Supplementary Materials for
Wired Perspectives: Multi-View Wire Art Embraces Generative AI

Zhiyu Qu[1] Lan Yang[2] Honggang Zhang[2] Tao Xiang[1] Kaiyue Pang[1] Yi-Zhe Song[1]

[1]SketchX, CVSSP, University of Surrey [2]Beijing University of Posts and Telecommunications

{z.qu, t.xiang, k.pang, y.song}@surrey.ac.uk {ylan, zhhg}@bupt.edu.cn

https://dreamwireart.github.io

1. Differentiable 3D MVWA rendering proof

Any Bézier curve can be represented as the following format:

$$B(t) = (1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3. \tag{1}$$

Given a plane π, its normal vector N and a point q in the plane, the projection of any point $P(t)$ of curve $B(t)$ on plane π could be represented as $P'(t)$:

$$P'(t) = P(t) - (N \cdot (P(t) - q)) * N, \tag{2}$$

where $\cdot$ denotes the dot product operation.

Our objective is to prove that *the projection of* $B(t)$ *on the plane* π ($B'(t)$) is **equal** to *the curve formed by the projected points of the control points of that curve* ($B''(t) = \{p'_0, p'_1, p'_2, p'_3\}$) on the same plane π, *i.e.*, $B'(t) = B''(t)$.

Based on Eqs. 1 and 2, we can obtain the projection curve $B'(t)$ of $B(t)$ on the plane π:

$$\begin{aligned} B'(t) = &[(1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3] \\ &- \{N \cdot [(1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3 - q]\} * N. \end{aligned} \tag{3}$$

We can also obtain the curve $B''(t)$ formed by the projected points of the control points on the plane π:

$$\begin{aligned} B''(t) = &(1-t)^3[p_0 - (N \cdot (p_0 - q)) * N] + 3(1-t)^2 t[p_1 - (N \cdot (p_1 - q)) * N] + 3(1-t)t^2[p_2 - (N \cdot (p_2 - q)) * N] + t^3[p_3 - (N \cdot (p_3 - q)) * N] \\ = &[(1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3] \\ &- [(1-t)^3(N \cdot (p_0 - q)) * N + 3(1-t)^2 t(N \cdot (p_1 - q)) * N + 3(1-t)t^2(N \cdot (p_2 - q)) * N + t^3(N \cdot (p_3 - q)) * N]. \end{aligned} \tag{4}$$

Comparing Eq. 3 and 4, we find that the red parts are the same. Therefore, we only need to show that the blue part in Eq. 4 equals to the green part in Eq. 3. Since vector dot product satisfies the distributive over vector addition properties, *i.e.*, $\vec{a} \cdot (\vec{b} + \vec{c}) = \vec{a} \cdot \vec{b} + \vec{a} \cdot \vec{c}$, we can deform the blue part of Eq. 4 as follows:

$$\begin{aligned} &(1-t)^3(N \cdot (p_0 - q)) * N + 3(1-t)^2 t(N \cdot (p_1 - q)) * N + 3(1-t)t^2(N \cdot (p_2 - q)) * N + t^3(N \cdot (p_3 - q)) * N \\ =&\{N \cdot [(1-t)^3(p_0 - q) + 3(1-t)^2 t(p_1 - q) + 3(1-t)t^2(p_2 - q) + t^3(p_3 - q)]\} * N \\ =&\{N \cdot [(1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3 - ((1-t)^3 + 3(1-t)^2 t + 3(1-t)t^2 + t^3) * q)]\} * N \\ =&\{N \cdot [(1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3 - q]\} * N \end{aligned} \tag{5}$$

Based on the above proof, we can transform the 3D Bézier curve rendering problem into a 2D rendering problem based on the projection of 3D control points. Given a 3D wire and a viewpoint, we can have a raster sketch with the utilisation of [3] in our 3D to 2D renderer. And more importantly, the whole process is *differentiable*.

2. Bad case analysis

When we use VectorFusion [2] for 2D sketch generation, the results are limited by the diffusion prior. In general, given any text prompt, when the diffusion model generates images well, VectorFusion can also generate great sketches. However, this is not the case for DreamWire, as there are multi-view conflicts to consider. We analyse this problem with the following examples.

Fig. 1 shows the concepts of "apple", "banana" and "flower". However, we can notice that there is a lot of redundancy present in the sample of "apple", *e.g.*, some leaves and a logo of "apple" in the centre. Compared to bananas and flowers, apples have a simpler structure. Therefore, if they are drawn within the same number of strokes, there are only two scenarios: the apples are drawn very intricately, or the bananas and flowers are drawn more abstractly. However, generative diffusion models like those in [5, 6], trained solely on images, lack significant abstraction skills for simple stroke outlines. Consequently, *DreamWire* is better at making complex sketches, *e.g.*, "portraits of three people". This may go against human intuition. In addition, MVWA [1] mentioned that their method produces clearly visible artifacts due to the difficulty in resolving inconsistency from simple contours. Enhancing generative models' abstraction abilities could mitigate these challenges.

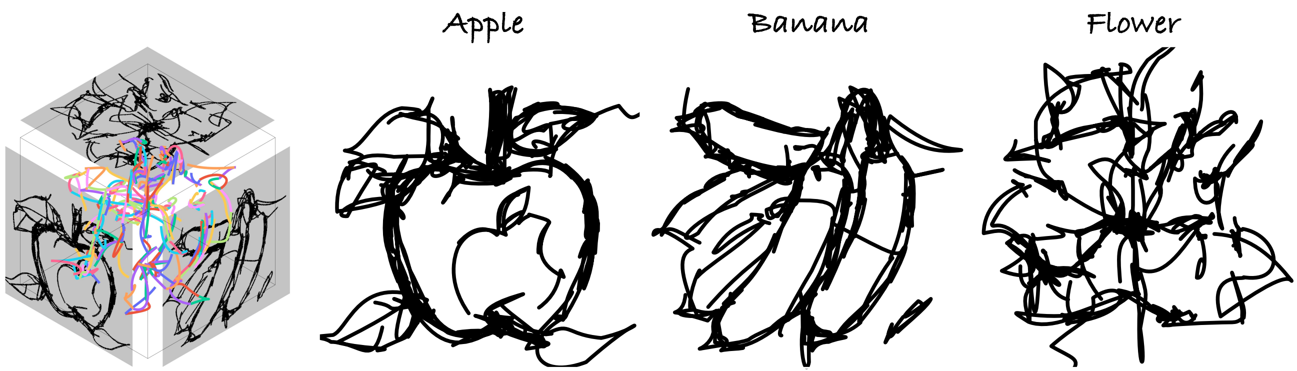

Figure 1. The textual prompts for three viewpoints are "apple", "banana" and "flower", respectively. All prompts possess the following prompt prefix: "a simple drawing of [text]".

Fig. 2 shows the concepts of "Walter White", "Jesse Pinkman" and "Saul Goodman". We can notice that there is a lot of redundancy in the sample of "Walter White". It's common knowledge that Walter White is bald while Jesse Pinkman is not. During our generation process, their heads are positioned at the same height along the z-axis. Therefore, in order to draw Jesse's hair, *DreamWire* has to add some extra strokes to the sideburns of Walter White. The potential for conflicting content of different viewpoints is an issue that users need to consider in future applications.

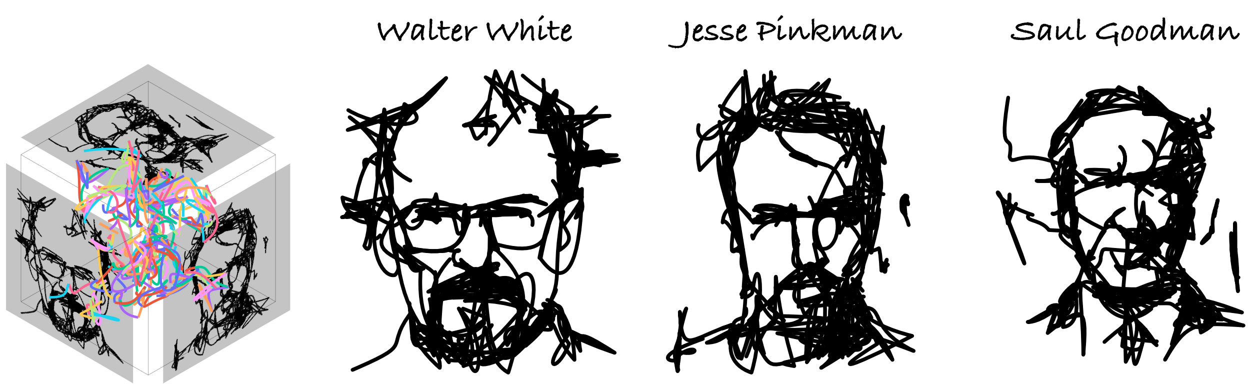

Figure 2. The textual prompts for three viewpoints are "Walter White", "Jesse Pinkman" and "Saul Goodman", respectively. All prompts possess the following prompt prefix: "a head of [text]".

3. Time-consuming analysis

The time required to create a piece of Multi-View Wire Art varies significantly across different approaches: a professional artist typically requires several months. Compared to this, *DreamWire* and the baseline method [1, 4] show a substantial improvement in time. ShadowArt [4] is based on the resize and voxel search of the target image, and it usually takes only 1 minute to complete. However, when there is a conflict between the three input target images (which is a very common situation), it produces a poor visual hull or even crashes. MVWA [1] takes about 2 to 10 hours, depending on the voxel resolution and the complexity of the input images, whereas *DreamWire* significantly reduces this to approximately 30 minutes. Our method utilises a text-to-image generation model to raise the upper limit of creativity with acceptable time consumption compared to rule-based methods. While SDS may not be recognised for its efficiency in AIGC, it emerges as the most efficient method known to us for the task of Multi-View Wire Art creation. In addition, with the widespread use of SDS for 3D generation tasks, the acceleration of SDS has been investigated [7]. We anticipate that the performance of our proposed method will be enhanced with advancements in generative AI.

4. Physical display

In our main paper, we highlight that utilising $\mathcal{L}_{\text{MST}}$ compromises aesthetics. To address this, we employ laser crystal in creating our multi-view wire art, as demonstrated in Fig. 3. Please watch the video to experience the fun of changing perspectives.

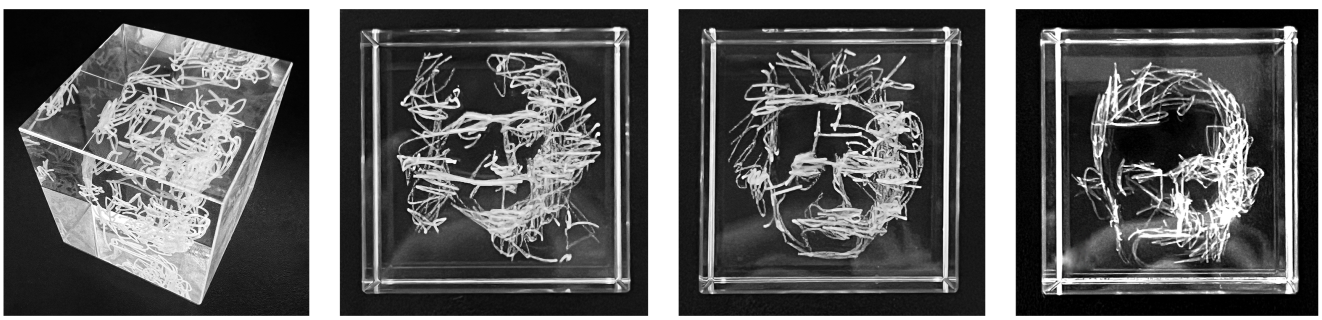

Figure 3. The textual prompts for three viewpoints are "Isaac Newton", "Albert Einstein" and "Alan Turing", respectively. All prompts possess the following prompt prefix: "a head of [text]".

References

[1] Kai-Wen Hsiao, Jia-Bin Huang, and Hung-Kuo Chu. Multi-view wire art. *ACM Transactions on Graphics*, 2018. 2, 3

[2] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *CVPR*, 2023. 2

[3] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics*, 2020. 1

[4] N. J. Mitra and M. Pauly. Shadow art. *ACM Transactions on Graphics*, 2009. 3

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[6] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. In *ACL*, 2020. 2

[7] Linqi Zhou, Andy Shih, Chenlin Meng, and Stefano Ermon. Dreampropeller: Supercharge text-to-3d generation with parallel sampling. *arXiv:2311.17082*, 2023. 3