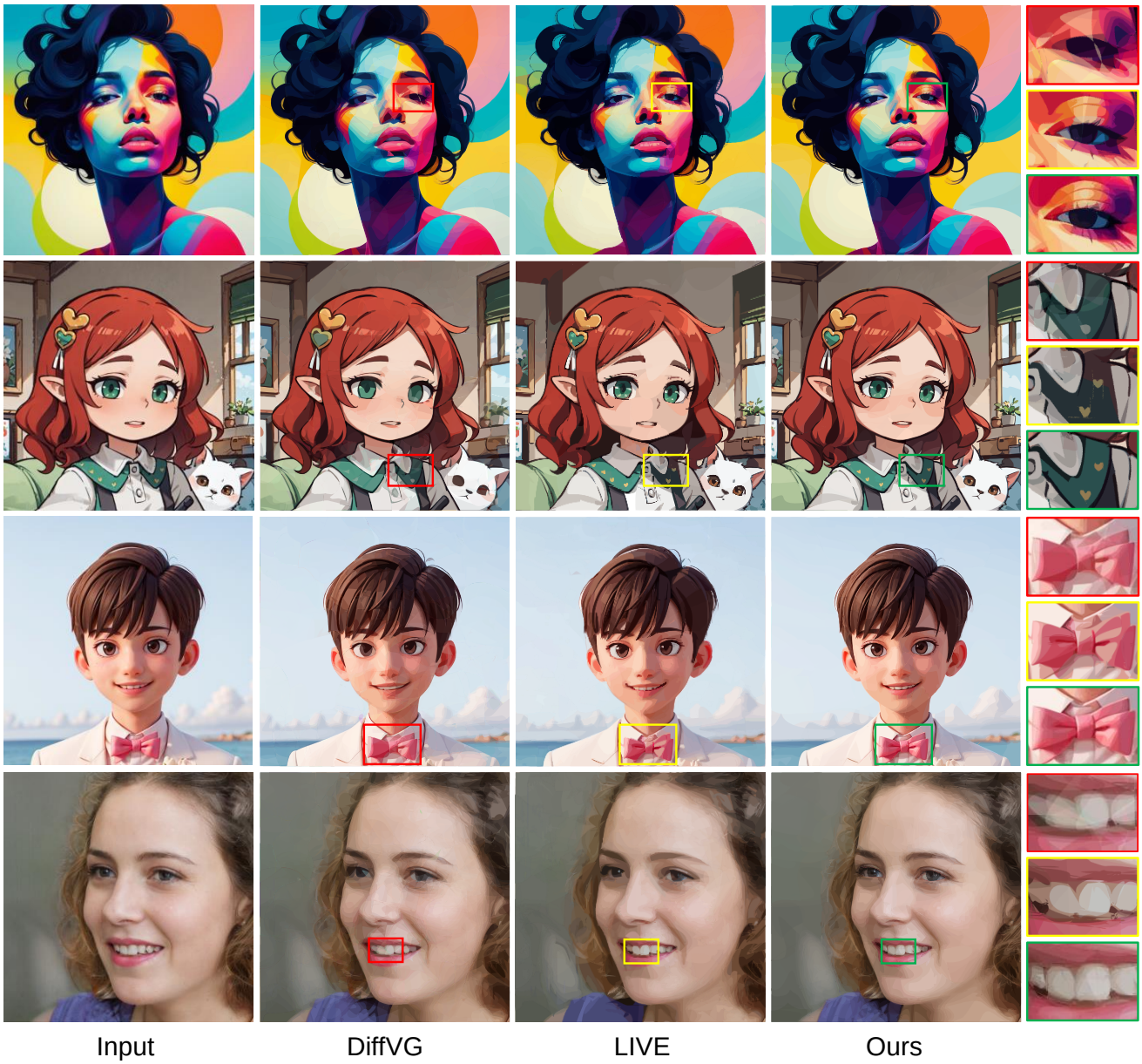

Figure 4. Qualitative results on SVG reconstruction. We compare our progressive vectorization algorithm with DiffVG and LIVE. The experiments illustrate that our method produces better results. We highlight the differences in the boxes on the right, and readers can zoom in for a clearer view.

3.1. Progressive Vectorization

Coarse-to-fine reconstruction. The first step of VectorTalker is to faithfully reconstruct the SVG image given an input raster portrait. SVG can be regarded as a parametric abstraction of the raster image. Therefore, a good vectorization of the raster image should capture most image structures having as few as possible paths. For a set of the optimizable paths, $\mathcal{P}$, random initialization in differentiable rendering usually leads to early convergence to tiny image structures and harms fidelity.

Our approach is to use the progressive vectorization algorithm, which employs the coarse-to-fine strategy. The first step is to create a stack of smoothed images, called $\{\mathbf{S}^1\}$, using N-level smoothing. We do this by applying the l_0 regularized image smoothing to the input raster image as shown in Fig. 3. The smoothing strength is gradually increased for each level of the stack. For more information about the image smoothing algorithm, please see [26].

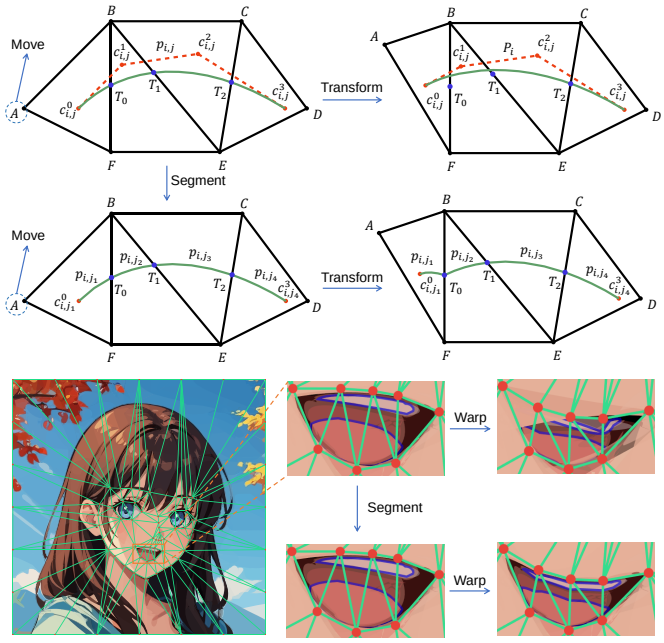

Figure 5. Animation on SVG. The illustration of SVG animation and curve segmentation are shown. The abstraction illustration is shown above, and the qualitative display is shown below. If the paths are warped without curve segmentation, the result is unreasonable(lower) because a curve may span across multiple triangles and will be simultaneously affected by four affine transformations(upper). curve segmentation can maintain the shape of the curves and make paths be warped correctly. The abstracted illustration shows that the Bézier curve $\mathbf{p}_{i,j}$ and its four control points $\mathbf{c}^*_{i,j}, * = 0, 1, 2, 3$ are scattered in 4 triangles. The affine transformations corresponding to triangles ΔAFB, ΔBFE, ΔBEC and ΔCED are M_0, M_1, M_2 and M_3. The entire deformed curve $\mathbf{p}_{i,j}$ will be simultaneously affected by four affine transformations, which is unfavorable for the SVG talking animation. Although we only move point A, the shape of the entire curve changes even if some parts are not within ΔAFB. This issue can be effectively solve with the curve segmentaion by splitting $\mathbf{p}_{i,j}$ into $\mathbf{p}_{i,j_1}$, $\mathbf{p}_{i,j_2}$, $\mathbf{p}_{i,j_3}$ and $\mathbf{p}_{i,j_4}$.

By using l_0 regularized image smoothing, we can produce piece-wise constant images that depict the abstraction of the raster image in different degrees. This allows us to capture details in the different levels of the image.

Given the $\mathcal{S}$, we perform the differentiable vectorization $rast(\cdot)$ progressively. With the decreasing of the smoothing level, we recursively initialize additional paths with a larger radius whose sampling positions are guided by the current error map and add them into the optimization variable stack. Then we optimize the set of paths $\mathcal{P}$ to fit the raster image in the current smoothing level by the MSE loss: $MSE(\mathbf{I}^l_s - rast(\mathcal{P}))$. Finally, the process above is repeated progressively across the levels of smoothing. Fig. 4 show that our progressive vectorization significantly outperforms the baseline methods in image fidelity.

Semantic Layering. To facilitate subsequent SVG animation, we implement a semantic hierarchical design. Specifically, we leverage the off-the-shelf segmentation model, e.g. SAM [6], to extract the foreground mask $\mathbf{M}_{fore}$ and the local mask $\mathbf{M}_{local}$. The set of paths is separated into foreground, middle, and background layers using masks. Then, the sets of paths are merged semantically from back to front.

3.2. Animating SVG Portraits

The aim of our work is to enable natural talking portraits by adjusting the position of the face landmarks. Similar to MakeItTalk [31], but, instead of the raster image, we have to perform the animations on SVG images. After extracting the facial key points $\mathbf{k}_m$, Delaunay triangulation is then performed to divide the image into triangle patches based on the original landmarks. Although the coordinates of the triangle vertices change, the corresponding landmarks subscripts of the vertices remain fixed. We use facial landmarks as an intermediate representation between audio and visual animation, which transforms the subsequent animation process into the triangle transformation process. For each triangle, an affine transformation can be calculated based on the offsets and the original vertex coordinates. As long as the landmark topology remains unchanged, the texture on each triangle transfers across frames. In this paper, we adapt this approach, commonly used for raster images, into the case of vector graphics.

For the animation process of vector graphics, we use an off-the-shelf detector [27] to predict the facial landmarks in the original image and perform Delaunay triangulation, following MakeItTalk [31], and then we predict audio-driven sequence of landmarks, each of them determines the facial expression changes in a new frame. However, unlike raster images, SVG is composed of many paths of different shapes and colors and path is composed of several Bézier curves. Changing the shape of the paths requires changing the curves but simply performing affine transformation on the curve control points within each triangle may result in the path shape suffers unwanted distortion. The reason is that a Bézier curve may span across multiple triangles so that the control points are located within different triangles which often causes unreasonable motions. To address this issue, our solution is to segment all Bézier curves to ensure that each complete curve is inside only one triangle to avoid impact from other vertices. Specifically, we first calculate the intersections of each Bézier curve with all the line segments of the triangulation.

Determining intersections only requires solving a cubic equation. By leveraging the properties of cubic Bézier curves, we can easily split the curve at the intersection points, ensuring that the shape of the path remains un-

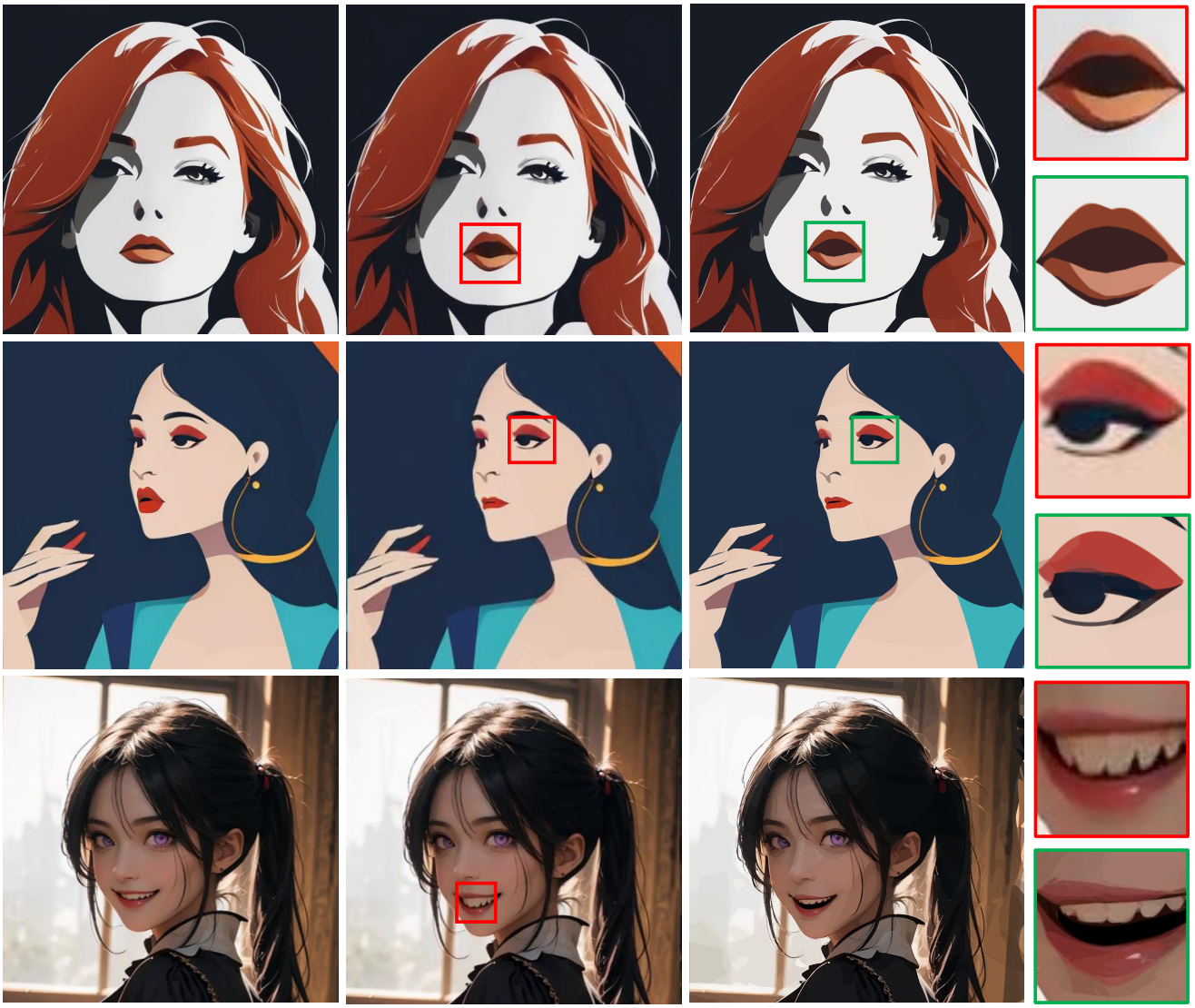

Figure 6. Qualitative comparison with MakeItTalk []. The experiments illustrate that our results remain clear and sharp. Benefiting from the implementation of semantic layering in vectorization, the teeth perform reasonable motions.

changed before and after the segmentation. When all control points of a curve are located within the same triangle, the entire curve undergoes the only affine transformation. As shown in the upper picture of Fig. 5, if $\mathbf{p}_{i,j}$ is divided into 4 Bézier curves at intersections T_0, T_1 and T_2 and then we change the position of point A, only parts within ΔAFB will be changed and other parts will remain fixed. Although deformation after segmentation may destroy the smoothness of the curve, it is indispensable for subsequent animation. The lower picture of Fig. 5 shows the effect of curve segmentation on the SVG mouth region. If we directly warp paths without curves segmentation, the result is strange and messy. In contrast, segmenting a curve can maintain the shape of the path and make paths be warped correctly.

After completing the above steps, we can start driving SVG animation. However, as landmarks coordinates change, if we simply apply the same operation to all paths teeth and eyes may suffer unreasonable deformations and the background will also be distorted, such as the teeth becoming very large, the eyes being squeezed and the background Keep shaking. Thanks to the semantic layering we implement during reconstruction, our method can handle these problems well. Specifically, We can only warp the

Figure 7. Our results of SVG animation. Our method makes SVG talk vividly. It allows the control of lip motion, eye blink and head poses.

path in the foreground and local layers and keep the paths in the background layer fixed. For the eyes and mouth, we reselect landmarks for triangulation, so that the eyes move with the eye sockets and the mouth moves with the lower jaw. When blinking, the eyes can be naturally covered by the upper foreground layer without being squeezed and deformed and the teeth will not stretch to completely cover the entire mouth when lip opens.

4. Experiments

In this section, we present experimental results on vectorization and SVG animation. We tested portraits of various styles, including watercolor, painting, manga, and cartoon avatars. Our system supports images of any resolution.

4.1. Implementation Details

In order to convert SVG images to raster images, we use a differentiable renderer called DiffVG, and then optimize the path parameters by employing the Adam optimizer. The learning rates for point and color are set to 1 and 0.01, respectively. By default, each path consists of eight segments of third-order Bézier curves. Five levels are exploited to construct the stacke of smoothed images. We adopted 500 paths in total across all the smoothing levels.

All parameters are optimized for 200 iterations in each smoothing level, and 400 iterations in the stage fitting the finest image. For the SVG animation, we use audio-predicted landmarks to drive the SVG portrait to talk. All new frame vector images are deformed from the reconstructed SVG, and the number and order of paths will not change. Additionally, we added eye blinks and head tilts to make the animation more vivid.

4.2. SVG Reconstruction

In this section, we created a benchmark containing 20 raster portraits with different styles and resolutions. We then evaluated the SVG Reconstruction of VectorTalker quantitatively and qualitatively, comparing the results with DiffVG and LIVE. DiffVG initializes all paths at once, while LIVE gradually adds new paths, similar to our method. For all experiments, we set the same total number of paths, curve segments, and iterations. As shown in Fig. 4, our

method achieved better reconstruction results on various styles of portrait images. Additionally, our method was able to capture complex details that the other methods could not. Even for portraits of real people, we were able to reconstruct an expressive SVG. We highlighted the differences in the boxes on the right, and readers can zoom in for a clearer view.In this section, we created a benchmark containing 20 raster portraits with different styles and resolutions. We then evaluated the SVG Reconstruction of VectorTalker quantitatively and qualitatively, comparing the results with DiffVG [9] and LIVE [10]. DiffVG initializes all paths at once, while LIVE gradually adds new paths, similar to our method. For all experiments, we set the same total number of paths, curve segments, and iterations. Both qualitative Fig. 4 and quantitative Tab. 1 demonstrate that our method achieved better reconstruction results on various styles of portrait images by capturing complex details.

4.3. SVG animation

In this section, we present the results of SVG animation. We qualitatively compare ours with MakeItTalk for raster images. Then the vivid control of the SVG portrait is displayed.

We present a comparison of our results with MakeItTalk in Fig. 6. Our approach, which uses vector graphics, offers inherent advantages, allowing animations to remain clear and sharp. In contrast, MakeItTalk warps all pixels of the entire image through affine transformation, causing blurry effects and distortions. Additionally, we implemented controlling of eye blink and head poses in addition to the lip motions to make the talking SVG portrait more vivid, as shown i Fig. 7.

Paths	MSE↓	LPIPS↓	PSNR↑	SSIM↑
100	0.00498	0.350	22.856	0.772
250	0.00234	0.291	26.561	0.841
500	0.00131	0.258	29.835	0.886
750	0.00112	0.222	30.085	0.893
1000	0.00101	0.205	30.814	0.902

Table 2. Ablation on the number of paths. Quantitative evaluation using MSE, LPIPS, PNSR and SSIM on benchmark. More paths will improve reconstructed SVG fidelity and details.

4.4. Ablation Studies

We perform several ablation studies on different factors, the number of the paths, the number of the smoothing levels and the effect of the curve segmentation.

Number of paths. The total number of paths is a non-differentiable hyperparameter. We ablate the vectorization results using different numbers of paths on our benchmark as shown in Tab. 2. We set the paths to 100, 250, 500, 750 and 1000 respectively and keep other parameters the same.

Levels	MSE↓	LPIPS↓	PSNR↑	SSIM↑
1	0.00213	0.290	27.799	0.857
2	0.00163	0.285	28.524	0.872
3	0.00146	0.269	29.267	0.881
4	0.00138	0.262	29.693	0.883
5	0.00131	0.258	29.835	0.886

Table 3. Ablation on the number of smoothing levels. Progressive vectorization helps improve reconstruction quality. Note using one level is equivalent to directly using DiffVG.

Figure 8. Ablation of curves segmentation. The deformed SVG appears messy and distorted without curves segmentation, particularly in the face area where landmarks and triangles are more dense. In contrast, our full method produces accurate results.

Number of Levels. Through progressive vectorization, we apply different numbers of smoothing levels. In Tab. 3, we examine the impact of the number of smoothing levels on our benchmarks, using MSE, LPIPS, PNSR, and SSIM with various styles of portraits.

Curve segmentation. In order to illustrate whether curves should be split at their intersections with triangulation before warping paths, we compare the animation results of segmented and non-segmented curves. As shown in Fig. 8, the curve segmentation helps to preserve reasonable image structures in animation.

5. Conclusion

Our research proposes the VectorTalker, a novel approach for generating one-shot audio-driven talking SVG portraits. Our progressive vectorization algorithm allows us to accurately reconstruct the input raster image in vector graphics. We extract facial key points and use an affine-transformation-based warping system to animate the SVG portrait with audio-driven facial key point offset prediction. Our extensive experiments demonstrate that our progressive vectorization significantly outperforms other baseline methods. Additionally, our method effectively accomplishes the task of talking SVG generation. In the future, we plan to utilize more prior knowledge about humans to achieve more

vivid control, such as hair and emotion.

References

[1] Lilin Cheng, Suzhe Wang, Zhimeng Zhang, Yu Ding, Yixing Zheng, Xin Yu, and Changjie Fan. Write-a-speaker: Text-based emotional and rhythmic talking-head generation. In *AAAI Conference on Artificial Intelligence*, 2021. 1, 2

[2] Manuel Ladron de Guevara, Jose Echevarria, Yijun Li, Yannick Hold-Geoffroy, Cameron Smith, and Daichi Ito. Cross-modal latent space alignment for image to avatar translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 520–529, 2023. 2

[3] Sefik Emre Eskimez, Ross K. Maddox, Chenliang Xu, and Zhiyao Duan. End-to-end generation of talking faces from noisy speech. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1948–1952, 2020. 1, 2

[4] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[5] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. You said that?: Synthesising talking faces from audio. *Int. J. Comput. Vision*, 127(11–12):1767–1779, 2019. 1, 2

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 5

[7] Yu-Kun Lai, Shi-Min Hu, and Ralph R. Martin. Automatic and topology-preserving gradient mesh generation for image vectorization. In *ACM SIGGRAPH 2009 Papers*, New York, NY, USA, 2009. Association for Computing Machinery. 2

[8] Gregory Lecot and Bruno Levy. Ardeco: Automatic region detection and conversion. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, page 349–360, Goslar, DEU, 2006. Eurographics Association. 2

[9] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6), 2020. 2, 8

[10] Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16293–16302, 2022. 2, 8

[11] Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. Diffusion curves: A vector representation for smooth-shaded images. *ACM Trans. Graph.*, 27(3):1–8, 2008. 2

[12] Pradyumna Reddy, Michaël Gharbi, Michal Lukáč, and Niloy J. Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2124–2133, 2021. 2

[13] Peter Selinger. Potrace : a polygon-based tracing algorithm. 2003. 2

[14] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. *First Order Motion Model for Image Animation*. Curran Associates Inc., Red Hook, NY, USA, 2019. 2

[15] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. *First Order Motion Model for Image Animation*. Curran Associates Inc., Red Hook, NY, USA, 2019. 1

[16] Linsen Song, Wayne Wu, Chaoyou Fu, Chen Qian, Chen Change Loy, and Ran He. Everything's talkin': Pareidolia face reenactment, 2021.

[17] Linsen Song, Wayne Wu, Chen Qian, Ran He, and Chen Change Loy. Everybody's talkin': Let me talk as you want. *IEEE Transactions on Information Forensics and Security*, 17:585–598, 2022. 1, 2

[18] Jian Sun, Lin Liang, Fang Wen, and Heung-Yeung Shum. Image vectorization using optimized gradient meshes. *ACM Trans. Graph.*, 26(3):11–es, 2007. 2

[19] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 36(4), 2017. 1, 2

[20] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. In *the 30th International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.

[21] Suzhen Wang, Lincheng Li, Yu Ding, and Xin Yu. One-shot talking face generation from single-speaker audio-visual correlation learning. In *AAAI 2022*, 2022.

[22] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 1

[23] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, page 690–706, Berlin, Heidelberg, 2018. Springer-Verlag. 1, 2

[24] Tian Xia, Binbin Liao, and Yizhou Yu. Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Trans. Graph.*, 28(5):1–10, 2009. 2

[25] Guofu Xie, Xin Sun, Xin Tong, and Derek Nowrouzezahrai. Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph.*, 33(6), 2014. 2

[26] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l0 gradient minimization. *ACM Trans. Graph.*, 30(6):1–12, 2011. 4

[27] Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: Landmark detection and geometric style in portraits. *ACM Trans. Graph.*, 38(4), 2019. 5

[28] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8652–8661, 2023. 2, 3

[29] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3660–3669, 2021.

[30] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4174–4184, 2021. 1

[31] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: Speaker-aware talking-head animation. *ACM Trans. Graph.*, 39(6), 2020. 2, 3, 5

VectorTalker: SVG Talking Face Generation with Progressive Vectorisation

Supplementary Material

A. Algorithm

Algorithm 1 shows the pipeline of VectorTalker, consisting of two stages, progressive vectorization and SVG talking animation. For vectorization, given a reference image, we segment it to get semantic mask and smooth it to obtain different levels of smoothed representations as target images. We progressively add paths and perform the differentiable vectorization to reconstruct three SVG layers and the merged SVG. All results are constrained by current smoothing level target and we optimize the paths by MSE loss. For animation, we extract landmarks to perform triangulation and predict new landmarks from an audio clip. Then we split all the Bezier curves in the SVG at the intersections with the triangulation line segments. Finally, we can warp SVG paths by affine transformation to get animated results.

Algorithm 1 Algorithm of VectorTalker

Input: I; //reference iamge
Output: V; //SVG talking animation
Procedure:
mask=segment(I)
$\{\mathbf{S}^l | l = 1, ..., N\}$ = smooth(I)
para = []; //list of path parameters
errormap = 0;
for i in N **do**
  newpara = init(errormap,n);
  para = concat(para, newpara);
  for j in M **do**
    $\hat{I}$ = render(para);
    L = loss($\hat{I}, S^i, mask$);
    para = update(L, para);
  end for
  errormap = $\left\| S^i - \hat{I} \right\|_2$
end for
$\hat{I}$ = render(para);
lmk = detect($\hat{I}$);
lmks = pred(lmk, audio);
para = segcurves(para, lmk)
V = []
for k in K **do**
  para = warp(para, lmks[k], lmk)
  frame = render(para);
  V = concat(V, frame)
end for

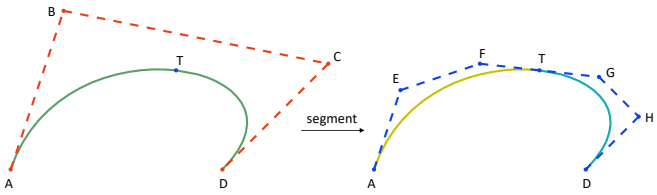

Figure 1. Diagram of curve segmentation. The curve is segmented into two at T. The shape will remain unchanged and the number of control points will increase.

B. Loss Function

To facilitate SVG animation, we implement a semantic hierarchical design during vectorization to fit three layers respectively. Our loss function, including the MSE loss of each layer and the overall MSE loss of merged SVG, is calculated as follows:

$$L_{img} = \lambda_1 L_{back} + \lambda_2 L_{fore} + \lambda_3 L_{local} + \lambda_4 L_{merged}$$

$$L_{back} = \left\| inpainting(S^l * mask_{back}) - \hat{I}_{back} \right\|_2^2$$

$$L_{fore} = \left\| S^l * mask_{fore} - \hat{I}_{fore} \right\|_2^2$$

$$L_{local} = \left\| S^l * mask_{local} - \hat{I}_{local} \right\|_2^2$$

$$L_{merged} = \left\| S^l - \hat{I}_{merged} \right\|_2^2$$

Where L_{back}, L_{fore} and L_{local} are the MSE loss of the background layer, foreground layer, and local layer respectively. S^l represents the target smoothed image in the l-th level, and $\hat{I}$ represents our rendered result. By default, we set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$. It is worth mentioning that we keep the background layer fixed in the svg talking animation process. In order to avoid showing a hollow background when head moving, we inpaint the background layer after segmentation.

C. Curve Segmentation

As shown in Fig. 1, the bezier curve has four control points(A, B, C and D). The point T is on the curve and $T = (1-t)^3 A + 3(1-t)^2 tB + 3(1-t)t^2 C + t^3 D$ where t is the position parameter ranging from 0 to 1. We segment the curve into two parts at T. New control points can be calculated as:

$$(A, E, F, T)^T = M_0 \times (A, B, C, D)^T$$
$$(T, G, H, D)^T = M_1 \times (A, B, C, D)^T$$
$$M_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1-t & t & 0 & 0 \\ (1-t)^2 & 2t(1-t) & t^2 & 0 \\ (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{pmatrix}$$
$$M_1 = \begin{pmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \\ 0 & (1-t)^2 & 2t(1-t) & t^2 \\ 0 & 0 & 1-t & t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix M_0 and M_1 can be solved using the method of undetermined coefficients. To segment the curve into multiple parts, we can perform the above operation recursively.