ViewCraft3D: High-Fidelity and View-Consistent 3D Vector Graphics Synthesis

Chuang Wang [1]* **Haitao Zhou** [1]* **Ling Luo** [2]† **Qian Yu** [1]†
[1]Beihang University [2]Ningbo University
*Equal contribution †Corresponding author

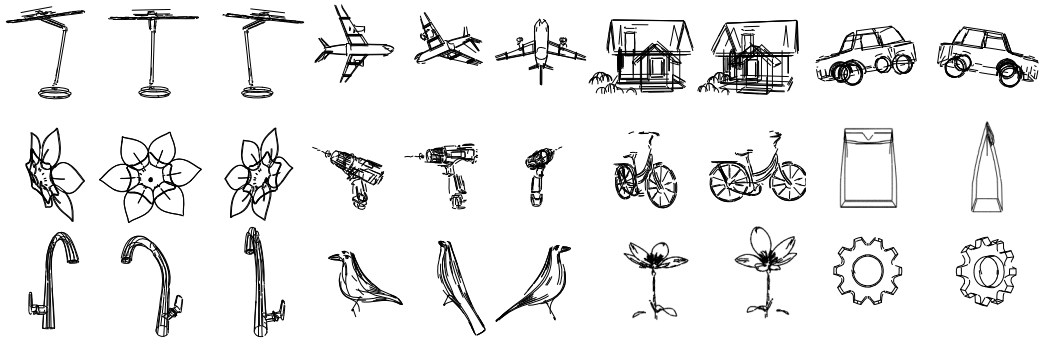

Figure 1: We propose *ViewCraft3D (VC3D)*, a method to generate 3D vector graphics from a single image. VC3D can leverage 3D prior knowledge to generate high-quality and view-consistent 3D vector graphics.

Abstract

3D vector graphics play a crucial role in various applications including 3D shape retrieval, conceptual design, and virtual reality interactions due to their ability to capture essential structural information with minimal representation. While recent approaches have shown promise in generating 3D vector graphics, they often suffer from lengthy processing times and struggle to maintain view consistency. To address these limitations, we propose **V**iew**C**raft**3D** (VC3D), an efficient method that leverages 3D priors to generate 3D vector graphics. Specifically, our approach begins with 3D object analysis, employs a geometric extraction algorithm to fit 3D vector graphics to the underlying structure, and applies view-consistent refinement process to enhance visual quality. Our comprehensive experiments demonstrate that VC3D outperforms previous methods in both qualitative and quantitative evaluations, while significantly reducing computational overhead. The resulting 3D sketches maintain view consistency and effectively capture the essential characteristics of the original objects. Project page: `https://zhtjtcz.github.io/VC3D_page/`.

1 Introduction

Three-dimensional vector graphics offer a unique balance between abstraction and comprehensibility, using minimal line elements to convey complex spatial information. These economical representations have become integral to diverse computing applications, from improving immersive experiences in virtual environments to facilitating 3D shape retrieval and reconstruction tasks [49, 20, 19, 13, 48]. In virtual reality creation environments, 3D vector graphics serve as intuitive building blocks that allow artists to materialize spatial concepts directly within immersive spaces [51, 1, 50], bridging the gap between imagination and digital realization. Recent interactive sketching tools [2, 1, 52]

Preprint. Under review.

have enhanced these creative capabilities by enabling direct manipulation in 3D space. Despite these advances, creating effective 3D vector graphics remains prohibitively difficult for non-specialists due to the intricate combination of spatial reasoning, technical interface skills, and artistic judgment required. This expertise barrier significantly limits widespread adoption and accessibility, highlighting the need for automated approaches that can generate high-quality 3D vector graphics without requiring users to have specialized training or artistic expertise.

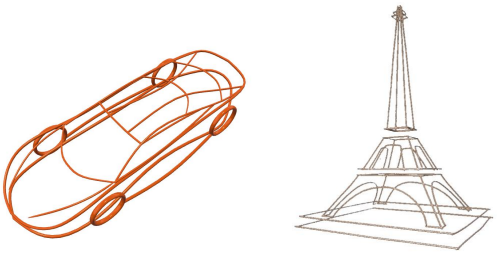

Figure 2: Examples of VR sketches [31].

Recent years have witnessed remarkable progress in 2D vector graphics generation. Works like CLIPasso [34] and CLIPDraw [7] pioneered the use of CLIP's visual-semantic understanding to guide vector graphics optimization. Building on these foundations, methods such as VectorFusion [11], DiffSketcher [41], and SVGDreamer [43] further leveraged diffusion models to achieve higher fidelity and controllability in vector graphics generation. Concurrently, the field of 3D content creation [44, 54, 15, 38, 35, 46] has been revolutionized by neural rendering techniques and generative models, making high-quality 3D asset creation increasingly accessible. The convergence of these advancements has catalyzed research in 3D vector graphics, with pioneering works like 3Doodle [5] and Diff3DS [53] demonstrating the feasibility of generating expressive 3D line drawings. These approaches have achieved impressive results in creating 3D vector graphics. However, existing methods predominantly rely on 2D generative priors—leveraging models like CLIP [26] and diffusion model [28] as supervision signals—while employing Score Distillation Sampling (SDS) [23] for optimization in 2D projection space rather than directly in 3D. These indirect approaches inherit a fundamental limitation of 2D SDS optimization: cross-view inconsistency, which constrains the ability of methods [23, 53]—where the same 3D element appears inconsistently from different viewpoints. Even with the use of more powerful pretrained models, these approaches often struggle to generate coherent 3D vector graphics that remain consistent across arbitrary viewpoints. For example, Diff3DS [53] employs MVDream [30] to tackle this issue, but the improvement is only partial. On the other hand, 2D priors from pretrained image generation models offer only conceptual-level guidance, lacking precise recovery of critical lines typically found in human-drawn 3D sketches, as illustrated in Figure 2. As a result, the generated outputs often suffer from messy strokes, missing details, and low structural fidelity.

To overcome these challenges, we propose **V**iew**C**raft**3D** (VC3D), a novel approach that leverages 3D priors for generating high-fidelity and view-consistent 3D vector graphics. Instead of relying on optimization using 2D priors [53, 5], our method is grounded in 3D geometric attributes within the 3D domain. This allows it to naturally inherit the cross-view consistency of the 3D object while faithfully preserving its spatial structure and geometric details, as illustrated in Figure 1. Specifically, we start by reconstructing a 3D mesh using a pre-trained image-to-3D model. Based on the resulting mesh, we identify salient regions in 3D space that capture the object's key structural features. We then perform point-level clustering using spatial proximity and orientation alignment. These clusters are subsequently fitted with 3D Bézier curves, and Chamfer Distance loss is used to ensure accurate geometric approximation. To further refine these vector graphics, we introduce a 3D score distillation sampling loss based on pretrained 3D generative models, which optimizes the Bézier curve parameters to enhance both visual quality and structural fidelity. This approach maintains view consistency by construction, as the optimization occurs directly in 3D space guided by 3D priors.

In summary, our contributions are threefold:

- We propose ViewCraft3D (VC3D), a novel framework for generating high-fidelity 3D vector graphics that leverages 3D priors rather than 2D projections;
- We develop a two-phase optimization approach combining geometric fitting with 3D-prior guided refinement, significantly improving visual quality.
- We conduct extensive experiments demonstrating that our approach outperforms existing methods in both view consistency and generation speed. The results suggest promising directions for future studies.

2 Related Work

2.1 2D Vector Graphics Generation

Early approaches in 2D SVG generation, such as CLIPasso [34] and CLIPDraw [7], utilized the visual-semantic understanding of the CLIP model [26] to guide vector optimization. Subsequent research introduced more sophisticated approaches, notably those employing diffusion models [8, 28, 6]. Work like VectorFusion [11], DiffSketcher [41], SVGDreamer [43], and SVGDreamer++ [42] demonstrated significant improvements in generation quality by employing Score Distillation Sampling [23]. This technique effectively transfers the generative capabilities of pixel-based models to the vector domain. In addition, SVGFusion [40] explored the use of the DiT architecture [22] to generate SVG. Furthermore, specialized approaches have been developed for specific applications. These include Word-as-image [10] for typographic design, CLIPascene [33] for scene sketching with varying abstraction levels, and VectorPainter [9] for stylized graphics synthesis.

More recently, efforts have focused on mitigating the computational cost associated with iterative optimization. Works based on autoregressive models, such as Iconshop [37], have demonstrated the potential for rapid generation, significantly reducing processing times. Concurrently, the adaptation of large language models (LLMs) for SVG generation has emerged as another promising research avenue, with works like LLM4SVG [39] and Chat2SVG [36]. And OmniSVG [45] attempts to employ Vision-Language Models (VLMs) as end-to-end multimodal SVG generators. Together, these recent advancements aim to ensure high-quality generation while paving the way for future extensions into 3D representations.

2.2 Recent Advances in 3D Content Generation

Recent years have witnessed remarkable progress in 3D content generation driven by diffusion-based approaches. Early works like Zero-123 [17] pioneered single-image view synthesis using geometric priors from diffusion models, while One-2-3-45 [16] extended this to generate full 360-degree textured meshes. Multi-view consistency became a focus with MVDream [30], which serves as an implicit 3D prior through multi-view image generation, and Wonder3D [18], which employs cross-domain attention for consistent normal and color generation. Recent innovations have further elevated capabilities: Unique3D [35] improved fidelity through multi-level upscaling, HunYuan3D [44, 54] achieved photorealistic quality, TripoSG [15] utilized triplane optimization with large-scale data, and Hi3DGen [46] enhanced geometric fidelity through normal bridging. These cutting-edge approaches primarily focus on generating complete 3D assets with textures and materials, while our work emphasizes the creation of 3D vector graphics that maintain characteristic abstractions and representational efficiency. By leveraging the 3D understanding embedded in these advanced models, particularly TripoSG's structural representations, we guide our vector optimization toward semantically meaningful and view-consistent results.

2.3 3D Vector Graphics Generation

Building upon both the 2D vector graphics techniques and recent 3D generation advances discussed above, 3D vector graphics generation has emerged as a promising research direction. These representations extend the fundamental advantages of 2D vector graphics while leveraging 3D generative capabilities to model complex spatial structures and depth information. This integration enhances their utility in diverse fields, including web development and digital art. In artistic contexts, works like DreamWire [25] and Fabricable 3D Wire Art [32] have showcased the potential of 3D vector graphics to create compelling, view-dependent visual effects, where the perceived objects change based on viewing angle.

To harness these benefits and enable such advanced applications, the development of robust 3D vector graphics generation techniques has become a key research focus. Initial explorations in this area include 3Doodle [5], which pioneered a method for generating 3D vector graphics from multi-view images of the target object. Subsequently, Diff3DS [53] utilized the Score Distillation Sampling to produce 3D vector graphics conditioned on text or image input. Dream3DVG [47] leverages the optimization process of a 3D Gaussian Splatting [12] to establish a coarse-to-fine generation approach. However, a notable aspect of these current generative approaches is their predominant reliance on 2D view-specific loss for optimization. View-specific loss is computed independently

per camera view, and gradients are aggregated across views during optimization. Without strong 3D regularization (e.g., geometry priors or multi-view constraints), this results in locally optimal solutions per view that can conflict globally. While yielding impressive outcomes, this strategy may not fully exploit 3D spatial cues, potentially leading to challenges such as view inconsistency in the final 3D vector representations.

3 Methodology

3.1 Overview

In this section, we introduce **V**iew**C**raft**3D** (**VC3D**), an optimization based method that creates a 3D vector graphic $\mathcal{S}^{3D}$ based on an input image $\boldsymbol{I}$. We define a 3D vector graphic $\mathcal{S}^{3D}$ as a set of 3D Bézier curves $\{C_i\}_{i=1}^{n}$. The curves are defined by a set of control points $\{P_{i,j}\}_{j=1}^{m}$, where $P_{i,j} \in \mathbb{R}^3$ is the j-th control point of the i-th curve.

Our method workflow is illustrated in Figure 3. It begins by reconstructing a 3D mesh $\mathcal{M}$ from a user-provided image $\boldsymbol{I}$ using an image-to-3D model [15]. We then apply a two-stage process on the resulting mesh, consisting of Bézier curve fitting followed by detail refinement. The primary structure fitting stage identifies high-curvature regions in the reconstructed mesh, converts them into a point cloud, and fits Bézier curves to approximate these structures. The detail refinement stage re-initializes additional curves in regions overlooked during the first stage and optimizes them using Score Distillation Sampling(SDS) loss [23], leveraging priors from the diffusion model to guide the optimization and enhance fine-grained details in the resulting 3D vector graphic representation. This two-stage approach ensures both structural accuracy and high-fidelity detail preservation.

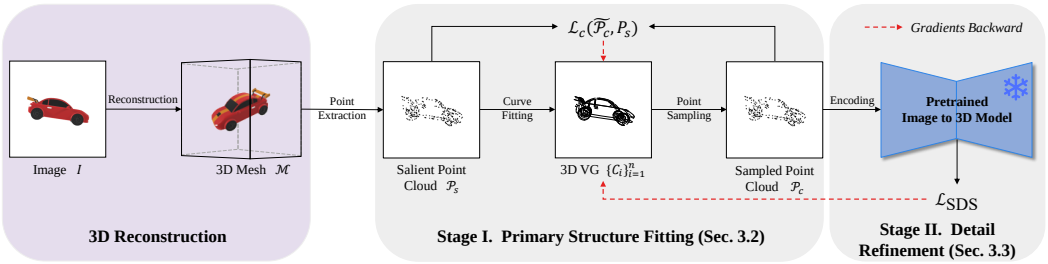

Figure 3: The overall architecture of the proposed method, showcasing the initial generation of 3D Vector Graphic (3D VG) from an input image and subsequent detail refinement using a pretrained image-to-3D model.

3.2 Stage I: Primary Structure Fitting

In this stage, we extract key structural information from the reconstructed mesh $\mathcal{M}$ and use it to fit 3D Bézier curves. The fitting process is further optimized using a specially designed Chamfer Distance loss.

3.2.1 Salient Point Cloud Extraction

To identify high-curvature regions on the mesh, we adopt the Sharp Edge Sampling (SES) process from Dora [4] to extract a salient point cloud. We traverse each edge of the mesh. For each edge, if it belongs to two adjacent faces, we compute the angle between the normal vectors of these two faces. If the angle is below a predefined threshold, we consider this edge as a salient edge. To address the challenge of extracting salient edges from smooth surfaces (e.g., spheres), we uniformly sample camera parameters on a horizontal plane. For each sampled viewpoint, we compute the front faces and back faces. Edges shared by a front face and a back face are identified as silhouette edges. All such silhouette edges are subsequently incorporated into the salient edge set. After identifying all salient edges, we sample points along these edges to create a point cloud $\mathcal{P}_s$ as ground truth.

3.2.2 Point Cloud Clustering

After obtaining the salient point cloud $\mathcal{P}_s$, we aggregate these discrete points into clusters suitable for Bézier curve fitting. Inspired by EdgeGaussians [3], we perform clustering for edge fitting based on vertex orientations. While EdgeGaussians directly utilize the principal directions of 3D Gaussians as orientation vectors, such directional information is absent in our discrete point cloud $\mathcal{P}_s$. To address this, we introduce an initialization step to estimate orientation vectors for each point in $\mathcal{P}_s$. Specifically, for each point p, we first identify its k nearest neighbors and then apply Principal Component Analysis (PCA) [21] to the local neighborhood. The resulting primary eigenvector is used as an approximation of p's orientation vector $\vec{v_p}$.

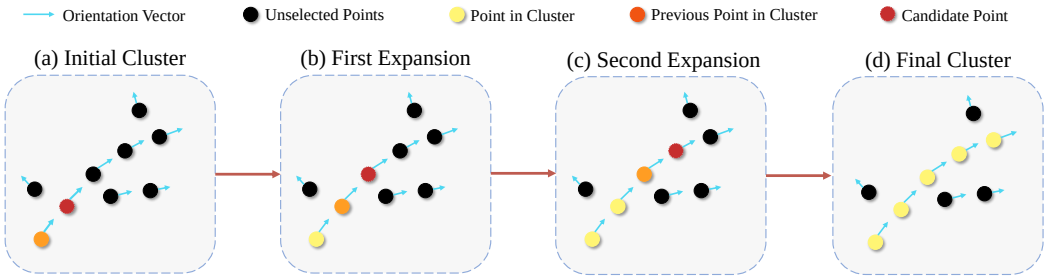

Figure 4: The visualization process of Point Cloud Clustering. Each point is assigned an orientation vector (blue arrows). In (a), the orange point initializes the cluster, and a candidate point (red) is evaluated based on spatial proximity and orientation similarity. In (b), the candidate point meets both criteria and is incorporated into the cluster. This process iterates until the final cluster is formed, as shown in (d).

Once orientation vectors are assigned to each point, we partition the point cloud into multiple clusters using an iterative expansion process as shown in Fig 4. Each cluster is initialized from a randomly selected starting point and grows by progressively incorporating neighboring points that satisfy both spatial proximity and orientation similarity. Specifically, we use the most recently added point in the cluster, denoted as p, to guide the selection of the next candidate. A candidate point q is added to the current cluster if it meets two criteria: (1) it lies within the spatial neighborhood of the cluster, i.e., $dis(p, q) \leq d_{\text{thresh}}$; and (2) the direction of the new edge formed by p and q aligns with the orientation vector of q, i.e., $\arccos\left(\left|\frac{\vec{pq}}{|\vec{pq}|} \cdot \frac{\vec{v_q}}{|\vec{v_q}|}\right|\right) < \theta_{\text{thresh}}$. The expansion continues until no additional points can be incorporated under these constraints, completing one cluster.

This process repeats across the point cloud to generate a complete set of directional clusters, each representing potential curves for subsequent Bézier fitting. The randomized clustering method ensures comprehensive coverage of all geometric features while avoiding bias toward specific regions. Finally, we filter the small clusters with fewer than τ points, as they are likely to be noise or outliers.

3.2.3 Bézier Curves Fitting

After obtaining these clusters, we attempt to fit them with either straight lines or Bézier curves, selecting the one with the least error as the fitting result for that cluster.

The fitting process is performed using the Chamfer Distance loss function [24] to minimize the distance between the Bézier curves and the salient point cloud $\mathcal{P}_s$ obtained in Sec. 3.2.1. The fitted Bézier curves are denoted as $\{C_i\}_{i=1}^n$. For our implementation, we use cubic Bézier curves with four control points $P_0, P_1, P_2, P_3 \in \mathbb{R}^3$. The parametric equation for a 3D cubic Bézier curve can be written as:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \quad t \in [0, 1] \tag{1}$$

To generate a point cloud from a set of Bézier curves $\{C_i\}_{i=1}^n$, we uniformly sample s points along each curve by evaluating the parametric function $B(t)$ at $t_j = \frac{j-1}{s-1}$ for $j = 1, \ldots, s$. The resulting point cloud $\mathcal{P}_c$ is defined as:

$$\mathcal{P}_c = \left\{ B_i(t_j) \mid i \in \{1, \ldots, n\},\, j \in \{1, \ldots, s\} \right\}. \tag{2}$$

Chamfer Distance loss is computed as follows:

$$\mathcal{L}_c(\tilde{\mathcal{P}}_c, P_s) = \frac{\lambda}{|\tilde{\mathcal{P}}_c|} \sum_{p \in \tilde{\mathcal{P}}_c} \min_{q \in \mathcal{P}_s} \|p - q\|^2 + \frac{1}{|\mathcal{P}_s|} \sum_{q \in \mathcal{P}_s} \min_{p \in \tilde{\mathcal{P}}_c} \|p - q\|^2 \tag{3}$$

where $\tilde{\mathcal{P}}_c$ denotes $\mathcal{P}_c$ augmented with Gaussian noise (introduced for data augmentation), and λ is a hyperparameter to balance the two terms. The generation of point cloud $\mathcal{P}_c$, which relies on the Bézier curve formulation in Eq. 1, is differentiable with respect to the curve's control points. Consequently, the chain rule enables the gradients from $\mathcal{L}_c$ to propagate back to these control points, facilitating their iterative optimization.

3.3 Stage II: Detail Refinement

Some objects may contain intricate-to-approximate regions that Stage I might miss due to limitations in the salient point cloud extraction or clustering process. These regions are identified by analyzing the mesh's vertex distribution and locating areas not adequately covered by the point cloud $\mathcal{P}_c$ generated in Stage I. For such cases, we introduce an additional refinement stage to handle these regions by distilling priors from a pretrained image-to-3D model.

First, the parameters of the initial curves $\{C_i\}_{i=1}^{n}$ from Stage I are frozen. We randomly initialize new Bézier curves $\{C_i'\}_{i=1}^{n'}$ (with parameters θ') in regions that are intricate to approximate, thereby complementing the primary structure. To jointly represent both curve sets, we sample a combined point cloud $\mathcal{P}_{combined} = \mathcal{P}_c \cup \mathcal{P}_{c'}$ and encode it into a latent space $\mathcal{Z}$ using a pretrained VAE encoder from [15]: $z = \mathcal{E}(\mathcal{P}_{combined})$. Then we refine only the new parameters θ' via SDS loss [23], supervised by the input image $\boldsymbol{I}$.

$$\nabla_{\theta'} \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t,\epsilon} \left[w(t) \left(\epsilon_\phi(z_t, \boldsymbol{I}; t) - \epsilon \right) \frac{\partial z}{\partial \theta'} \right] \tag{4}$$

where z_t is the noised latent variable at timestep t, ϵ_ϕ is the denoising model conditioned on $\boldsymbol{I}$, and $w(t)$ is a weighting function. This process iteratively adjusts the newly added Bézier curves to fill in missing details from Stage I, ensuring consistency and high fidelity in the final 3D vector representation $\mathcal{S}^{3D} = \{C_i\}_{i=1}^{n} \cup \{C_i'\}_{i=1}^{n'}$.

3.4 3D Vector Graphics Rendering

To enable both qualitative visualization and quantitative evaluation, 3D vector graphic S^{3D} should be projected onto a 2D plane. As proved by 3Doodle [5], the perspective projection of a 3D Bézier curve onto a 2D plane yields a 2D rational Bézier curve. Given a 3D curve $B(t)$ and image plane at $z = f$ (where z is the depth axis and f the focal length), the projection is:

$$B^{2D}(t) = \begin{pmatrix} B_x(t) \frac{f}{B_z(t)} \\ B_y(t) \frac{f}{B_z(t)} \end{pmatrix} \tag{5}$$

Consequently, by defining camera parameters in 3D space, we obtain a set of 2D rational Bézier curves corresponding to the camera's viewpoint. These curves can be rendered using DiffVG [14] to generate corresponding SVG files, which are subsequently utilized for both quantitative and qualitative analysis.

4 Experiment

4.1 Implementation Details

Our VC3D framework is implemented in PyTorch. For primary structure fitting stage, we set the distance threshold $d_{\text{thresh}} = 0.05$ and angle threshold $\theta_{\text{thresh}} = 50°$. Each Bézier curve is defined by 4 control points, with $s = 64$ sample points per curve for optimization. For detail refinement stage, we employ TripoSG [15] as our pretrained image-to-3D model, with SDS loss weight set to 2×10^{-4}. We use the SGD optimizer [27] with a learning rate of 5×10^{-3}.

All experiments are conducted on a single NVIDIA RTX 4090 GPU. For each input, our method typically produces fewer than 100 Bézier curves. Our full method takes about 30 minutes to generate a vector graphic, with 100 optimization steps for Stage I and 200 steps for Stage II. We collected 40 images from prior works and online sources as inputs. All generated 3D vector graphics are rendered into 12 views using identical camera parameters, upon which the metrics are computed.

4.2 Experimental Results

We compare our approach with two state-of-the-art methods in 3D vector graphics generation: Diff3DS [53], which designs a depth-aware differentiable rasterizer and leverages 2D diffusion model priors through SDS loss to generate 3D vector graphics from text or images, and 3Doodle [5], which employs perceptual losses with multi-view guidance to obtain 3D Bézier curve representations of objects. To comprehensively evaluate the quality and fidelity of the generated 3D vector graphics, we employ CLIPScore [26] to measure semantic alignment between rendered views and input images Furthermore, we use an aesthetic indicator [29] to quantify the aesthetic value.

4.2.1 Qualitative Evaluation

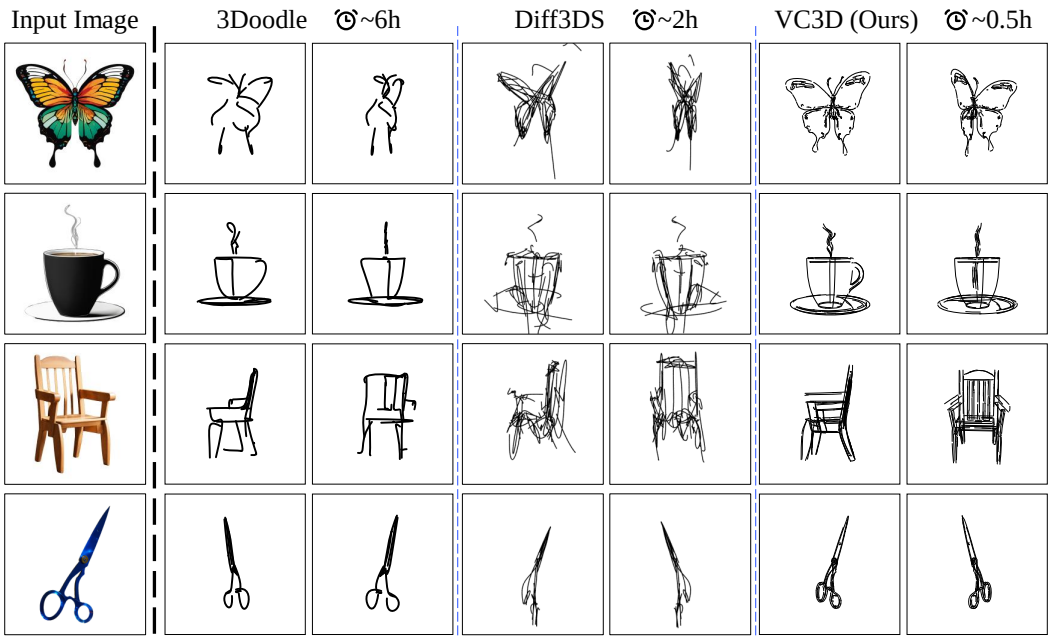

Figure 5: Qualitative comparison of different methods. Diff3DS and VC3D use a single image $\boldsymbol{I}$ as input, while 3Doodle uses 120 rendered images of the mesh reconstruction result $\mathcal{M}$ as input.

Figure 5 presents qualitative comparisons between our method and previous work, 3Doodle [5] and Diff3DS [53]. As shown, VC3D produces cleaner, more accurate, and more view-consistent 3D vector graphics. Previous works struggle to capture fine details in reference images, such as the patterns on butterfly or the handle of the coffee cup. Additionally, their outputs often contain excessive messy lines (e.g., the chair example).

4.2.2 Quantitative Evaluation

Table 1 presents the quantitative analysis results of all methods. Our method outperforms both previous approaches in CLIPScore and Aesthetic Score metrics. Our method achieves a cosine similarity of 0.799, which is higher than the 0.729 achieved by 3Doodle and the 0.673 achieved by Diff3DS. At the same time, we achieved the highest score in Aesthetic Score. These superior results demonstrate our method's ability to generate semantically and geometrically superior 3D vector graphics.

In addition to the metrics mentioned above, our method demonstrates significant advantages in generation time. Our method requires only a few SDS loss optimization steps, significantly reducing generation time. The total runtime for two stages is approximately 0.5 hours, showing notable improvements compared to 3Doodle (~6 hours) and Diff3DS (~2 hours).

Table 1: Quantitative comparison of VC3D and previous methods on evaluation metrics. The **bold numbers** represent the best performance.

Method	CLIPScore ↑	Aesthetic Score ↑
3Doodle	0.729	4.122
Diff3DS	0.673	3.769
VC3D (Ours)	**0.799**	**4.352**

Table 2: Ablation study results comparing different variants of our proposed method. The **bold numbers** represent the best performance.

Method	CLIPScore ↑	Aesthetic Score ↑
Variant 1	0.779	4.096
Variant 2	0.805	4.167
Full Method	**0.818**	**4.297**

4.3 Ablation Studies and Analysis

To demonstrate the respective contributions of the Chamfer Distance loss and SDS loss, we performed ablation experiments. We selected a subset of 20 images from the inputs in Section 4.2.1, where all examples were optimized with SDS loss. And results were recorded after three distinct stages, corresponding to three variants: (1) **Variant 1:** This variant refers to the model with the salient point cloud extraction and the point cloud clustering, (2) **Variant 2:** This variant is the model with the first stage only, *i.e.*, Primary Structure Fitting, and (3) **Full Method**: This is our proposed method, which comprises two stages. The results are shown in Table 2. The improvements of the **Variant 2** against the **Variant 1** indicate the benefits brought by using CD loss for optimization. Comparing the performance of **Variant 2** and **Full Method**, it is clear that the detail refinement stage can further improve the performance in CLIPScore and Aesthetic Score metrics.

In Figure 6, we show the optimization process of Chamfer Distance loss. The initially fitted Bézier curves often fail to accurately cover the salient point cloud $\mathcal{P}_s$. The coherence between curves is also suboptimal. As optimization progresses, the curves gradually extend to form more complete structures, ultimately achieving both improved coverage of the salient features and enhanced inter-curve coherence while preserving the overall geometric fidelity of the original shape.

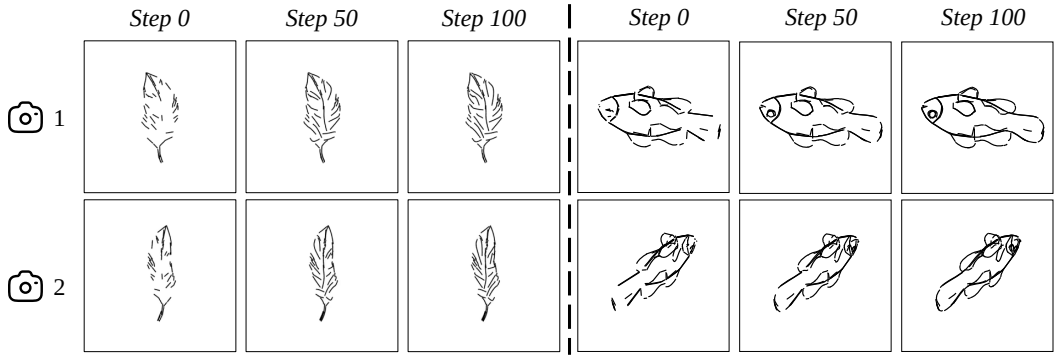

Figure 6: Illustration of the optimization process using Chamfer Distance loss $\mathcal{L}_c$.

The visual improvements brought by the SDS loss can be observed in Figure 7, where the refinement stage compensates for previously overlooked details (e.g., terminal branches on corals) and improves the structural coherence of the 3D vector graphics.

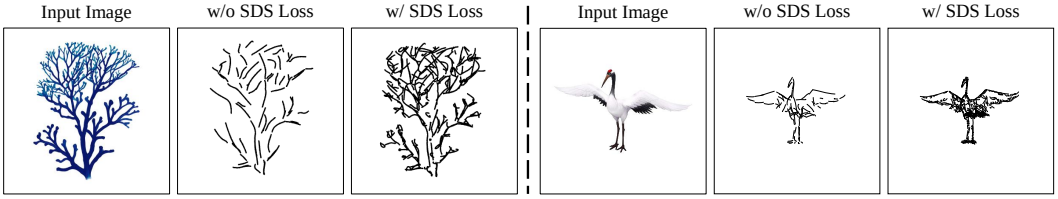

Figure 7: Illustration of the optimization effect of SDS loss $\mathcal{L}_{sds}$. SDS loss effectively recovers missing structural information while enhancing geometric detail representation.

These results demonstrate that our two-stage approach effectively balances structural accuracy with visual quality, leading to more compelling and semantically accurate 3D vector graphics.

We also experimented with the number of Bézier curves. Since the number of Bézier curves is equal to the number of point clusters, we can control the number of curves by adjusting the cluster count, i.e., by changing the filtering threshold τ in Point Cloud Clustering stage. As shown in Figure 8, when $\tau = 10$, point clusters with fewer than 10 points are removed. Increasing the threshold eliminates more clusters, reducing the number of Bézier curves retained and producing an abstract result.

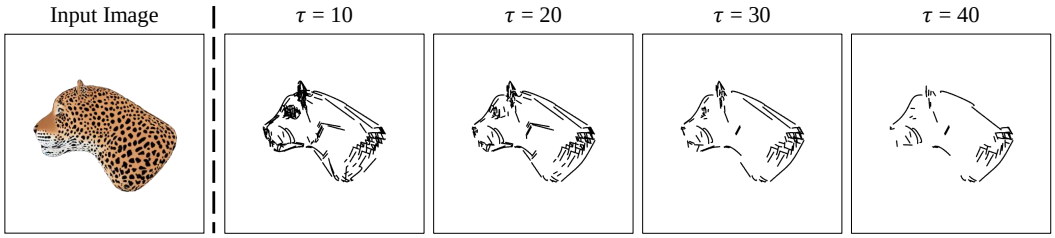

Figure 8: Effect of the filtering threshold in Point Cloud Clustering. The number of Bézier curves gradually decreases as τ increases, producing a more abstract effect.

5 Limitations and Future Works

While VC3D efficiently generates view-consistent 3D vector graphics, it currently lacks occlusion relationships between curves. When rendering to 2D images, all curves share uniform transparency, which may compromise visual fidelity. Future work could address this by utilizing the corresponding mesh. Since the mesh is available, the position of each Bézier curve relative to the camera parameters can be determined, allowing for the processing of occlusion relationships.

In addition, considering that our method can generate corresponding 3D vector graphics from meshes with minimal time cost, we can build 3D vector graphics datasets based on open-source mesh datasets in the future, providing a research foundation for subsequent work.

6 Conclusion

In this paper, we present VC3D, a novel framework for generating view-consistent 3D vector graphics using 3D priors. Operating directly in 3D space rather than 2D projection planes, our approach effectively addresses view inconsistency issues. Our two-stage algorithm first identifies salient structures through geometric clustering and Bézier curve fitting, then refines results using SDS loss with a pretrained image-to-3D model. Experiments demonstrate that VC3D preserves geometric characteristics while maintaining view consistency across viewpoints, with advantages in generation efficiency. This research makes high-quality 3D vector graphics creation more accessible and applicable to virtual reality, shape retrieval, and conceptual design.

References

[1] Rahul Arora and Karan Singh. Mid-air drawing of curves on 3d surfaces in virtual reality. *ACM Transactions on Graphics (TOG)*, 40(3):1–17, 2021.

[2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160, 2008.

[3] Kunal Chelani, Assia Benbihi, Torsten Sattler, and Fredrik Kahl. Edgegaussians-3d edge mapping via gaussian splatting. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3268–3279. IEEE, 2025.

[4] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. *arXiv preprint arXiv:2412.17808*, 2024.

[5] Changwoon Choi, Jaeah Lee, Jaesik Park, and Young Min Kim. 3doodle: Compact abstraction of objects with 3d strokes. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.

[6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[7] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35:5207–5218, 2022.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Juncheng Hu, Ximing Xing, Jing Zhang, and Qian Yu. Vectorpainter: Advanced stylized vector graphics synthesis using stroke-style priors. *arXiv preprint arXiv:2405.02962*, 2024.

[10] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-as-image for semantic typography. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

[11] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2023.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[13] Jaeah Lee, Changwoon Choi, Young Min Kim, and Jaesik Park. Recovering dynamic 3d sketches from videos. *arXiv preprint arXiv:2503.20321*, 2025.

[14] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.

[15] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

[16] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023.

[17] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[18] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024.

[19] Ling Luo, Pinaki Nath Chowdhury, Tao Xiang, Yi-Zhe Song, and Yulia Gryaditskaya. 3d vr sketch guided 3d shape prototyping and exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2023.

[20] Ling Luo, Yulia Gryaditskaya, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Towards 3d vr-sketch to 3d shape retrieval. In *2020 International Conference on 3D Vision (3DV)*, pages 81–90. IEEE, 2020.

[21] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[22] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[23] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[25] Zhiyu Qu, Lan Yang, Honggang Zhang, Tao Xiang, Kaiyue Pang, and Yi-Zhe Song. Wired perspectives: Multi-view wire art embraces generative ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6149–6158, 2024.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[27] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[29] Christoph Schuhmann. Improved aesthetic predictor, 2022.

[30] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[31] Sketchfab. Sketchfab - the platform for 3d and ar on the web, 2023. 3D VR sketch.

[32] Kenji Tojo, Ariel Shamir, Bernd Bickel, and Nobuyuki Umetani. Fabricable 3d wire art. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[33] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4146–4156, 2023.

[34] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

[35] Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[36] Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. *arXiv preprint arXiv:2411.16602*, 2024.

[37] Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. Iconshop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

[38] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[39] Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. Empowering llms to understand and generate complex vector graphics. *arXiv preprint arXiv:2412.11102*, 2024.

[40] Ximing Xing, Juncheng Hu, Jing Zhang, Dong Xu, and Qian Yu. Svgfusion: Scalable text-to-svg generation via vector space diffusion. *arXiv preprint arXiv:2412.10437*, 2024.

[41] XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15869–15889. Curran Associates, Inc., 2023.

[42] Ximing Xing, Qian Yu, Chuang Wang, Haitao Zhou, Jing Zhang, and Dong Xu. Svgdreamer++: Advancing editability and diversity in text-guided svg generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[43] Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. Svgdreamer: Text guided svg generation with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4546–4555, 2024.

[44] Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, et al. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293*, 2024.

[45] Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. Omnisvg: A unified scalable vector graphics generation model. *arXiv preprint arXiv:2504.06263*, 2025.

[46] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 3, 2025.

[47] Li Yidi, Xiao Jun, Lu Zhengda, Wang Yiqun, and Jiang Haiyong. Empowering vector graphics with consistently arbitrary viewing and view-dependent visibility. pages 1–14, 2025.

[48] Emilie Yu, Rahul Arora, J Andreas Baerentzen, Karan Singh, and Adrien Bousseau. Piecewise-smooth surface fitting onto unstructured 3d sketches. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022.

[49] Emilie Yu, Rahul Arora, Tibor Stanko, J Andreas Bærentzen, Karan Singh, and Adrien Bousseau. Cassie: Curve and surface sketching in immersive environments. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–14, 2021.

[50] Emilie Yu, Kevin Blackburn-Matzen, Cuong Nguyen, Oliver Wang, Rubaiat Habib Kazi, and Adrien Bousseau. Videodoodles: Hand-drawn animations on videos with scene-aware canvases. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.

[51] Emilie Yu, Fanny Chevalier, Karan Singh, and Adrien Bousseau. 3d-layers: Bringing layer-based color editing to vr painting. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[52] Xue Yu, Stephen DiVerdi, Akshay Sharma, and Yotam Gingold. Scaffoldsketch: Accurate industrial design drawing in vr. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 372–384, 2021.

[53] Yibo Zhang, Lihong Wang, Changqing Zou, Tieru Wu, and Rui Ma. Diff3ds: Generating view-consistent 3d sketch via differentiable curve rendering. *arXiv preprint arXiv:2405.15305*, 2024.

[54] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.