






# View-Independent Wire Art Modeling via Manifold Fitting

HuiGuang Huang<sup>1</sup> , Dong-Yi Wu<sup>1</sup> , Yulin Wang<sup>1</sup> , Yu Cao<sup>2</sup>  and Tong-Yee Lee<sup>\*†1</sup> 

<sup>1</sup>National Cheng Kung University, Taiwan

<sup>2</sup>Saint Francis University, Hong Kong

## Abstract

This paper presents a novel fully automated method for generating view-independent abstract wire art from 3D models. The main challenge in creating line art is to strike a balance among abstraction, structural clarity, 3D perception, and consistent aesthetics from different viewpoints. Many existing approaches have been proposed, including extracting wire art from mesh, reconstructing it from pictures, etc. But they all suffer from the fact that the wires are usually very unorganized and cumbersome and usually can only guarantee the observation effect of specific viewpoints. To overcome these problems, we propose a paradigm shift: instead of predicting the line segments directly, we consider the generation of wire art as an optimization-driven manifold-fitting problem. Thus we can abstract/generalize the 3D model while retaining the key properties necessary for appealing line art, including structural topology and connectivity, and maintain the three-dimensionality of the line art with a multi-perspective view. Experimental results show that our view-independent method outperforms previous methods in terms of line simplicity, shape fidelity, and visual consistency.

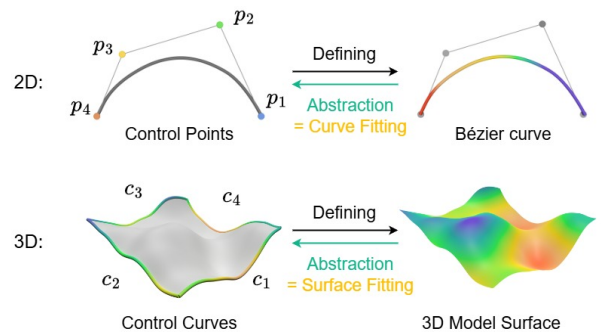
## CCS Concepts

• **Computing methodologies** → **Shape modeling**; • **Mathematics of computing** → **Mathematical optimization**;

## 1. Introduction

Wire art is an artistic technique that abstracts objects into linear forms using metal wires or other similar materials, and it has found widespread application in various handcrafted objects and artistic creations. The creation of effective wire art pieces typically necessitates significant artistic knowledge and experience. Specifically, accurately abstracting the salient features of an object while ensuring that the resulting linear form maintains visual appeal and coherence when viewed from arbitrary viewpoints poses significant challenges for artists.

In previous work, Yeh et al. [YLHL22] proposed simplifying the input mesh via an objective function to extract a wire art representation. Another approach, proposed by Yang et al. [YXFH21], extracts wire art by identifying feature points on the mesh and using a specific viewpoints as a key constraint. However, a common limitation of these early methods is that their results are often optimized for, and only visually appealing from, a single or some selected viewpoint. Furthermore, they typically necessitate manual intervention for refinement and selection. Therefore, it is more laboring work and user-selected dependence. More recently, driven



**Figure 1:** Main idea of our work: wire art can be viewed as a high-level abstraction of the model, like how a set of control points serves as a compact abstraction of a Bézier curve.

by the advancements in deep learning over the past few years, several optimization-based methods have emerged that leverage pre-trained neural networks, often utilizing perceptual loss functions to guide the process [TSBU24, CLPK24]. However, these methods often still struggle with achieving satisfactory multi-view consistency and aesthetic quality. Additionally, the generated wireframes can sometimes be overly dense or cluttered, rendering the results difficult or even impossible to fabricate manually.

† \*Corresponding author: Tong-Yee Lee.

Our project web site available at <https://github.com/THLsword/3DWireArt>

To address these challenges, our work is inspired by recent advances in manifold-based 3D reconstruction [SBS21] and traditional parametric curve representations, leading to a new framework for 3D model fitting. As shown in Figure 1, similar to how Bezier curves can be effectively represented by a lower-dimensional set of control points, we extend this concept to 3D model fitting using Coons patches. By employing our specifically designed energy function, we aim to achieve a concise representation of the 3D model through control curves that define the boundaries of these Coons patches. In this work, we take a 3D model as input. In contrast to existing model-based feature line extraction methods, such as those by Ye et al. [YLHL22] and Yang et al. [YXFH21], we introduce a paradigm shift: rather than directly outputting discrete line segments, we cast the generation of line drawings as an optimization-driven manifold-fitting problem. Our approach is composed of two primary stages. First, we adopt a coarse-to-fine optimization strategy guided by a modality-specific weighting scheme. A visual-saliency prior steers the coarse stage toward globally important regions, whereas feature-space weights—derived from geometric descriptors—refine semantically meaningful details in later stages. Furthermore, we introduce an aesthetic energy term  $E_{aesthetic}$  that plays a critical role in controlling the visual layout of the Coons-patch boundaries. As a post-processing step, we iteratively prune wires by measuring their silhouette contribution: we extract alpha-shape contours in each rendered view, compute the average multi-view Intersection-over-Union with the ground-truth silhouette, and greedily delete the wire in which removal yields the smallest drop in this IoU score.

Our main contributions can be summarized as follows:

- We introduce a novel view-independent wire art modeling method and address the previous wire abstraction challenge by solving a 3D manifold fitting problem.
- We design a coarse-to-fine, multi-modal optimization scheme that requires no human intervention.
- Our framework allows for seamlessly incorporating additional aesthetic controls, such as symmetry, balance, and smoothness, to accommodate specific task-dependent aesthetic requirements.

## 2. Related Work

### 2.1. Wire Art Research

Existing wire art research has two primary paradigms: 3D reconstruction of 2D images or videos and abstraction from 3D models. Liu et al. [LCL\*17] and Wang et al. [WLC\*20] rely on establishing correspondences between adjacent images to construct the 3D structure, with the quality of the wire art being highly dependent on the accuracy of these correspondences. WrapIt [ILB15] decomposes an input 2D line drawing into subgraphs and solves a graph-labeling problem to identify the smoothest, most concise wire composition—though the result remains flat, i.e., restricted to 2D. Hsiao et al. [LCC\*18], and Qu et al. [QYZ\*24] each propose reconstruction-based methods that fuse multi-view observations to generate 3D wire (or sketch) art models from three views. Tojo et al. [TSBU24] reconstruct 3D wire art from multi-view images by leveraging CLIP’s vision-language priors to guide abstract structural recovery. And 3Doodle [CLPK24] leverages multi-view im-

ages as supervision to guide the distribution of sparse and disordered 3D curves into coherent spatial structures. It maintains high multi-view consistency and preserves certain geometric details, but the resulting structure is not physically manufacturable.

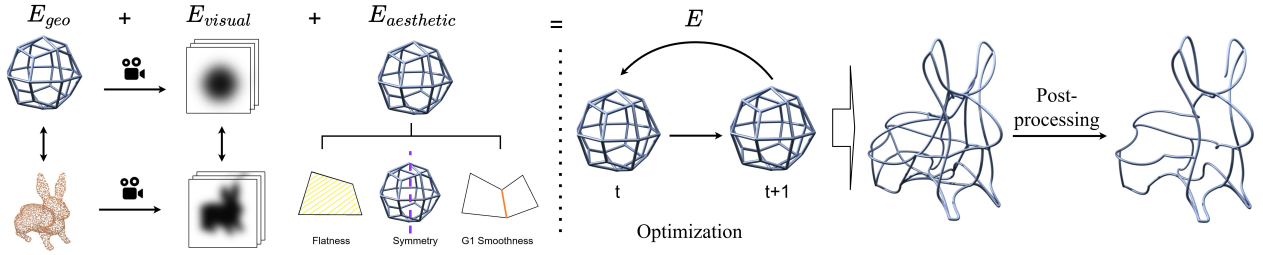
Alternatively, wires can be extracted from an existing 3D model as a simplified representation, used as a standalone artwork, or in rapid prototyping. Miguel et al. [LFZ18] break the sculpture into parts that can be fabricated with a wire-bending machine by solving for collision-free Eulerian paths. WireRoom [YXFH21] assumes the wire follows a single continuous (Eulerian) path and formulates a dynamic traveling-salesman problem to find the optimal route. Yeh et al. [ZCT16], which leverages deep signed-distance-function networks to generate dense wire-like structures. Our method combines 3D and 2D information, modeling 3D wire art via an optimization-driven manifold-fitting strategy without the extensive manual intervention required by previous computational methods. De Goes et al. [DGGDV11] extract skeletal flow lines from 3D surfaces to serve as wire generation paths; Zehnder et al. [ZCT16] leverage deep signed-distance function networks to generate dense, wire-like structures along the model surface. Lira et al. [LFZ18] decompose a 3D sculpture into parts manufacturable by a wire-bending machine by solving collision-free Eulerian paths.

### 2.2. Point Cloud Feature Extraction

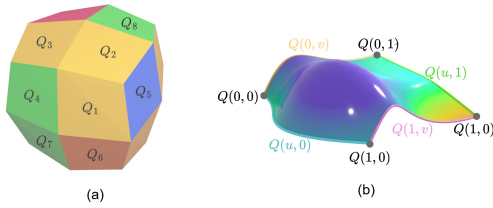
With the rapid advancement of neural network architectures, learning rich representations of 3D geometry has become increasingly effective, offering new opportunities to enhance optimization-based reconstruction pipelines. PointNet [CSKG17] pioneered this direction by operating directly on raw point clouds and employing a sequence of multilayer perceptrons (MLPs) to aggregate global features. Building on this foundation, PointNet++ [JWZ\*18, LD18, BHC21, ZWSW22, WZPB23]—have further advanced point cloud feature learning through innovations such as adaptive neighborhood encoding, grid-based convolutions, and transformer-style attention. By integrating these learned representations into energy terms, modern pipelines achieve more robust and visually coherent 3D reconstructions.

### 2.3. Differentiable rendering

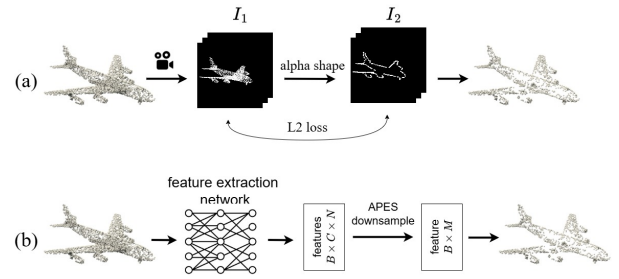
Differentiable rendering techniques have been developed across several domains—including physically based methods [LADL18, NDVZJ19, ZWZ\*19], rasterization-based approaches [LHK\*20], 2D vector graphics renderers [LLGRK20], and custom shader pipelines [YBAF22, BWL\*23]. Worchel and Alexa [WA23] enable differentiable rendering of parametric surfaces by introducing a method to extract triangle meshes from Bézier, B-spline, and NURBS representations in a differentiable manner. Moreover, the PyTorch3D library [RRN\*20] provides an extensible, PyTorch-native framework for efficient differentiable rendering and 3D operations, facilitating seamless integration with deep learning workflows.



**Figure 2:** Pipeline overview. Starting from an input point cloud, we initialize a coarse set of Coons-patch curves and refine them through gradient-based optimization of the total energy  $E = E_{geo} + E_{visual} + E_{aesthetic}$ . The process converges to a concise wire-art representation that remains visually consistent from all viewpoints.



**Figure 3:** (a) A 24-patch manifold template. (b) A Coons patch.



**Figure 4:** The pipeline of visual downsampling (a) and feature downsampling (b).

### 3. Method

#### 3.1. Coons Patch and Template

We observe that prior work in [YLHL22, YXFH21, TSBU24] often fails to maintain good visual quality across multiple viewpoints, resulting in unsatisfactory reconstructions from many angles. This limitation largely stems from their primary reliance on view constraints without effectively exploiting the intrinsic geometric structure or features of the underlying 3D model. Furthermore, approaches such as 3Doodle [CLPK24] and Qu et al. [QYZ\*24] initialize with a high density of disorganized curves, making physical fabrication of the resulting structures challenging. Consequently, there is a clear need for a method that simultaneously achieves geometric conciseness and viewpoint independence for the wire art task. To address these limitations, we propose employing a structure composed of Coons patches as an initial template. The closed topology of basic template such as sphere, cone, torus etc. produces a smoother energy landscape that facilitates more stable global convergence during optimization. This design therefore promotes curve conciseness, view robustness, and reliable convergence in a single unified framework.

Coons patch is a class of parametric surface patches commonly employed in geometric modeling. Defined by their boundary curves and associated blending functions, they provide a mechanism for smoothly interpolating and constructing surfaces. In our work, we adopt Coons patches due to their boundary-driven nature, which ensures exact interpolation of the four boundary curves. Additionally, they offer a more straightforward implementation and incur sub-

stantially lower computational overhead for sampling and rendering, in contrast to more complex higher-order parametric surfaces. As an example shown in Fig. 3, a Coons patch  $Q: [0, 1]^2 \mapsto \mathbb{R}^3$  is bounded by four connected cubic Bézier curves. Each curve  $c$  can be written as  $c(t) = cp_1(1-t)^3 + 3cp_2t(1-t)^2 + 3cp_3t^2(1-t) + cp_4t^3$  in terms of four control points  $cp_1, \dots, cp_4 \in \mathbb{R}^3$ . A Coons patch can then be expressed as

$$Q(u, v) = (1-v)c_1(u) + vc_3(1-u) + uc_2(v) + (1-u)c_4(1-v) - (c_1(0)(1-u)(1-v) + c_1(1)u(1-v) + c_3(1)(1-u)v + c_3(0))uv. \quad (1)$$

The final 3D template consists of a collection of Coons patches, which can be uniquely specified by all control points  $p = \{cp_1, cp_2, \dots, cp_n\}$ .

#### 3.2. Coarse-to-Fine Optimization

Let  $cp$  be the complete set of cubic Bézier control points that bound the Coons-patch manifold. We iteratively minimize a differentiable energy

$$cp^{t+1} = cp^t - \eta \nabla_{cp} E(cp^t), \quad cp^* = \arg \min_{cp} E(cp), \quad (2)$$

where  $\eta > 0$  is the step size and the gradient  $\nabla_{cp} E$  is computed with PyTorch's reverse-mode automatic differentiation. The so-

lution  $cp^*$  represents the optimal control-point configuration that minimizes the total energy  $E$ ; its associated Coons-patch boundaries constitute the final wire-art abstraction.

We adopt a *coarse-to-fine* optimization strategy to mitigate the sensitivity of Coons-patch fitting to local minima encountered when directly optimizing all degrees of freedom. In our context, “coarse-to-fine” does not refer to changes in grid or sampling resolution, but rather to a multi-stage optimization framework in which each stage employs a different energy formulation and weighting scheme. The optimization process begins with energy terms emphasizing global contour alignment and gradually incorporates terms that capture finer local geometric details. This progressive refinement stabilizes the solution space by first enforcing a reliable global structure before focusing on high-frequency accuracy.

Therefore, each of our stages is driven by a distinct **modality-specific weighting scheme**. In the first (coarse) stage we apply *visual weighting* based on view-dependent saliency, ensuring visually prominent areas dominate the early solution. In the second and third (fine) stages, we switch to *feature-space weighting*, computed from learned geometric descriptors, so that subtle but semantically meaningful details receive greater emphasis. These stage-wise weights are injected into the geometric energy, producing a per-point importance factor that scales all residuals in  $E_{geo}$ , which will be described in the next section. Consequently, regions judged salient, whether in image space or feature space, exert a proportionally stronger influence on the descent direction.

*Visual Weighting.* As illustrated in Fig. 4(a), we render the input model using a differentiable renderer [RRN\*20] from  $n$  view-points to obtain an image set  $I_1$ . In our experiments, we set  $n = 6$ , uniformly distributed around the object. Silhouette points are extracted via the alpha-shape algorithm [EM94], producing a refined set  $I_2$ . The per-pixel difference  $\|I_1 - I_2\|_2$  is treated as a differentiable loss, and its gradients are back-propagated through the renderer to compute a saliency map  $W_{vis}^1$  on the surface. During the coarse stage, each sampled surface point is therefore weighted by  $W_{vis}$  when computing chamfer distances, effectively up-weighting highly visible regions.

*Feature Weighting.* However, visual weighting only influences regions that are well observed in the rendered views and primarily serves to initialize the coarse stage. Therefore, as shown in Fig. 4(b), We adopt the APES [WZPB23] strategy, which identifies important points by computing, for each point, the standard deviation of distances between its feature embedding and those of its  $k$ -nearest neighbors in feature space—points with larger deviations are deemed more informative. In our work, we use a pretrained APES network to perform feature-level down-sampling of the input point cloud at 1/4 and 1/8 resolutions, yielding two saliency-guided subsets. These down-sampled points generate weight maps  $W_{feat}^2$  and  $W_{feat}^3$  for stages two and three, respectively.

Coarse-to-fine optimization and the modality-specific weighting scheme establish stable global alignment while injecting image-space saliency and feature-space semantics into the energy, yielding precise and perceptually coherent wire-art abstractions. As shown in Fig. 8, our ablation study further validates the effectiveness of the proposed approach.

### 3.3. Energy functions

In this section, we detail the energy functions that govern the final appearance and quality of the optimized curves derived from our template. The overall energy function,  $E$ , is composed of three main categories:

$$E = E_{geo} + E_{visual} + E_{aesthetic} \quad (3)$$

The complete optimization procedure is summarized in Algorithm 1. And in the experiments, each energy function is assigned a specific weight to balance its influence during optimization, as listed in Tab. 4.1.

#### 3.3.1. Geometric Energy Terms

The optimization is dominated by  $E_{geo}$ , which ensures surface alignment and geometric coherence of the curves. The total geometric energy  $E_{geo}$  consists of three components:

$$E_{geo} = E_{ch} + E_n + E_{ce} \quad (4)$$

**Area-weighted Chamfer Energy [SBS21].** We measure the surface fitting quality with Chamfer loss by sampling points on our patch-based manifold and calculating the nearest point in the point cloud. However, sampling uniformly in the patch’s parameter space  $(u, v)$  does not capture the surface area measure and, consequently, does not yield uniform samples on the regions with higher curvature that end up oversampled. To avoid this problem, given a point cloud  $\mathcal{P}$  and a patch  $\mathcal{Q}$ , the area-weighted Chamfer distance is

$$E_{ch}(\mathcal{P}, \mathcal{Q}) = \sum_{i, (u,v) \in U_{\square}} w_{iuv} \cdot d(\mathcal{Q}(u,v), \mathcal{P}) + \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} d(p, \mathcal{Q}), \quad (5)$$

where  $w_{iuv}$  is the area weighting

$$w_{iuv} = \frac{|J_i(u,v)|}{\sum_{i, (u,v) \in U_{\square}} |J_i(u,v)|}. \quad (6)$$

$J(u, v)$  is the Jacobian of the parametric surface  $\mathcal{Q}(u, v)$  and the notation  $U_{\square} \sim U[1, 2]^2$  indicates that the parameters  $(u, v)$  are uniformly sampled from the unit square  $[1, 2]$ . Additionally,  $d(x, Y)$  denotes the minimum Euclidean distance:

$$d(x, Y) = \min_{y \in Y} \|x - y\|. \quad (7)$$

**Normal Alignment Energy [SBS21].** Similar to the Chamfer energy, we want our patches to have a similar normal as the point cloud. We use

$$d_N(x, Y) = \min_{y \in Y} \|n_x - n_y\|_2^2, \quad (8)$$

where  $n_x$  is the unit normal at point  $x$ . The normal energy function is shown below

$$E_n(\mathcal{P}, \mathcal{Q}) = \sum_{i,(u,v) \in U_{\square}} w_i \cdot d_N(Q(u,v), \mathcal{P}) + \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} d_N(p, \mathcal{Q}). \quad (9)$$

**Concavity Enhancement Energy.** Chamfer energy and normal alignment energy are commonly used, but relying solely on them makes it difficult to achieve sufficient template deformation for highly concave regions. Chamfer energy, in particular, can get trapped in local minima, restricting deformation. To address this, we propose the *concavity enhancement energy*, which improves manifold fitting in concave or cavity regions. Specifically, sampled points on the manifold search along their normal directions for corresponding ground truth points, and the L2 loss is computed accordingly. The energy function is:

$$E_{ce}(\mathcal{P}, \mathcal{Q}') = \sum \|q' - N_P(q')\|^2, \quad (10)$$

where  $\mathcal{Q}'$  represents the set of sampled points from the manifold that fail to fit properly,  $q' \in \mathcal{Q}'$  and  $N_P(q')$  is the projection of  $q'$  onto  $P$  in the normal direction. And  $\mathcal{Q}'$  can be identified by computing the bidirectional KNN. Moreover, our experiments show that since this energy function does not rely on the ground truth point cloud's normal vectors, it therefore potentially enhances enhances the robustness of our method against noise.

### 3.3.2. Visual Energy Terms

**Silhouette Alignment Energy.** To stabilize and globally constrain the *coarse* optimization stage, we introduce a silhouette alignment energy. For each view  $v$ , we employ the differentiable renderer from PyTorch3D [RRN\*20] to produce binary silhouettes:

$$S_C^v = R(C; v), S_P^v = R(P; v). \quad (11)$$

In our method, we will first sample the data into point sets and then use Pytorch3D for differentiable rendering. Here,  $R(\cdot; v)$  denotes the differentiable renderer. The input  $C$  represents the current set of the edges  $c_i$  of templates and  $P$  denotes the target point cloud. Then, we smooth them with a Gaussian kernel  $G_{\sigma}$ . The silhouette alignment energy is the mean-squared error across all pixels and views:

$$E_{sil} = \sum_{v=1}^V \|G_{\sigma} S_C^v - G_{\sigma} S_P^v\|_2^2 \quad (12)$$

### 3.3.3. Aesthetic Energy Terms

Although the geometric energy  $E_{geo}$  effectively drives face fitting, it offers little control over the visual arrangement of the Coons-patch edges. As a result, curves alone cannot faithfully abstract the model, and naively designing curve-specific losses often introduces clutter and visual noise. To address this limitation, we introduce an aesthetic energy  $E_{aesthetic}$  defined on entire patches. These terms promote edge sparsity, uniform spacing, and overall compositional clarity, thereby enhancing the expressive power of the patch boundaries while preserving a clean and coherent visual appearance. The total geometric energy  $E_{aesthetic}$  consists of three components:

$$E_{aesthetic} = E_{flat} + E_{sym} + E_{G1} \quad (13)$$

**Flatness Energy [SBS21].** This is an auxiliary energy function that also contributes to aesthetic control. Flatness energy is designed to make the manifold smoother, prevent self-intersections, and help align its edges with sharp features. We normalize the flatness of each patch, encouraging it to approximate a linear map. For each patch, we calculate the fitting plane  $F_Q = AU_{\square} + b \approx Q$ , using least squares error, and define our energy function as follows

$$E_{flat}(Q_i) = \sum_{i,(u,v) \in U_{\square}} w_{iuv} \|F_{Q_i}(u,v) - Q_i(u,v)\|_2^2. \quad (14)$$

**Symmetry Energy.** We apply symmetry energy when the object exhibits a certain degree of symmetry. This will make the results look more aesthetically pleasing. Let  $S$  denote the fitted plane of symmetry. Our energy function can be defined as:

$$E_{sym}(Q) = \sum \left\| (Q_x^i - S, Q_y^i, Q_z^i) - (S - Q_x^j, Q_y^j, Q_z^j) \right\|_2^2, \quad (15)$$

where  $Q^i$  and  $Q^j$  are a symmetric pair of control points on the template.

**G<sub>1</sub> Smoothness Energy.** Let  $Q = \{q_i\}$  denote the set of all Coons patches, and let  $c_{ij}$  be the common boundary curve between two adjacent patches  $q_i$  and  $q_j$ . For each shared edge  $c_{ij}$  we sample  $K$  points using the normalized curve parameter  $t \in [0, 1]$  (i.e.,  $t_k = \frac{k-1}{K-1}$ ). Denote the unit normals on  $q_i$  and  $q_j$  at parameter  $t_k$  by  $\mathbf{n}_{q_i}(t_k)$  and  $\mathbf{n}_{q_j}(t_k)$ , respectively. The  $G_1$  smoothness loss is then defined as

$$E_{G1} = \frac{1}{|\mathcal{C}_{share}|K} \sum_{c_{ij} \in \mathcal{C}} \sum_{k=1}^K \|\mathbf{n}_{q_i}(t_k) - \mathbf{n}_{q_j}(t_k)\|^2, \quad (16)$$

where  $\mathcal{C}_{share}$  is the set of all shared boundary curves. Minimizing  $E_{G1}$  aligns normals across  $c_{ij}$ , thereby enforcing  $G_1$  continuity between neighbouring patches.

### 3.4. Post-processing: Wire Trimming

The optimized wire set is often overly dense, leading to redundant structures that do not contribute to the perceived silhouette. To prune these redundancies, we adopt a contour-aware greedy strategy that removes wires in which their absence least degrades silhouette fidelity.

For each of the  $N$  predefined viewpoints, we render (i) the target point cloud and (ii) the current wire reconstruction, consistent with the formulation in Equation (11). We set  $N = 9$ , comprising eight side views evenly distributed around the object and one top view. From each binary silhouette image we extract the outer contour using alpha-shape algorithm [EM94]. Let  $A_P^v$  and  $A_C^v$  be the sets of pixels enclosed by the corresponding alpha-shape contours in view  $v$ .

The per-view Intersection-over-Union (IoU) is defined as

$$\text{IoU}^v = \frac{|A_P^v \cap A_C^v|}{|A_P^v \cup A_C^v|}, \quad (17)$$

**Algorithm 1:** Coarse to Fine Optimization

---

**Input:** point cloud  $P$  with normals  
initial control points  $cp^0$   
maximum iteration  $T$  and stage thresholds  $T_1, T_2$   
weighting schedule  $W^s$   
**Output:** optimized control points  $cp^*$   
**Initialize:** Step size  $\eta$ , iteration index  $i \leftarrow 0$

```

1 for  $i \leftarrow 0$  to  $T$  do
2   if  $i < T_1$  then
3     // Stage1: coarse
4      $E = W_{vis}^1 E_{geo} + E_{visual} + E_{aesthetic}$ 
5   else if  $i < T_2$  then
6     // Stage2: mid
7      $E = W_{feat}^2 E_{geo} + E_{visual} + E_{aesthetic}$ 
8   else
9     // Stage3: fine
10     $E = W_{feat}^3 E_{geo} + E_{visual} + E_{aesthetic}$ 
11    // -- gradient update via PyTorch AD
12     $\mathbf{g} \leftarrow \nabla_{cp} E$ ;
13     $cp^{i+1} \leftarrow cp^i - \eta \mathbf{g}$ ;
14 return  $cp^* = cp^T$ 

```

---

and the overall silhouette alignment score for the current curve set  $C$  is the mean IoU across all views:

$$Sil(C) = \frac{1}{N} \sum_{v=1}^N \text{IoU}^v. \quad (18)$$

For each wire  $c_i \in C$ , denoted by  $C \setminus c_i$  the set obtained after deleting  $c_i$ . At every iteration we remove the wire that causes the smallest drop in the alignment score:

$$C^* = \arg \min_{c_i \in C} [Sil(C) - Sil(C \setminus c_i)]. \quad (19)$$

Note that our procedure terminates once the number of remaining wires reaches a user-specified budget  $T$ . This contour-aware pruning discards visually redundant wires while preserving the global silhouette fidelity measured by the IoU metric.

## 4. Experiments

### 4.1. Results and Implementation Details

As illustrated in Fig. 5, our method achieves a consistent abstraction of the input 3D geometry. By leveraging a principled manifold-based surface fitting scheme combined with a contour-aware post-processing strategy, we are able to extract compact wire abstractions that preserve the structural essence of the shape. The first row in the Fig. 5 is the input point cloud. The second row is the intermediate result - the fitted coons patch. The third to fifth rows are our final results. And as shown in the second column of the Fig. 5, for models with genus greater than 0, we can get the beautiful wire art by using the template corresponding to the genus. The final wire art exhibits strong multi-view consistency and maintains high perceptual discriminability, effectively capturing both global topology and salient geometric features.

**Data preparation.** As described in Sec. 3.3, optimization is driven by the 3-D Chamfer distance. Each input mesh is therefore uniformly sampled to a point cloud of 4096 points and linearly normalized into the range  $[-1, 1]^3$ . Normals are estimated by a  $k=16$  PCA neighbourhood and used in the normal and concavity terms.

**Optimization hyper-parameters.** We employ plain gradient descent with a fixed step size  $\eta = 1.5 \times 10^{-3}$ . All experiments converge within 1500–2000 iterations. Training runs on a single AMD 9600X CPU paired with an NVIDIA 4060Ti GPU and requires roughly 5 minutes per model.

**Weights.** Tab. 4.1 lists the weights used for the three optimization stages. The *Silhouette Alignment* term is employed only during the coarse stage to facilitate a more stable and reliable initialization. In contrast, the *G1 Smoothness Energy* is activated exclusively in the final stage to enhance the aesthetic quality of the resulting wires—specifically, to ensure smoother transitions and to prevent overly sharp angles at patch boundaries.

Energy term	Stage 1	Stage 2	Stage 3
Chamfer, $E_{ch}$	1.00	1.00	1.00
Normal, $E_n$	0.02	0.02	0.02
Concavity Enhancement, $E_{ce}$	0.10	0.10	0.10
Silhouette Alignment, $E_{sil}$	0.20	0.00	0.00
Flatness, $E_{flat}$	2.00	2.00	2.00
Symmetry, $E_{sym}$	0.10	0.10	0.10
Smoothness, $E_{G1}$	0.00	0.00	0.01

**Table 1:** Stage-wise energy weights.

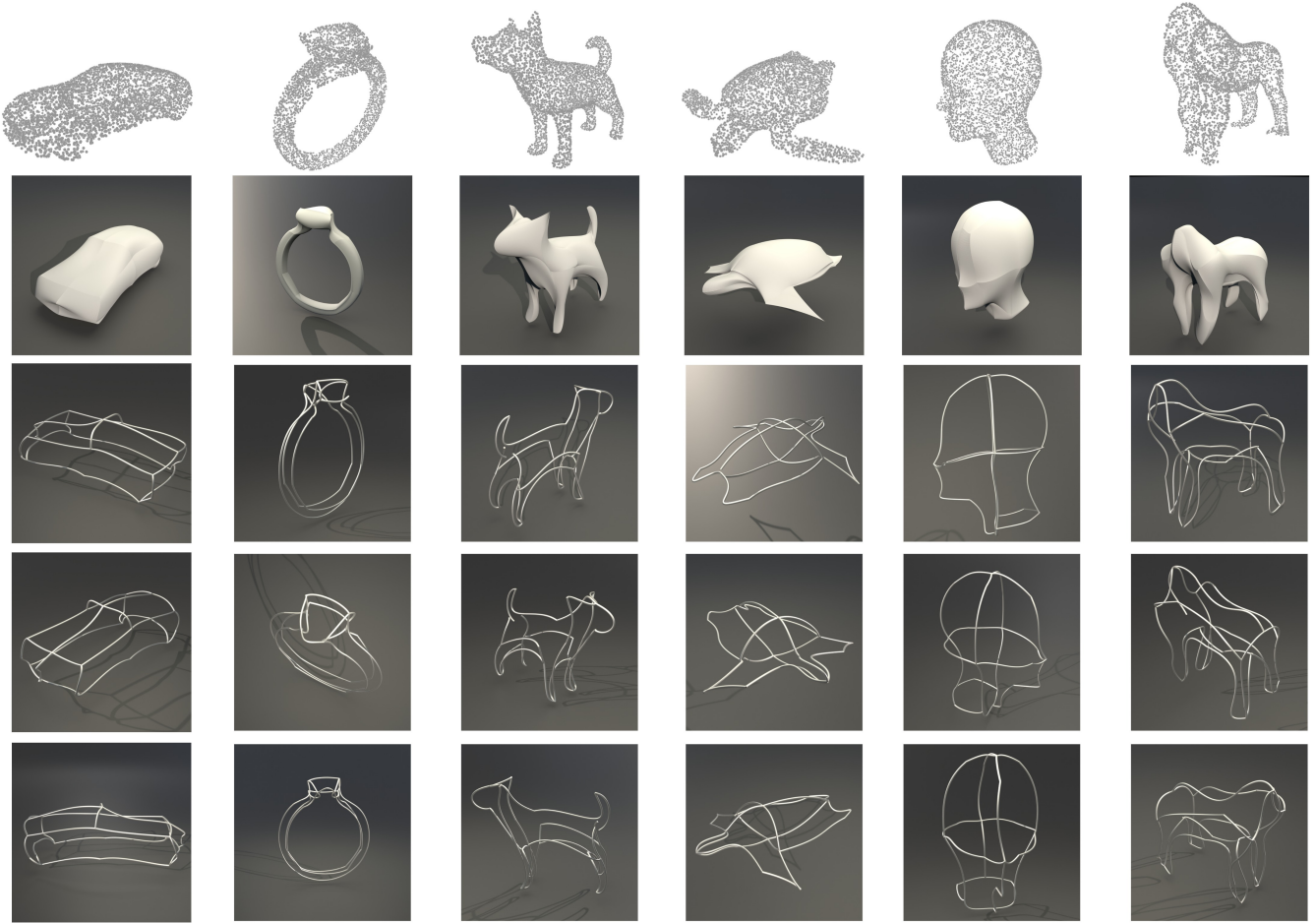
### 4.2. Comparisons

Before showing the comparison, we need to state that in principle we should compare with both Yeh et al. [YLHL22] and WireRoom [YXFH21]. However, due to practical challenges in reproducing the execution environment required by WireRoom [YXFH21], we were unable to include it in our evaluation.

Therefore, we benchmark our method against the recent wire art approach of Yeh et al. [YLHL22], which extracts wire art directly from 3-D meshes without requiring extensive user intervention. All evaluations are performed on six uniformly distributed camera views. Performance is quantified using (i) multiview silhouette Intersection-over-Union, capturing geometric consistency across viewpoints, and (ii) CLIP perceptual loss, which evaluates the semantic fidelity of the projected wires in image space. These complementary metrics jointly assess structural accuracy and high-level visual resemblance. The quantitative scores and visual comparisons are reported in Tab. 2 and Fig. 6, respectively.

### 4.3. Results from Methods with Different Stylizations

In this section, we qualitatively present the results produced by our method alongside those of Fab3D [TSBU24] and 3Doodle [CLPK24]. No quantitative comparisons are provided here for two main reasons. First, our method differs significantly in stylistic intent from them, making direct metric-based evaluation less



**Figure 5:** Our wire art model has different levels of complexity. The top row shows the input point clouds. The second row displays the intermediate fitted Coons patches. The bottom row presents our final wire art results rendered in Blender. Each sample is given the rendering results from three perspectives to demonstrate the view independence of our method.

Method	IoU (higher is better)		CLIP Loss ↓
	mean ↑	std ↓	mean ↓
Ours	<b>86.23</b>	<b>6.77</b>	<b>0.327</b>
Yeh <i>et al.</i>	77.03	17.27	0.271

**Table 2:** Quantitative comparison of multi-view IoU and CLIP perceptual loss.

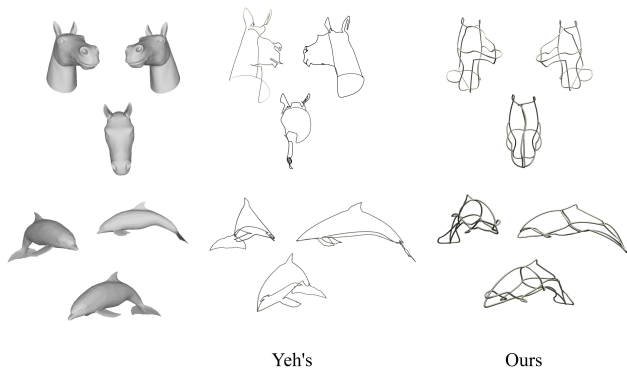
meaningful. Second, their approach involves 2D-to-3D reconstruction from multiple images, which is fundamentally different from ours.

As shown in Fig. 7, (a) show the result of 3Doodle [CLPK24], which takes 100 different views and text prompts as input. It can restore details well, but is not manufacturable. (b) shows the result of Fab3D [TSBU24]. Its design uses a single, simple wire, which is manufacturable but less effective for complex inputs due to the limited expressive power of a single wire. (c) is our result,

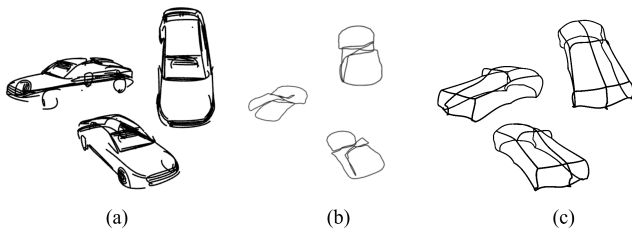
which strikes a balance between simplicity and expressiveness, and is manufacturable.

#### 4.4. Ablation Study

**Stage-wise Weighting Strategy.** We first evaluate how the proposed three-stage coarse-to-fine weighting scheme influences performance. Starting from a baseline without any weighting, we incrementally incorporate visual weighting (stage 1) and then both visual and feature-space weighting (stages 2–3). We compare the resulting wires at each stage to the original model to evaluate the impact of each weighting strategy. The qualitative differences are illustrated in Fig. 8. The orange boxes highlight the missing fine-grained structures in (a) and (b), particularly around the ear region. The blue boxes indicate overly sharp geometry at the foot in (a) and (b), where the model fails to accurately preserve the original shape. This analysis reveals how the coarse visual prior stabilizes early optimization, while feature-space weights refine high-frequency details in later stages.



**Figure 6:** Comparison with the results of Yeh et al. [YLHL22] from different viewpoints.



**Figure 7:** (a) 3Doodle's results. (b) Fab3D's results and (c) our results

**Energy Term Contribution.** Fig. 9 shows the effect of successively enabling the three groups of energy terms. To better illustrate the optimization dynamics, we extract and visualize intermediate results at 400, 800, and 1200 steps during the optimization process.

*Geometry Only.* With purely geometric energy terms, the optimization captures the coarse structure of the model. However, the convergence is relatively slow, and the resulting curves fail to accurately reflect key geometric features.

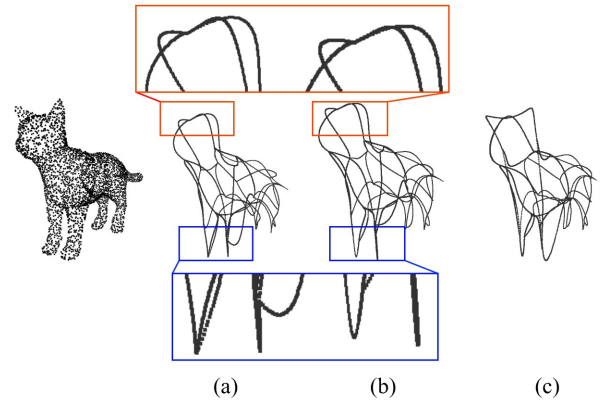
*Geometry + Visual.* Incorporating the multi-view silhouette term  $E_{\text{sil}}$  provides image space supervision, which stabilizes the early stages of optimization and significantly accelerates convergence. However, some curves still do not adequately represent the detailed geometry of the model.

*Full (Geometry + Visual + Aesthetic).* By further introducing aesthetic energy terms, the optimization produces well-structured curves that are perceptually coherent, evenly distributed, and better aligned with both the model geometry and visual appearance.

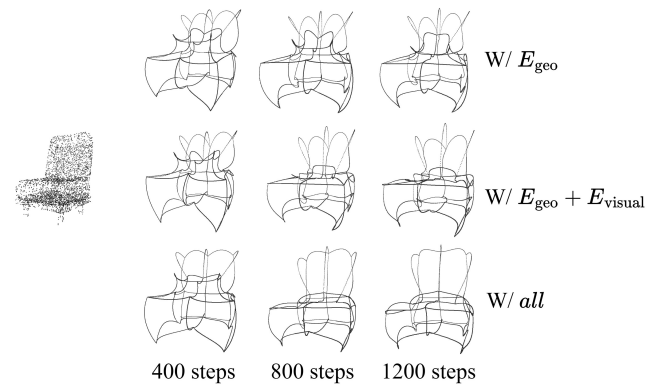
Overall, the progressive integration of visual and aesthetic energy terms significantly enhances both the convergence behavior and the perceptual quality of the resulting curves.

#### 4.5. Limitations and Future Works

As shown in Fig. 10, our method struggles when the object exhibits complex internal details that are difficult to abstract—specifically,



**Figure 8:** The ablation study of the weights. (a) Result without any weighting. (b) Result with visual weighting only. (c) Result with both visual and feature-space weighting.



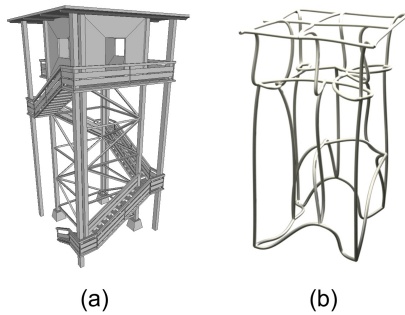
**Figure 9:** Ablation study of the energy terms. The first row shows results using only geometric energy terms. The second row incorporates both geometric and visual energy terms. The third row includes all energy terms, combining geometric, visual, and aesthetic constraints.

continuous spiral staircases and double-layered frame structures. For a Chamfer-distance-driven optimisation, such structurally intricate cases cannot be captured by a concise set of wires, making them a considerable challenge for our approach.

We plan to enrich the optimization with higher-dimensional, semantics-aware priors so that the system can reason not only about geometry and appearance but also about the functional meaning of regions. Incorporating such semantic cues—e.g., via learned shape embeddings or text-guided diffusion features—should enable the generation of multi-layered wire abstractions that better capture highly complex structures.

#### 5. Conclusion

This paper introduces a novel view-independent wire art modeling method with manifold representation. We formulate this task as a dual problem of manifold fitting rather than directly predicting wire



**Figure 10:** Failure case. (a) Input model. (b) The optimization result without post-processing.

structures, drawing inspiration from Bézier curves. Our manifold fitting-based method can produce appealing 3D wire art modeling results by considering 3D point cloud features and 2D projection features. Our method can also control the complexity of modeling results using different numbers of curves. Unlike previous computational methods [YXFH21, YLHL22], our method does not require complex user intervention. Extensive qualitative and quantitative studies demonstrate that our approach outperforms other methods regarding multi-view quality. Enhancing the aesthetic quality of wire art modeling is an open and challenging problem for subsequent researchers. Additionally, we observed that automatically generating a template with appropriate complexity may be more interesting than manually adjusting the template for complex models. Developing this algorithm could be our future work.

### Acknowledgement

This work is supported by the National Science and Technology Council, Taiwan under Grant 113-2221-E006-161-MY3 and Grant 114-2221-E-006-114-MY3.

### References

- [BHC21] BHATTACHARYYA P., HUANG C., CZARNECKI K.: SA-Det3D: self-attention based context-aware 3D object detection. In *IEEE International Conference on Computer Vision Workshops* (2021), pp. 3022–3031. 2
- [BWL\*23] BANGARU S. P., WU L., LI T.-M., MUNKBERG J., BERNSTEIN G., RAGAN-KELLEY J., DURAND F., LEFOHN A., HE Y.: Slang.d: Fast, modular and differentiable shader programming. *ACM Trans. Graph.* 42, 6 (2023). URL: <https://doi.org/10.1145/3618353>, doi:10.1145/3618353. 2
- [CLPK24] CHOI C., LEE J., PARK J., KIM Y. M.: 3Doodle: compact abstraction of objects with 3D strokes. *ACM Transactions on Graphics* 43, 4 (2024), 107:1–107:13. 1, 2, 3, 6, 7
- [CSKG17] CHARLES R. Q., SU H., KAICHUN M., GUIBAS L. J.: PointNet: deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 77–85. 2
- [DGGDV11] DE GOES F., GOLDENSTEIN S., DESBRUN M., VELHO L.: Technical section: Exoskeleton: Curve network abstraction for 3d shapes. *Comput. Graph.* 35, 1 (Feb. 2011), 112–121. URL: <https://doi.org/10.1016/j.cag.2010.11.012>, doi:10.1016/j.cag.2010.11.012. 2
- [EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. *ACM Trans. Graph.* 13, 1 (Jan. 1994), 43–72. URL: <https://doi.org/10.1145/174462.156635>, doi:10.1145/174462.156635. 4, 5
- [ILB15] IARUSSI E., LI W., BOUSSEAU A.: WrapIt: computer-assisted crafting of wire wrapped jewelry. *ACM Transactions on Graphics* 34, 6 (2015), 221:1–221:8. 2
- [JWZ\*18] JIANG M., WU Y., ZHAO T., ZHAO Z., LU C.: PointSIFT: a sift-like network module for 3D point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652* (2018), 1–10. 2
- [LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018). URL: <https://doi.org/10.1145/3272127.3275109>, doi:10.1145/3272127.3275109. 2
- [LCC\*18] LIU L., CHEN N., CEYLAN D., THEOBALT C., WANG W., MITRA N. J.: Curvifusion: Reconstructing thin structures from rgbd sequences. *ACM Trans. Graph.* 37, 6 (Dec. 2018), 218:1–218:12. URL: <http://doi.acm.org/10.1145/3272127.3275097>, doi:10.1145/3272127.3275097. 2
- [LCL\*17] LIU L., CEYLAN D., LIN C., WANG W., MITRA N. J.: Image-based reconstruction of wire art. *ACM Transactions on Graphics* 36, 4 (2017). 2
- [LD18] LE T., DUAN Y.: PointGrid: a deep network for 3D shape understanding. In *IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 9204–9214. 2
- [LFZ18] LIRA W., FU C.-W., ZHANG H.: Fabricable eulerian wires for 3d shape abstraction. *ACM Trans. Graph.* 37, 6 (Dec. 2018). URL: <https://doi.org/10.1145/3272127.3275049>, doi:10.1145/3272127.3275049. 2
- [LHK\*20] LAINE S., HELLSTEN J., KARRAS T., SEOL Y., LEHTINEN J., AILA T.: Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020). URL: <https://doi.org/10.1145/3414685.3417861>, doi:10.1145/3414685.3417861. 2
- [LLGRK20] LI T.-M., LUKÁČ M., GHARBI M., RAGAN-KELLEY J.: Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.* 39, 6 (2020). URL: <https://doi.org/10.1145/3414685.3417871>, doi:10.1145/3414685.3417871. 2
- [NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: a retargetable forward and inverse renderer. *ACM Trans. Graph.* 38, 6 (2019). URL: <https://doi.org/10.1145/3355089.3356498>. 2
- [QYZ\*24] QU Z., YANG L., ZHANG H., XIANG T., PANG K., SONG Y.-Z.: Wired Perspectives: multi-view wire art embraces generative ai. In *IEEE Conference on Computer Vision and Pattern Recognition* (2024), pp. 6149–6158. 2, 3
- [RRN\*20] RAVI N., REIZENSTEIN J., NOVOTNY D., GORDON T., LO W.-Y., JOHNSON J., GKIOXARI G.: Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501* (2020). 2, 4, 5
- [SBS21] SMIRNOV D., BESSMELTSEV M., SOLOMON J.: Learning manifold patch-based representations of man-made shapes. In *International Conference on Learning Representations* (2021), pp. 1–24. 2, 4, 5
- [TSBU24] TOJO K., SHAMIR A., BICKEL B., UMETANI N.: Fabricable 3d wire art. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, USA, 2024), SIGGRAPH '24, Association for Computing Machinery. URL: <https://doi.org/10.1145/3641519.3657453>, doi:10.1145/3641519.3657453. 1, 2, 3, 6
- [WA23] WORCHEL M., ALEXA M.: Differentiable rendering of parametric geometry. *ACM Trans. Graph.* 42, 6 (Dec. 2023). URL: <https://doi.org/10.1145/3618387>, doi:10.1145/3618387. 2
- [WLC\*20] WANG P., LIU L., CHEN N., CHU H.-K., THEOBALT C., WANG W.: Vid2Curve: simultaneous camera motion estimation and thin

- structure reconstruction from an rgb video. *ACM Transactions on Graphics* 39, 4 (2020), 132:1–132:12. 2
- [WZPB23] WU C., ZHENG J., PFROMMER J., BEYERER J.: Attention-based point cloud edge sampling. In *IEEE Conference on Computer Vision and Pattern Recognition* (2023), pp. 5333–5343. 2, 4
- [YBAF22] YANG Y., BARNES C., ADAMS A., FINKELSTEIN A.: A: autodiff for discontinuous programs - applied to shaders. *ACM Trans. Graph.* 41, 4 (2022). URL: <https://doi.org/10.1145/3528223.3530125>, doi:10.1145/3528223.3530125. 2
- [YLHL22] YEH C.-K., LE T.-N.-H., HOU Z.-Y., LEE T.-Y.: Generating virtual wire sculptural art from 3D models. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 2 (2022), 51:1–51:23. 1, 2, 3, 6, 8, 9
- [YXFH21] YANG Z., XU P., FU H., HUANG H.: WireRoom: model-guided explorative design of abstract wire art. *ACM Transactions on Graphics* 40, 4 (2021), 128:1–128:13. 1, 2, 3, 6, 9
- [ZCT16] ZEHNDER J., COROS S., THOMASZEWSKI B.: Designing structurally-sound ornamental curve networks. *ACM Trans. Graph.* 35, 4 (July 2016). URL: <https://doi.org/10.1145/2897824.2925888>, doi:10.1145/2897824.2925888. 2
- [ZWSW22] ZHANG C., WAN H., SHEN X., WU Z.: PatchFormer: an efficient point transformer with patch attention. In *IEEE Conference on Computer Vision and Pattern Recognition* (2022), pp. 11789–11798. 2
- [ZWZ\*19] ZHANG C., WU L., ZHENG C., GKIOULEKAS I., RAMAMOORTHY R., ZHAO S.: A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019). URL: <https://doi.org/10.1145/3355089.3356522>, doi:10.1145/3355089.3356522. 2