DOI: 10.1111/cgf.70323

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*

3D Character Reconstruction from Hand-drawn Model Sheets

Hyejeong Yoon Wonjong Jang Yoonha Hwang Seungyong Lee†

POSTECH, South Korea

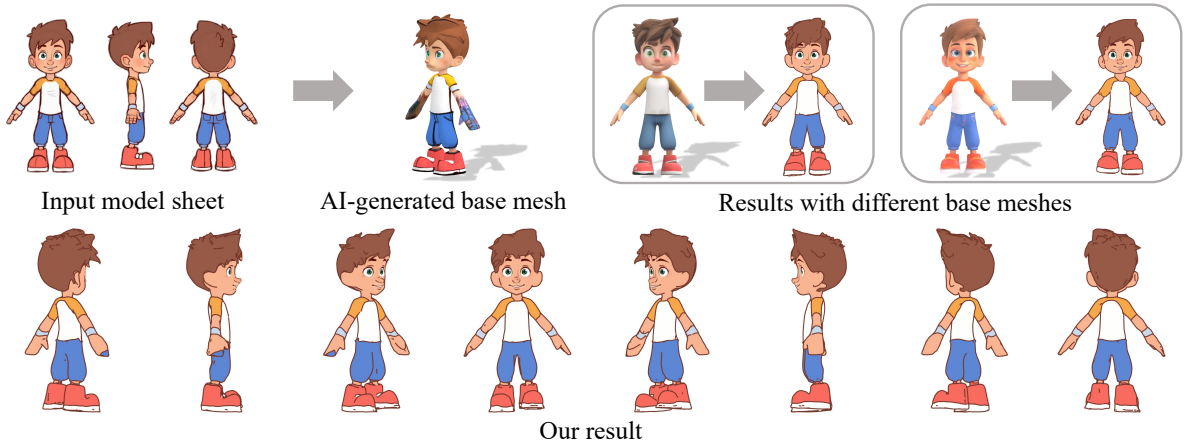

Figure 1: *Our method reconstructs 3D meshes from hand-drawn model sheets. Starting with a base mesh generated by a 3D generative model [XLX*25], we improve both geometry and appearance to faithfully reproduce the input style. Regardless of the 3D generative models [Dee25; LLC25] used to obtain the base mesh, our method produces consistent reconstruction results. Input image: ©LuigiL/DeviantArt, licensed under CC BY-NC-ND 3.0.*

Abstract

Hand-drawn model sheets are widely used in character design to define 3D shape and appearance through sparse multi-view drawings. Reconstructing 3D characters from such sparse inputs has traditionally been challenging due to insufficient visual information. Recently, 3D generative models have enabled automatic reconstruction of plausible 3D characters by learning from large-scale training data, but achieving reconstructions that accurately match the input model sheets remains limited. In this paper, we present a framework that leverages the power of a 3D generative model for initial reconstruction and enhances the output to faithfully reproduce input model sheets. For such faithful reconstruction, we must address two fundamental challenges: (1) the hand-drawn nature inherently introduces multi-view inconsistencies where the generated 3D geometry cannot perfectly align with all views, and (2) view-dependent line elements along geometry boundaries interfere with accurate texture reconstruction. To address these challenges, we optimize the geometry to minimize multi-view inconsistencies and introduce a deformable per-pixel camera ray representation that resolves residual discrepancies in cross-view correspondences. We also decompose drawings into three distinct layers of view-dependent lines, view-independent colors, and fine-detail decals to separately handle view-dependent and view-independent components for consistent cross-view reconstruction. Comprehensive experiments demonstrate that our method outperforms possible alternatives regardless of the choice of 3D generative model, while successfully preserving both artistic intent and visual fidelity of input model sheets.

CCS Concepts

• ***Computing methodologies*** → ***Mesh models;*** *Texturing; Non-photorealistic rendering; Image-based rendering;*

1. Introduction

Character model sheets are a fundamental tool for artists to efficiently define the 3D geometry and appearance of characters

† Corresponding author.

through multi-view drawings. Despite containing only a limited number of views, these carefully crafted drawings enable viewers to readily imagine the complete 3D shape and appearance of a character. This remarkable efficiency makes character model sheets the essential communication bridge between 2D character designers and 3D modeling artists, conveying meaningful 3D information about character design.

Automatically reconstructing 3D characters from these model sheets has traditionally been restricted due to sparse input views. Recently, the emergence of 3D generative models [XLX*25; HYY*25] has enabled plausible 3D character reconstruction from just a handful of drawings by leveraging learned priors from extensive training data. While these 3D generative models provide a significant breakthrough in handling sparse inputs, the generated 3D character models still exhibit notable visual discrepancies from the original drawings. The output geometry is often misaligned with the input views, and textures lose critical artistic details including line work, color choices, and shading styles that define the character's visual identity. Thus, our goal is to bridge this remaining gap by enhancing both the geometry and texture of the initial reconstruction to accurately match the input model sheets and preserve the artistic intent of the original drawings, as shown in Figure 1.

Achieving faithful 3D reconstruction from model sheets presents two main challenges. First, hand-drawn model sheets inherently contain *multi-view inconsistencies*, where no single 3D geometry can perfectly align with all views [ZQG*20]. This geometric ambiguity impedes a consistent 3D reconstruction that respects the appearance from all provided views. Second, *view-dependent line elements* along silhouettes and internal boundaries interfere with accurate texture reconstruction [ZXLF24]. These lines, essential to the structural definition of character drawings, change position and visibility with viewing angle. This view-dependency causes direct texture projection from multi-view drawings to produce undesirable line artifacts.

To better handle these challenges, we preprocess the input model sheets using a drawing decomposition step that converts hand-drawn inputs into a reconstruction-friendly representation. The goal of our drawing decomposition is to remove view-dependent components from the input drawings and to enforce consistent color labels for corresponding parts across different drawings. Specifically, we decompose each drawing into three layers: view-dependent line elements, view-independent flat colors, and fine-detail decals. By isolating view-dependent lines and assigning consistent color labels across views, this decomposition provides clean appearance information and cross-view consistency cues that are later exploited during geometry and texture reconstruction.

In this paper, we propose a framework that leverages 3D generative models for robust initial reconstruction, and present novel techniques to faithfully reproduce input model sheets by addressing multi-view inconsistencies and view-dependent line elements. Starting from the generated model, we optimize the geometry to minimize multi-view misalignment. To handle residual discrepancies that cannot be resolved through geometric optimization alone, we introduce a deformable per-pixel camera ray representation. The cross-view consistency cues established by drawing decomposition help guide this optimization by providing reliable part-level correspondences across drawings. By allowing each pixel's camera ray to deform independently, our approach achieves correct cross-view correspondences, even when the hand-drawn nature of model sheets makes perfect geometric consistency impossible.

Once we establish reliable correspondences through our deformable per-pixel ray representation, the next challenge lies in reconstructing textures from the input drawings. To avoid artifacts caused by view-dependent line elements, we reconstruct mesh textures using only the view-independent flat color layer obtained from drawing decomposition. Fine-detail decals, which represent small visual elements like facial features and decorative patterns, are captured separately as colored point clouds to maintain their precise appearance. Through this layered reconstruction approach, we ensure that the characteristic appearance of model sheets is well represented in the final 3D model.

In summary, our main contributions are:

- A novel framework that enhances 3D generative model outputs to faithfully reproduce input model sheets through geometry and texture enhancements.
- A drawing decomposition strategy that removes view-dependent elements and enforces cross-view color consistency, providing reliable cues for downstream reconstruction.
- A deformable per-pixel camera ray representation that enables accurate cross-view correspondences even in the presence of inherent multi-view inconsistencies in hand-drawn model sheets.
- A layered appearance reconstruction approach that separately handles view-dependent lines, view-independent colors, and fine-detail decals, ensuring accurate preservation of the appearance of input drawings.

2. Related Work

2.1. 3D Reconstruction from Sketches and Drawings

Sketch-based modeling represents one of the earliest attempts to derive 3D shapes from hand-drawn input [IMT99; SWSJ07; NISA07]. This approach established a new workflow for creating 3D geometry, but modeling is still challenging to deliberately manage view changes to realize the intended 3D shape. Motivated by modeling directly from 2D drawings, a line of work reconstructs 3D shapes from a single sketch. SmoothSketch [KH06] interpreted occlusions in sketches to predict their depth relations. Li et al. [LPL*18] employ CNNs to predict flow fields followed with normal and depth maps. Given the inherent ill-posedness of lifting sparse line drawings to 3D, most approaches are learning-based as deep learning has matured [GYS*22; ZPW*23]. To mitigate ambiguity in the single-view setting, several methods incorporate explicit annotations [LPL*17; XCS*14; LPBM20; PMKB23]. Due to the inherent uncertainty of single-view sketches, prior work has largely treated them as a medium for concept expression rather than accurate 3D specification.

To further constrain geometry, some methods accept multi-view sketches as input. Lun et al. [LGK*17] take front and side sketches to estimate multi-view normals and depths and fuse them into a point cloud. In a similar approach with image translation, Han et al. [HMLZ20] reconstruct shapes from predicted attenuance images. Delanoy et al. [DAI*18] and Kim et al. [KHW*22] refine

reconstructions progressively or iteratively to integrate multiple sketches. Zhou et al. [ZLY*23] introduce an interactive modeling system that allows users to refine rendered sketches from arbitrary views. Chen et al. [CYW*24] train a NeRF from sketches, and SAniHead [DHF*20] performs mesh deformation and refinement for 3D reconstruction. In practice, many of these methods rely on two orthogonal views or staged refinement to avoid multi-view inconsistency, and they report robustness only on synthetically generated sketches. A few studies directly address hand-drawn sketches rather than synthetic inputs. Zhong et al. [ZQG*20] observe that even professional sketches exhibit ambiguity and stroke misalignment. Wang et al. [WLY*22] point out distortions as a key challenging point of hand-drawn sketches. Xu et al. [XHY*22] further extensively analyzed challenges of free-hand sketches.

Drawings, which have colors in addition to sketch lines, have also been used for 3D reconstruction. Buchanan et al. [BMD13] estimate a 3D shape from a single concept art via skeleton extraction. Ink-and-Ray [SKČ*14] constructs bas-relief geometry using layered depth and inflation. Monster Mash [DSC*20] and CreatureShop [ZYC*22] further employ inflation operations to recover complete textured 3D meshes. RABIT [LCD*23] leverages a parametric model for biped cartoon characters, while methods for children's drawings preserve stylistic cues via 2.5D reconstruction [SZL*23; SHY25]. DrawingSpinUp [ZXLF24] uses AI-generated meshes and removes silhouette lines to produce visually pleasing textures. While these approaches yield plausible results, they typically rely on a single view, limiting users' ability to control shape across multiple orthographic views, which is a key requirement in character design.

2.2. Image-to-3D Generation

Recent progress in 3D generation has been accelerated by advances of 2D generative models. Early pipelines optimized NeRF-style radiance fields or 3D Gaussians directly under 2D priors [MRP*23; WLW*23; TRZ*23], but the Janus artifact underscored that strong multi-view consistency is essential for reliable shape recovery [SWY*23]. To enforce such consistency from a single view, a prominent line of work first synthesizes multi-view consistent images and then reconstructs 3D from them [SWY*23; LLZ*23; SCZ*23; LGL*24]. As a later work, Fancy123 [YLT*25] deforms 2D images and the 3D mesh to improve multi-view consistency and reduce ghosting artifacts.

In parallel, the approach that directly outputs 3D meshes has gained attention [GSW*22; CCJJ23; QMH*23]. Large-scale models, which are trained on massive datasets, encode the input image into a latent space and decode a 3D representation in the form of radiance field, Gaussians, or mesh [HZG*23; WLZ*24; LZL*25; XLX*25; HYY*25]. Despite their robustness to unknown cameras and in-the-wild inputs, encoding and decoding through a latent bottleneck tends to suppress delicate visual details, reducing fidelity to fine-scale geometric and appearance.

While most generation methods target photorealistic or rendered-style images as input, methods for character drawings have been proposed. PANiC-3D [CZS*23] reconstructs character heads from anime-style single drawings. CharacterGen [PZG*24] creates 3D character models in canonical poses from arbitrarily posed single character drawings. CharNeRF [CCRB24] targets concept art but handles only drawings from three fixed views. Toon3D [WPM*24] addresses distortions in scene-level drawings using user-annotated correspondences and ARAP warping. CoNR [LHH22] synthesizes neural-rendered images from anime character sheets. While these methods push towards 3D reconstruction from character drawings, they address distinct problem settings compared to our work, characterized by restricted input views or their specific input domains and output representations.

2.3. Texture Reconstruction

Given calibrated multi-view images and a reconstructed surface, classical pipelines recover seamless textures along two complementary axes: (i) photometric color optimization on the surface or UV domain to suppress seams and exposure drift, and (ii) optimizing where to sample colors from the images via per-texel (or per-face) source-view assignment. Early *seamless mosaicing* formulations select source views and seam boundaries by minimizing a global energy to reduce visible seams [LI07; GWO*10]. At large scales, Waechter et al. [WMG14] propose a unified pipeline integrating view selection and color adjustment, enabling robust texturing of large real-world reconstructions. Jeon et al. [JJKL16] optimize texture coordinates to reduce multi-view photometric inconsistency. To address geometry-texture misalignments, Zhou and Koltun [ZK14] and Fu et al. [FYY*18] employ non-rigid correction with control grids, while Bi et al. [BKR17] propose patch-based optimization to manage larger errors. Recently, Knodt et al. [KPWG23] jointly optimize the UV layout and the baking process to lower distortion and preserve high-frequency detail during color optimization. However, as these methods are tailored for photographic data with subtle misalignments, they are ineffective for the severe and irregular distortions in our hand-drawn inputs.

Recent methods synthesize textures for a given mesh under image guidance. TEXTure [RMA*23] presents a text-guided texture generation method that leverages a pretrained diffusion prior, and it also supports reference-image-guided texture transfer. StyleMesh [HJN22] performs style-image-guided texture stylization on reconstructed meshes by jointly optimizing a global texture with multiple views. TextureDreamer [YHK*24] transfers textures from a sparse set of input photographs to a target mesh using geometry-aware diffusion. Direct UV map synthesis has also been explored: TEXGen [YYG*24] trains a large diffusion model to generate high-resolution UV textures conditioned on a single reference image or text. For artwork inputs, AlignTex [ZXW*25] combines image alignment and UV-space inpainting to produce pixel-precise and multi-view consistent textures. These image-guided texture generation methods utilize learned priors to inject information beyond the observed pixels, where direct projections are incomplete. Both classical and learning-based approaches form the basis of texture reconstruction, and we compare representative methods in our evaluation.

3. Overview

Our framework reconstructs 3D characters from model sheets through four key components: base mesh generation, drawing de-

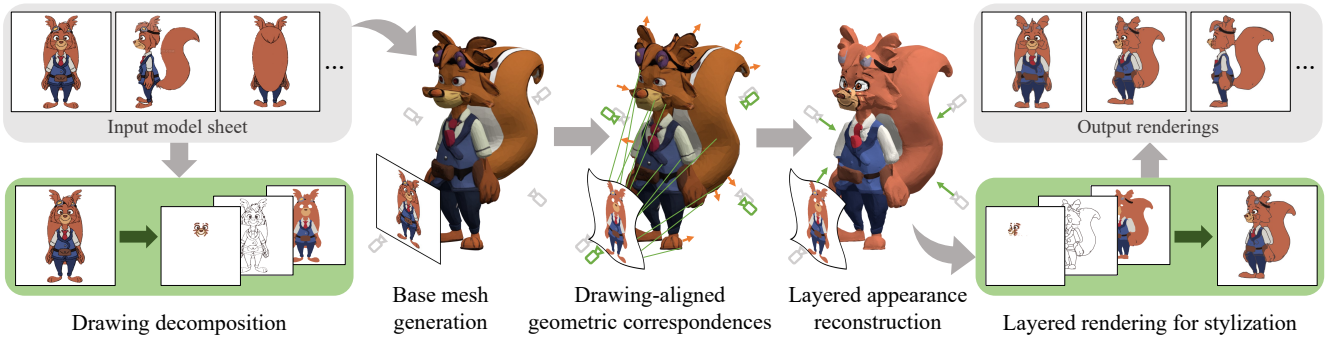

Figure 2: *Pipeline overview. Our method takes a hand-drawn model sheet comprising multi-view drawings as the input and produces a faithful 3D colored mesh. Our method also offers non-photorealistic layered rendering of the final 3D model to reproduce the input style. Input image: ©Aiyanimation/DeviantArt, CC BY-NC-ND 3.0.*

composition, drawing-aligned correspondences, and layered appearance reconstruction. In this section, we provide a brief overview of each component. We first introduce base mesh generation as the initial step of our pipeline. We then describe drawing decomposition as the next step, which will be presented in detail in Section 4. For our main contributions of drawing-aligned correspondences and layered appearance reconstruction, we will elaborate them in Sections 5 and 6, respectively. The overall pipeline is illustrated in Figure 2.

Base mesh generation We initialize our pipeline by generating a 3D base character mesh from the input model sheet using TRELLIS [XLX*25]; as we demonstrate in Section 7.5, any 3D generative model can be employed for base mesh generation. However, regardless of which 3D generative model is employed, the resulting geometry still suffers from misalignment with the input model sheet. The generated appearance also exhibits noticeable discrepancies from the original model sheets.

Drawing decomposition View-dependent elements in drawings, such as lines along silhouettes, change their positions and visibilities across views, making multi-view consistent 3D reconstruction challenging. To address these view-dependent effects, we decompose input drawings into three distinct layers: *view-dependent lines*, *fine-detail decals*, and *view-independent colors*. While the line and color layers are processed automatically, the extraction of fine-detail decals is facilitated by coarse user-defined masks to ensure precise isolation of intricate features.

Drawing-aligned geometric correspondences Establishing accurate alignment between geometry and cameras is crucial for accurate 3D reconstruction. We begin by calibrating cameras through rigid transformation to achieve initial alignment between the base mesh and input drawings. We then deform the mesh to minimize multi-view inconsistencies. To address residual misalignment, we introduce a deformable per-pixel camera ray representation that enables robust cross-view correspondence construction.

Layered appearance reconstruction Using the decomposed drawing layers, we reconstruct the 3D character appearance by separately handling each layer. View-dependent lines extracted from the drawings are excluded from texture reconstruction to eliminate view-dependent artifacts, then reintroduced during rendering through line drawing techniques to preserve the original sketch style. Fine-detail decals are reconstructed as colored point clouds attached on the base mesh surface to preserve their intricate details. View-independent colors, containing essential appearance information, are reconstructed using part-based mesh coloring through color palette and graph-cut optimization.

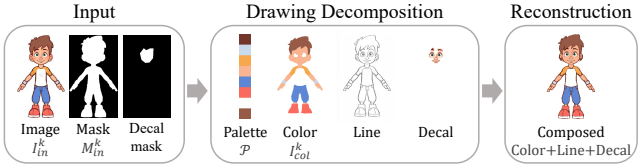

Figure 3: *Overview of the drawing decomposition step and the reconstruction result. Given an input drawing along with its mask and a user-defined decal mask, our method extracts a color palette and decomposes the drawing into color, line, and decal layers. Compositing these layers reconstructs an image that is visually consistent with the input, demonstrating the effectiveness of the decomposition.*

4. Drawing Decomposition

The goal of our drawing decomposition is to remove view-dependent components from the input drawings and to enforce consistent color labels for corresponding parts across different drawings. Character model sheets typically define a consistent color palette that captures character identity and employ flat coloring to maintain visual consistency across drawings. Based on these properties, we treat the flat color layer as view-independent and consistent across views. To this end, given input drawings $\{I^k_{\text{in}}\}$, their binary masks $\{M^k_{\text{in}}\}$, and user-defined decal masks, our method extracts the flat color layer from each drawing by first removing sketch lines and then separating fine-detail decals. The overview of the drawing decomposition pipeline is illustrated in Figure 3.

View-dependent lines View-dependent lines define character structure along silhouettes and internal boundaries, including occlusion lines and suggestive contours [DFRS03], naturally varying with viewpoint. While traditional edge detection methods [Can86; SF*68] can identify these features, they typically produce noisy and fragmented results unsuitable for clean line extraction. We therefore employ a state-of-the-art learning-based line drawing generation method, *Informative Drawings* [CDI22] that produces cleaner and more coherent line structures. To eliminate residual noise and ensure line continuity, we apply morphological operations as a post-processing step, yielding refined line drawings that accurately capture the essential structural elements of the input drawings.

View-independent colors After removing the view-dependent lines, we obtain view-independent color regions that maintain consistency across views, providing a reliable foundation for both establishing cross-view correspondences and reconstructing character appearance. We further refine these regions by removing color noise and inpainting areas left by line removal, yielding coherent flat color regions.

Specifically, we first construct a global color palette by clustering dominant colors from all input drawings in the CIELAB color space using HDBSCAN [CMS13], followed by iterative merging of small nearby clusters until convergence. This process results in a palette

$$\mathcal{P} = \{c_1, \ldots, c_L\}, \tag{1}$$

where L is the number of resulting clusters and each c_ℓ represents a representative RGB color shared across views.

Using this palette, we decompose each input drawing into view-independent flat color regions. Given an input RGB image I^k_{in}, we assign a discrete color label to each pixel by solving a per-drawing graph cut optimization [BVZ02], yielding a label image

$$Y^k = \text{GraphCut}(I^k_{\text{in}}; \mathcal{P}), \tag{2}$$

where $Y^k \in \{1, \ldots, L\}^{H \times W}$ encodes the flat color layer for view k. The corresponding RGB flat color image is then reconstructed by mapping each label to its palette color,

$$I^k_{\text{col}}(p) = c_{Y^k(p)}, \tag{3}$$

resulting in a clean, view-independent color image I^k_{col}. This flat view-independent color layer serves as supervision for image-based loss functions in subsequent optimization stages.

Fine-detail decals Fine-detail decals encompass intricate elements like facial features and decorative patterns, which commonly break view consistency due to their complexity. To preserve these critical features, we separately extract decals using user-defined masks specifying fine-detail areas, where each decal is extracted from a single user-specified view. We identify the dominant color label from the established palette for each decal region, and then employ a chroma-key method to remove background colors. This approach allows users to roughly specify masks without requiring precise boundaries. The non-background decals seamlessly integrate with the base color rendering, as the base color palette is also extracted from the drawing.

5. Drawing-aligned Geometric Correspondences

Establishing accurate geometric correspondences between the base mesh and input drawings requires addressing inherent multi-view inconsistencies in hand-drawn model sheets. We propose a three-stage alignment strategy: rigid camera calibration for initial view parameters, mesh deformation to minimize multi-view inconsistencies, and per-pixel camera ray deformation to handle residual distortions. This progressive refinement ensures robust cross-view correspondences between 3D geometry and 2D drawings, enabling faithful appearance reconstruction in Section 6.

5.1. Initial Camera Setup

We calibrate camera extrinsic parameters through rigid alignment, optimizing per-view scale and translation while maintaining fixed mesh geometry. We adopt an orthographic projection model and assume yaw-aligned input views, as commonly assumed in hand-drawn model sheets, which ensures consistent proportions across views and avoids perspective distortion. We initialize rotation matrices $\{\mathbf{R_k}\}_{k=1}^K$ from user-specified yaw angles and optimize scale variables $\{s_k\}_{k=1}^K$ and translation variables $\{\mathbf{t_k}\}_{k=1}^K$ for K input views.

With the fixed base textured mesh $\mathcal{M}$, we render the corresponding binary mask and the RGB color images under the view transform $[s_k\mathbf{R_k}|\mathbf{t_k}]$:

$$\begin{aligned} R^k_{\text{mask}} &= \text{Render}_{\text{mask}}(\mathcal{M}; [s_k\mathbf{R_k} \mid \mathbf{t_k}]), \\ R^k_{\text{col}} &= \text{Render}_{\text{col}}(\mathcal{M}; [s_k\mathbf{R_k} \mid \mathbf{t_k}]). \end{aligned} \tag{4}$$

The alignment objective minimizes discrepancies between rendered and input drawings:

$$\mathcal{L}_{align} = \sum_{k=1}^{K} \left\| R^k_{\text{mask}} - M^k_{\text{in}} \right\|_2^2 + \sum_{k=1}^{K} \left\| R^k_{\text{col}} - I^k_{\text{in}} \right\|_2^2. \tag{5}$$

We employ differentiable rendering [LHK*20] for gradient computation, optimizing s_k and $\mathbf{t_k}$ through gradient-based minimization of $\mathcal{L}_{align}$.

5.2. Mesh Deformation for Multi-view Consistency

To minimize view-dependent distortions across input drawings, we deform the base mesh using a Jacobian field-based approach [AGK*22; GAG*23; YKKS24; JAC*21] that produces smooth deformations. To prevent singularities, we preprocess the base mesh to guarantee two-manifold topology before deformation. We optimize the Jacobian fields while fixing the view matrices, guided by the mask and color losses defined in Eq. (5). At each iteration, vertex positions are recomputed from the updated Jacobian fields. This mesh deformation step effectively adapts the geometry to reduce multi-view inconsistencies, yielding well-aligned geometry for subsequent geometric correspondence construction.

5.3. Deformable Per-pixel Camera Ray

While geometric alignment reduces multi-view inconsistencies, hand-drawn model sheets inherently contain distortions that cannot be fully resolved through geometry deformation alone. To es-

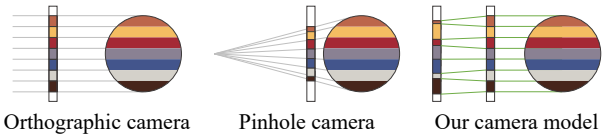

Figure 4: *Comparison of camera models projecting a 3D mesh (colored circle) onto the image plane (colored bar). Orthographic and pinhole models cannot adequately capture the irregular drawing distortions. In contrast, our deformable per-pixel camera ray decouples projection into two stages: the mesh is first orthographically projected (right bar), and then the ray origins are shifted to produce the final image-plane projection (left bar), providing high flexibility to handle multi-view inconsistent distortions inherent in hand-drawn model sheets.*

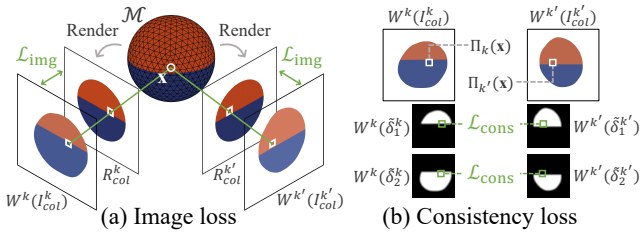

Figure 5: *Illustration of two loss functions for per-pixel camera ray deformation. (a) Image loss measures the difference between rendered mesh images and inversely warped view-independent color layers. During rendering the mesh, 3D points are projected onto each view through deformable per-pixel camera rays. (b) Consistency loss is computed by comparing colors at the projected pixels of identical 3D points across different views, using inversely warped label-wise masks.*

tablish meaningful cross-view correspondences despite these misalignments, we propose *deformable per-pixel camera ray* to better align the geometry with input drawings.

Traditional camera models [GD00; KB06; TE07; MR07] assume fixed or limited camera viewpoints, making them unsuitable for capturing the irregular perspective distortions in drawings. In contrast, ray-based camera models [GN01; GN05; SLPS20], where each pixel is represented as a ray with an origin and a direction, provide the flexibility needed for non-photographic content. We combine the ray-based camera model with orthographic projection: rays within each view maintain parallel directions but originate from displaced positions that have been shifted from the regular pixel grid. This formulation effectively decouples our camera model into two components, orthographic projection and image-space warping, providing high flexibility with low computational cost, as shown in Figure 4.

Under our orthographic assumption, displacing a ray's origin is equivalent to reassigning it to a different pixel's ray on the image plane. We therefore model per-pixel ray origins as a displacement field over the image domain, eliminating explicit ray casting during optimization. This formulation requires only a single rendered image per view, making the optimization process efficient and tractable while maintaining sufficient flexibility to capture local drawing distortions.

We parameterize the displacement field using B-spline free-form deformation [BS18; Ros16]. For each view, the displacement at pixel coordinate $(u,v)^\top$ is computed as:

$$\mathbf{D}^k(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} B_i(u)\,B_j(v)\,\mathbf{C}_{ij}^k, \tag{6}$$

where $\{B_i(u)\}_{i=0}^{n-1}$ and $\{B_j(v)\}_{j=0}^{n-1}$ are cubic B-spline basis functions, and $\{\mathbf{C}_{ij}^k \in \mathbb{R}^2\}_{i,j=0}^{n-1}$ are control points on an $n \times n$ grid. We denote inverse warping of image I at view k under the displacement field $\mathbf{D}^k$ by the operator W:

$$\left[W^k(I)\right](p) = I\left(p + \mathbf{D}^{k^{-1}}(p)\right). \tag{7}$$

To leverage the textured base mesh as guidance, we render it from each rigidly aligned view using orthographic projection. We optimize inverse warping to align the rendered images with the view-independent color layers. The image loss is defined as:

$$\mathcal{L}_{\text{img}} = \sum_{k=1}^{K} \left\| W^k(I_{\text{col}}^k) - R_{\text{col}}^k \right\|_2^2. \tag{8}$$

This formulation aligns each view to the shared mesh, thereby coupling the per-view deformations and implicitly enforcing multi-view consistency.

While the image loss provides indirect cross-view alignment through geometry, explicit correspondence constraints further improve multi-view consistency. We introduce a consistency loss that exploits the color labels from view-independent color layers. This loss ensures that 3D surface points, when projected through inverse warping fields, maintain consistent color labels across multiple views.

For a set of sampled surface points $\mathcal{S}$ and view pairs $\mathcal{V}$ constructed from adjacent views sorted by rotation angle, the consistency loss is:

$$\mathcal{L}_{\text{cons}} = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}} \frac{1}{|\mathcal{V}(\mathbf{x})|} \sum_{(k,k')\in\mathcal{V}(\mathbf{x})} \Psi(Y^k, Y^{k'}, \mathbf{x}), \tag{9}$$

where $\mathcal{V}(\mathbf{x}) \subset \mathcal{V}$ denotes the subset of view pairs (k,k') in which surface point $\mathbf{x}$ is visible in both views. The label-consistency function Ψ compares color labels across adjacent views using softly extended label masks after inverse warping.

Given the flat color label image Y^k, we define the discrete label indicator

$$\delta_l^k(p) = \mathbb{1}[Y^k(p) = l] \in \{0,1\}, \tag{10}$$

and its soft extension

$$\tilde{\delta}_l^k(p) = \max\left(0, 1 - \frac{d_l^k(p)}{\tau}\right) \in [0,1], \tag{11}$$

where $d_l^k(p)$ measures the distance to the nearest pixel with label l, and τ controls the extent of the soft transition. The label-

consistency function for a surface point $\mathbf{x}$ is then defined as

$$\Psi(Y^k, Y^{k'}, \mathbf{x}) = \sum_{l=1}^{L} \left| W^k(\tilde{\delta}_l^k)(\Pi_k(\mathbf{x})) - W^{k'}(\tilde{\delta}_l^{k'})(\Pi_{k'}(\mathbf{x})) \right|, \quad (12)$$

where the projection $\Pi_k(\mathbf{x})$ maps point $\mathbf{x}$ to its image coordinates in view k. This formulation encourages label consistency and ensures that the loss increases smoothly with boundary misalignment, preserving discrete color regions while remaining differentiable.

The computation process for image and consistency loss terms is illustrated in Figure 5. The complete objective function combines these terms with regularization:

$$\mathcal{L}_{\text{camera}} = \mathcal{L}_{\text{img}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \quad (13)$$

where $\mathcal{L}_{\text{reg}}$ penalizes excessive deformations through norms of control points and the warping field Jacobian. Refer to the supplementary document for the full definition of $\mathcal{L}_{\text{reg}}$.

6. Layered Appearance Reconstruction

We structure our rendering pipeline into three layers, mirroring the drawing decomposition described in Section 4. This section details the 3D reconstruction methods for color and decal layer. We then present a layered rendering approach, including construction of the line layer, that faithfully reproduces the stylized appearance of the input model sheet. This layered approach enables faithful reproduction of the input model sheet's stylized appearance while maintaining rendering flexibility across novel viewpoints.

6.1. Palette-based Mesh Coloring

A view-independent color layer represents a set of colors that remain consistent across multi-view drawings. To extract this layer, we first perform color histogram analysis for global color palette construction from input drawings. This palette enables the identification of consistent regions across multiple views and facilitates noise removal in flat color areas. Subsequently, we segment each drawing into pixel clusters sharing the same colors, considering only non-line pixels. We employ graph cut optimization [BVZ02] that ensures clean color boundaries by assigning corresponding color labels to all pixels, including removed sketch lines.

Given the global color palette, we propose a part-based mesh coloring approach that assigns discrete palette colors to mesh triangles. Rather than recovering per-pixel textures as in traditional methods [LI07; WMG14], our approach estimates color boundaries through robust graph-cut optimization [BVZ02]. While per-pixel optimization is sensitive to view inconsistency, boundary optimization absorbs pixel-level noise and boundary position uncertainty, consolidating them into consistent part regions across views.

We first compute an initial color for each triangle by aggregating pixel colors from its projections across all views. For triangle t, the initial color a_t is computed as:

$$a_t = \frac{1}{|K|} \sum_{k=1}^{K} \frac{1}{|\Omega_t^k|} \sum_{p \in \Omega_t^k} I_{\text{col}}^k(p), \quad (14)$$

where Ω_t^k represents the set of pixels covered by triangle t's projection in view k.

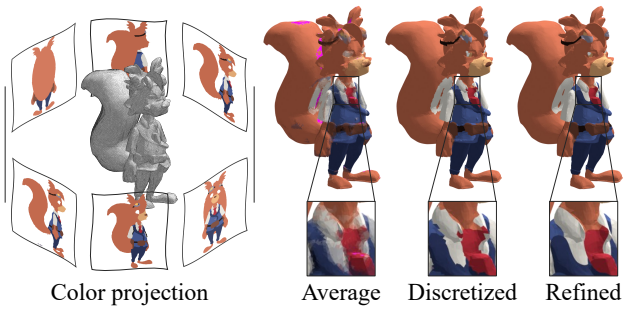

Figure 6: *Results of the palette-based mesh coloring process. View-independent color layers are projected onto the mesh using deformable per-pixel camera rays. Initial triangle-wise colors are computed by averaging projected pixels (magenta indicates triangles invisible from all views). These colors are then discretized to palette colors through graph-cut optimization. Finally, boundaries are refined through smoothing to produce cleaner part-wise colorization.*

We formulate palette color assignment as a discrete labeling problem optimized via graph cuts [VD]. Our energy function combines unary and pairwise terms:

$$E(q_t) = \sum_t U(t, q_t) + \lambda \sum_{(t,t')} P(t, t'), \quad (15)$$

where q_t denotes the palette label assigned to triangle t, and λ controls the regularization strength. The unary term $U(t,q) = \|a_t - c_q\|_2^2$ measures color similarity between the initial triangle color and palette colors. The pairwise term $P(t,t') = w_{t,t'} \cdot \mathbb{1}[q_t \neq q_{t'}]$ enforces label consistency between adjacent triangles while preserving important boundaries. Refer to the supplementary document for the formulation of edge weight $w_{t,t'}$.

This formulation naturally produces clean part boundaries aligned with both drawing lines and geometric features. The graph-cut optimization effectively suppresses noise and artifacts from view inconsistencies. The discrete color assignment inherently matches the abstraction level of flat-colored inputs, and it explicitly defines part boundaries, which enables rendering them as 2D lines in the later layered rendering step.

Boundary refinement The triangle-wise coloring can produce jagged boundaries that deviate from the smooth color boundaries presented in the original drawings. We address this through a refinement step that regularizes the boundaries while maintaining their alignment with the underlying surface geometry. We begin by applying graph-based label propagation to smooth out color assignments near boundaries, using majority voting among neighboring triangles to reduce label noise. Small outlier clusters are also identified and removed when they contain fewer than 20 triangles.

To achieve smooth boundary curves, we extract edge sequences along color discontinuities and treat them as 3D polylines. We then apply iterative Laplacian smoothing [Tau95], where each vertex is moved toward the average position of its neighbors along the chain. This smoothing is applied for a small number of iterations to mit-

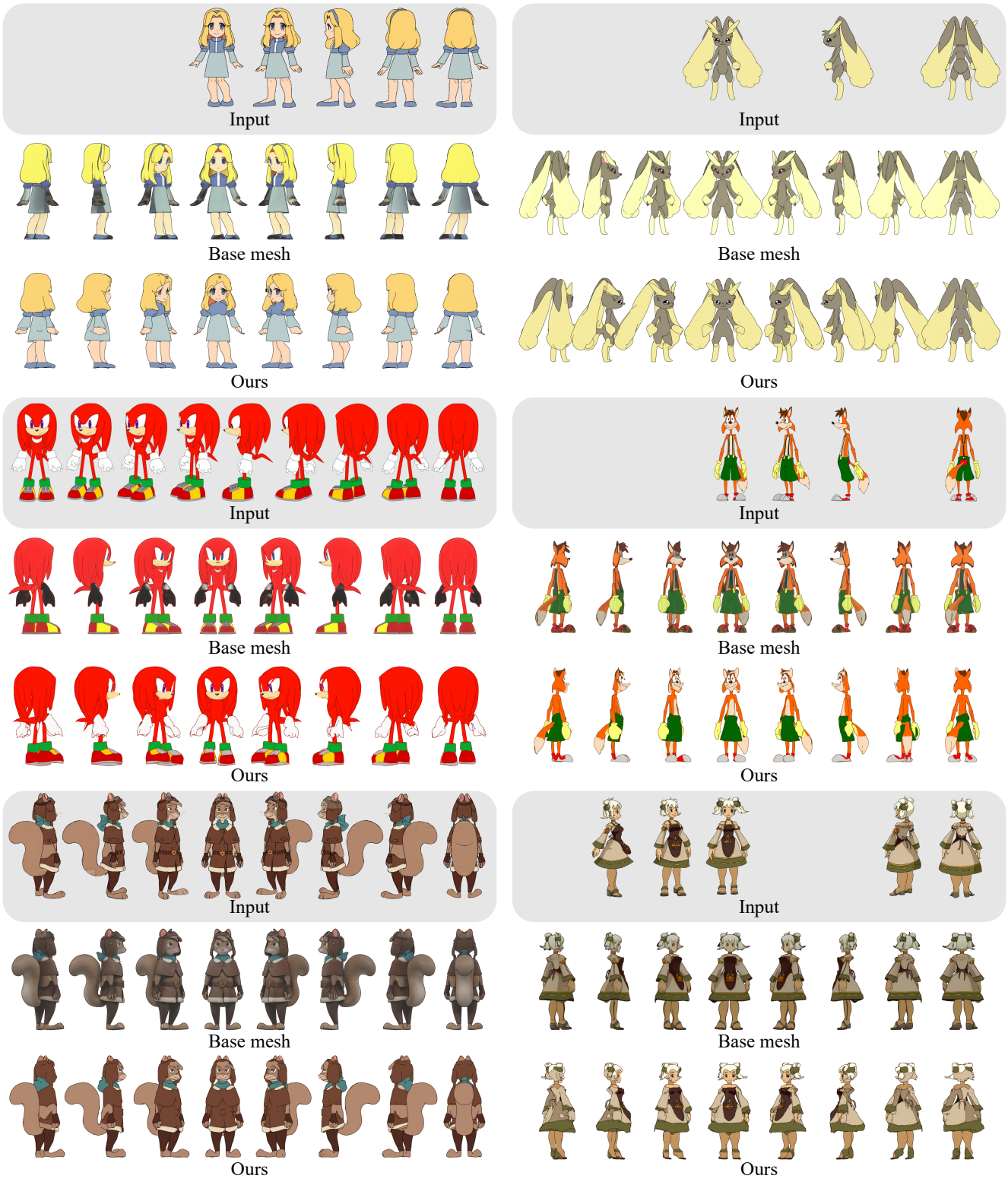

Figure 7: *Comparison with base mesh. Compared to the base mesh generated by TRELLIS [XLX*25], our results more accurately reproduce input colors and expressive details, such as hair silhouette and facial expressions. For the bottom-left example (5), the base mesh is generated from preprocessed images obtained by using Gemini [Dee25] to introduce shading information. Input images (left to right, top to bottom): ©Crystal-Ribbon/DevianArt (1, 3), ©MarwanGreenCritter/DevianArt (2), ©Atrox-C/DevianArt (4), ©Aiyanimation/DevianArt (5), ©Pepperistia/DevianArt (6); CC BY-NC-ND 3.0 (1–5); CC BY-NC-SA 3.0 (6).*

igate boundary irregularities without significantly deviating from the input drawings. After each smoothing iteration, the displaced vertices are projected back onto the mesh surface to maintain geometric consistency. This process produces visually pleasing 3D color boundaries that accurately represent the smooth color boundaries in the input drawings, as shown in Figure 6.

6.2. Fine-detail Decal Reconstruction

We lift the decal pixels obtained from Section 4 into colored 3D points using an extended depth map rendered from the mesh at the corresponding view. To mitigate errors near object silhouettes, the depth map is extended by filling background pixels via nearest-neighbor propagation, ensuring spatially consistent depth assignments.

6.3. Layered Rendering for Stylization

We structure our rendering process to mimic the layered composition of hand-drawn model sheets, comprising line, color, and decal layers. For the line layer, we apply non-photorealistic rendering technique [Com24] to draw lines on silhouettes and contours from the mesh geometry, with edges at color boundaries additionally rendered as lines. The color layer renders the mesh with flat diffuse colors without shading or illumination effects, producing an albedo-like representation that captures the part-based coloring characteristic in hand-drawn model sheets. For decals, we render them as point clouds where each point inherits visibility from its nearest mesh triangle. Notably, image-space warping fields are not applied during rendering, as they are designed primarily to compensate for inherent distortions in hand-drawn inputs. The final output is generated by compositing the color, line, and decal layers in sequence. This hierarchical composition faithfully reproduces the stylized appearance of the original hand-drawn model sheets.

7. Experiments

We validate our method on real-world character model sheets collected from online sources under Creative Commons licenses. Our test set was assembled by querying for *model sheet*, *character sheet*, and *turnaround sheet* among related terms. From each collected image, we manually filter out text, logos, and other non-character elements. We focused on hand-drawn model sheets, excluding heavily shaded artwork outside the scope of our method.

For base mesh generation, we primarily employed TRELLIS [XLX*25]. When TRELLIS failed to generate 3D base meshes due to insufficient 3D cues in hand-drawn inputs, we preprocessed the images using Gemini [Dee25] to introduce shading information and enhance resolution, enabling TRELLIS to better handle those challenging inputs.

All experiments were conducted on a single NVIDIA RTX 4090 GPU. Processing time scales linearly with the number of input views, requiring approximately 1 minute for three-view inputs and 3 minutes for eight-view inputs. Refer to the supplementary document for further implementation details including hyperparameters.

7.1. Results

Comparison with base mesh We present experimental results on several examples in Figure 7, showing the input drawings, the base meshes generated by TRELLIS [XLX*25], and the final results produced by our method. While the base meshes capture semantic similarity to the input drawings, they remains limited in accurately reflecting the colors and expressive details in the input drawings. In contrast, our method better aligns geometry with the input, enabling accurate color reproduction and the recovery of facial expressions through decal point clouds.

Step-by-step intermediate results We visualize intermediate results throughout our framework in Figure 8. The initial misalignments between the base mesh and the input drawings are progressively resolved via mesh deformation and camera ray optimization. The base mesh is first deformed under the guidance of the input drawings. The per-pixel camera rays are then optimized to establish exact correspondences between the base mesh and drawings. These accurate correspondences enable consistent color projection across views, resulting in final renderings that faithfully reproduce the input drawings.

7.2. Comparison

We compare our method against existing texture reconstruction and generation approaches in Figure 9. All methods used identical input drawings and base meshes generated by TRELLIS [XLX*25]. For MVS-Texturing, we provided the camera extrinsic parameters estimated during our initial camera setup stage, in which camera poses were rigidly aligned. For fair comparison, we rendered diffuse color and line layers for the two comparison methods in a manner similar to our layered rendering method.

MVS-Texturing [WMG14] represents traditional multi-view texture reconstruction, employing view selection and color blending to minimize seam artifacts. While robust for photorealistic inputs, it suffers from significant artifacts when applied to hand-drawn model sheets, as the method cannot distinguish between view-dependent and independent components. The view-dependent lines become baked into the texture, creating inconsistent line artifacts across the mesh surface that disrupt the intended appearance.

TEXTure [RMA*23] leverages diffusion models for texture generation from reference images without requiring camera parameters. Although it successfully transfers the overall style and color palette, the method struggles to preserve fine details and precise color boundaries. The generated textures often exhibit blurred details and color bleeding at region boundaries, failing to maintain the sharp, clean characteristic of hand-drawn model sheets.

In contrast, our method explicitly addresses the unique challenges of hand-drawn inputs through drawing decomposition and layered appearance reconstruction. By separating view-dependent lines and fine-detail decals from model sheets, we avoid line-baking artifacts while preserving the precise color boundaries and fine details that define character appearance. Our deformable camera ray representation ensures accurate texture projection despite multi-view inconsistency, resulting in textures that faithfully reproduce the original visual style.

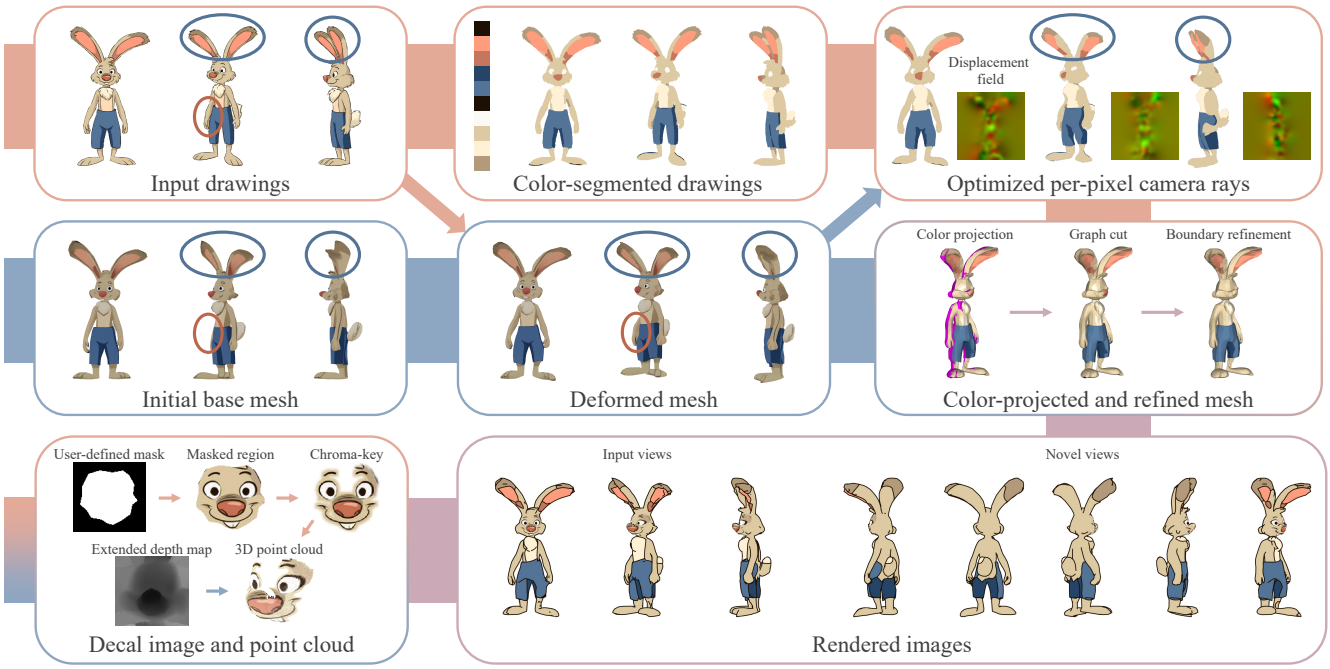

Figure 8: *Step-by-step intermediate results. The pink and blue streams indicate the flow of information from input drawings and the base mesh, respectively. The small arrows branching from each main stream denote optimization guidance. Magenta on the color-projected mesh indicates triangles that are invisible in all input drawing. Input image: ©tanya_buka/DeviantArt, CC BY-NC-ND 3.0.*

Table 1: *User study results: mean scores and p-values.*

Metric	Base mesh	Ours	*p*-value
Appearance fidelity	2.99	4.18	< 0.001
Geometric alignment	3.53	3.94	< 0.05
Novel view consistency	3.30	3.95	< 0.001

7.3. User Study and Professional Feedback

To validate our method's visual quality and practical utility, we conducted a subjective user study and gathered feedback from professional artists.

User study We conducted a user study with 30 participants to compare our method against the base mesh generated by TRELLIS [XLX*25]. The study was designed as a blinded A/B test using 10 character examples presented in this paper. For each example, participants were shown the original input model sheet alongside the base mesh and our result in a randomized order. To ensure a comprehensive assessment, we displayed rendered images from three input views and one novel view for each method.

Participants rated the results on a 5-point Likert scale across three metrics: (1) *appearance fidelity*, which evaluates the preservation of the original artistic style and unique character impression; (2) *geometric alignment*, which assesses the similarity of silhouettes and shapes to the input; and (3) *novel view consistency*, which measures how natural and plausible the character appears in viewpoints not present in the input.

We report the mean scores along with p-values derived from a paired t-test in Table 1. These quantitative results demonstrate that our reconstruction exhibits consistent improvements over the base meshes across all three metrics. Specifically, the most substantial improvements were observed in appearance fidelity and novel view consistency, indicating our method's superior capability in preserving the artistic intent. Geometric alignment also showed statistically significant improvement, although the margin was comparatively smaller than other metrics. See the supplementary document for the detailed distribution of user ratings.

Professional feedback To assess the practical value of our framework, we conducted in-depth interviews with four professional character artists, including one 2D artist, two 3D artists, and one artist designing both 2D and 3D characters. We presented a side-by-side comparison of the base meshes generated by TRELLIS [XLX*25] and our 3D reconstruction results, asking them to evaluate the outcomes based on prioritized qualities in production and potential utility within their workflows.

Artists identified distinct trade-off between geometric stability and artistic fidelity. They acknowledged that the base mesh frequently alters the character's impression, resulting in a generic look that deviates from the input identity, although it generally exhibits more realistic proportions and stable 3D forms. In contrast, they emphasized that our method successfully preserves the unique *feel* and design intent of the original drawings.

Notably, the artists identified the primary utility of our results as *3D visual blueprints* for concept verification. Given the context,

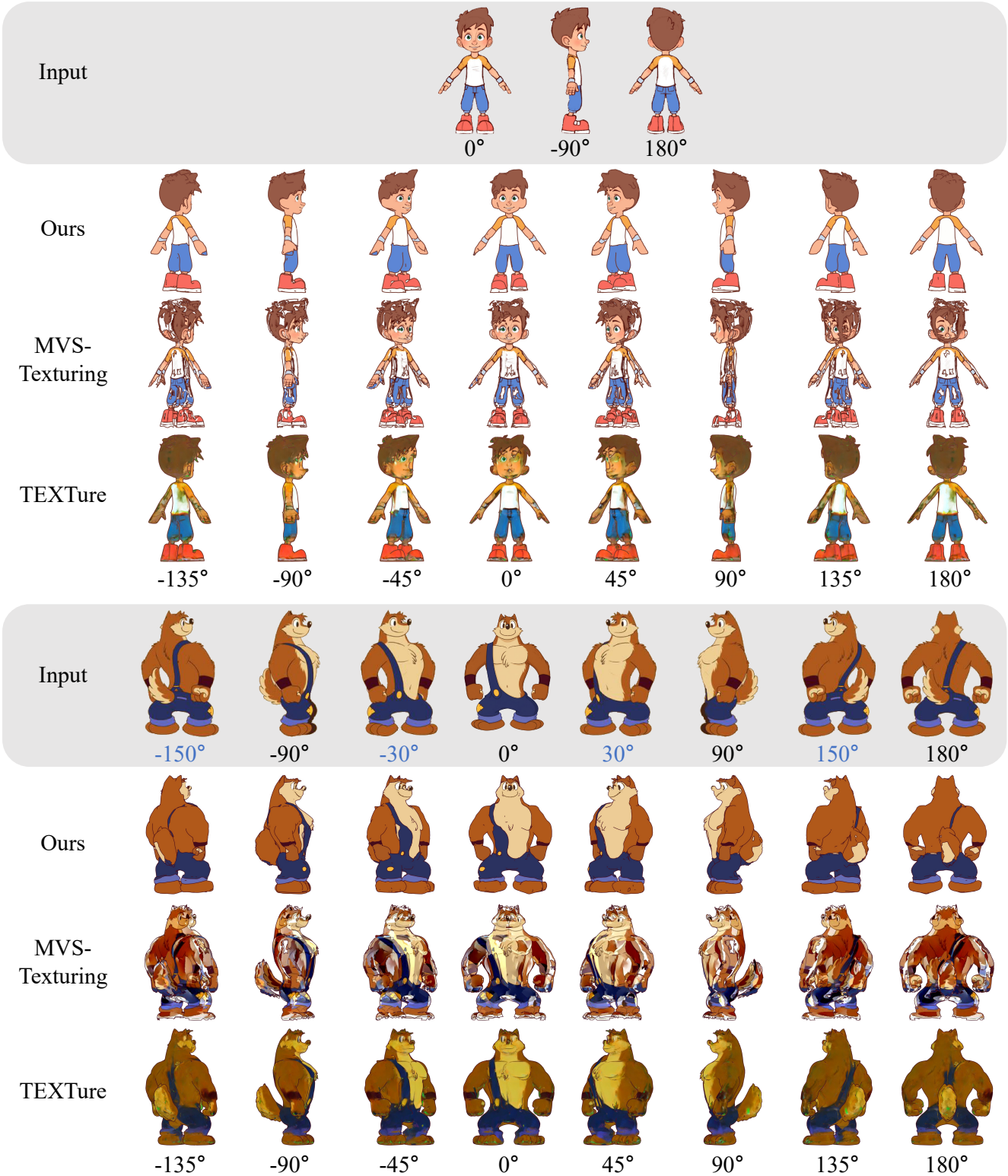

Figure 9: *Comparison with texture reconstruction and generation methods. Our method faithfully reconstructs input model sheets, while MVS-Texturing [WMG14] and TEXTure [RMA*23] fail to preserve details. Some input angles are highlighted in blue as they do not match the test view angles in the corresponding column. Input images: ©LuigiL/DeviantArt, CC BY-NC-ND 3.0, ©Piti Yindee/Wikimedia Commons, Creative Commons CC0 License.*

Figure 10: *Effect of mesh deformation. Mesh deformation significantly improves alignment with the input drawings, particularly evident in the head silhouette. Input image: ©residentevilffs/DeviantArt, CC BY-NC-SA 3.0.*

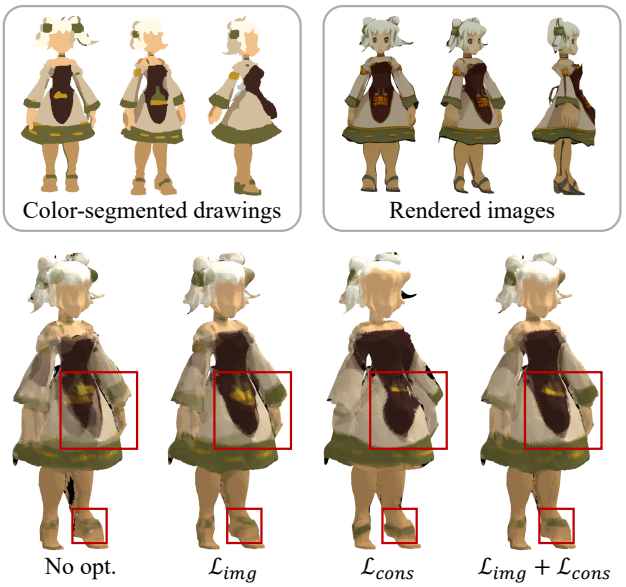

Figure 11: *Effect of deformable per-pixel camera ray optimization. Without camera ray optimization, highly inconsistent input images produce ghosting artifacts on the colored mesh caused by misaligned color-segmented drawings. Optimization with the image loss $\mathcal{L}_{img}$ improves alignment using the rendered images of the base mesh, but results in blurred color boundaries. The consistency loss $\mathcal{L}_{cons}$ alone sharpens boundaries but introduces undesirable distortions. Our method using both $\mathcal{L}_{img}$ and $\mathcal{L}_{cons}$ achieves well-aligned mesh colors with clean, sharp boundaries.*

they prioritized the preservation of character impression and stylistic fidelity over mesh topology and surface smoothness. They highlighted that our method's high source fidelity effectively bridges the communication gap between 2D and 3D departments, allowing teams to visualize and verify 3D concepts that align with the original vision before committing to expensive modeling resources.

7.4. Ablation Study

Mesh deformation We demonstrate the impact of mesh deformation on reconstruction quality in Figure 10. Even without mesh deformation, our deformable per-pixel camera ray successfully establishes reliable correspondences and produces plausible appearance reconstructions. However, incorporating mesh deformation achieves better geometric alignment between the 3D model and input drawings, particularly improving silhouette accuracy and overall shape fidelity across views.

Objective functions for deformable per-pixel camera ray We validate the effect of individual losses in our deformable per-pixel camera ray optimization. Figure 11 shows texture projection results using deformable per-pixel camera ray optimized under different loss configurations. We omit boundary refinement in this experiment to clearly show the impact of each loss.

Without deformable per-pixel camera ray optimization, standard orthographic projection produces severe ghosting artifacts, with brown apron regions overlapping incorrectly due to the misalignment of color-segmented images. This artifact highlights the challenge of multi-view inconsistencies in hand-drawn model sheets. The image loss $\mathcal{L}_{\text{img}}$ reduces overlapping artifacts by minimizing differences between color-segmented drawings and rendered images of the base mesh, but still yields blurred color boundaries. On the other hand, optimization using only the consistency loss $\mathcal{L}_{\text{cons}}$ produces sharp color boundaries, but introduces undesirable geometric distortions that deviate from the input drawings. Our complete formulation combining both $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{cons}}$ achieves clean color boundaries while maintaining fidelity to the input drawings.

7.5. Analysis

Geometric alignment analysis To demonstrate the effect of each step on geometric alignment, we visualize pixel-wise differences between intermediate results and input drawings in Figure 12. We capture results at key steps: base mesh generation, initial camera setup, mesh deformation, and deformable per-pixel camera ray optimization. The geometric misalignment progressively decreases through each stage. The base mesh provides an initial 3D approximation, rigid camera alignment establishes proper view correspondence, mesh deformation corrects major shape discrepancies, and camera ray optimization resolves remaining local distortions. This progression validates our multi-stage approach, where each component addresses specific challenges of the geometric alignment.

We further investigate the role of deformable per-pixel camera ray optimization in addressing residual geometric inconsistencies. As shown in Figure 13, while the preceding mesh deformation aligns the global shape, local misalignments persist in high-frequency regions such as extremities and internal boundaries (highlighted in circles). The visualized displacement fields demonstrate that our method selectively targets these discrepancies by warping input views to match the shared 3D geometry. The error

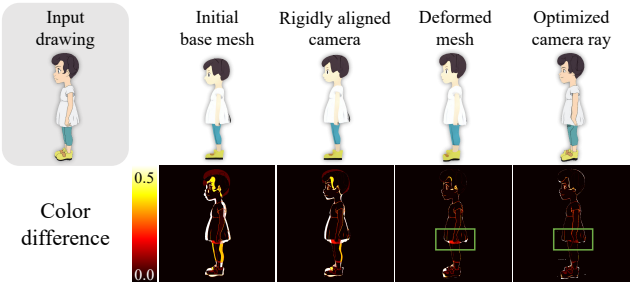

Figure 12: *Geometric alignment analysis. The top row shows the input drawing and results of each step: mesh rendering or warped drawing. The bottom row visualizes pixel-wise color differences between the result of each step and the input drawing. Input image: ©EliastRoven/DeviantArt, CC BY-NC-ND 3.0.*

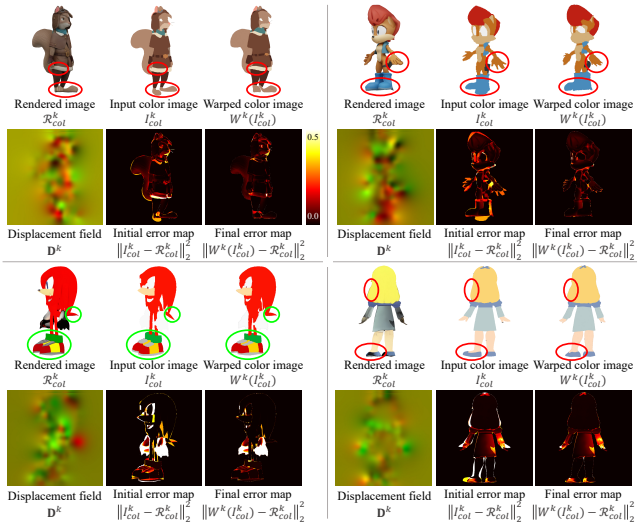

Figure 13: *Geometric alignment analysis of deformable per-pixel camera ray optimization. For each example, the top row compares the rendered base mesh $\mathcal{R}^k_{col}$, the input view-independent color layer I^k_{col}, and the warped result $W^k(I^k_{col})$. The bottom row presents the optimized displacement fields $\mathbf{D}^k$ and the corresponding L_2 error maps before and after optimization. Circles highlight regions with significant initial misalignments, which are resolved in the final results. All error maps share the same color scale.*

maps confirm the reduction in projection residuals, demonstrating that our optimization effectively resolves these local inconsistencies to achieve faithful geometric alignment.

Robustness to 3D generative model selection To demonstrate that our pipeline is agnostic to the choice of base mesh generation method, we evaluate our approach using three different 3D generative models: TRELLIS [XLX*25], TRELLIS with Gemini [Dee25] preprocessing for enhanced shading, and the commercial application Meshy [LLC25].

As shown in Figures 1 and 14, despite significant visual differences in the initial base meshes, our method consistently produces high-quality reconstructions that faithfully preserve the input model sheets. This consistency across diverse initialization conditions validates the robustness of our drawing-aligned geometric correspondences and layered appearance reconstruction approach.

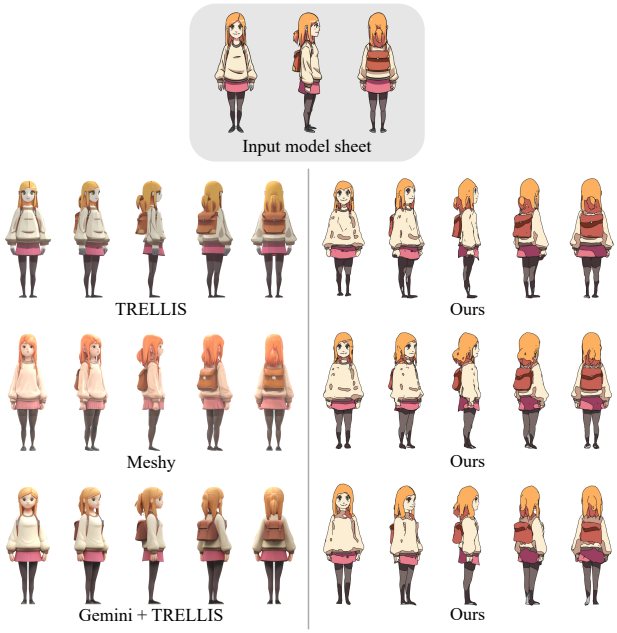

Figure 14: *Robustness to the base mesh. Top: input model sheet. Each row corresponds to a different base mesh generation method (TRELLIS [XLX*25], Meshy [LLC25], and TRELLIS [XLX*25] with preprocessed images using Gemini [Dee25] to introduce shading information). Our framework reconstructs consistent 3D characters regardless of the choice of 3D generative models. Input image: ©easy-ramos/DeviantArt, CC BY-NC-ND 3.0.*

Limitations Our method has a few limitations as shown in Figure 15. First, the reconstruction of fine-detail elements relies on user-specified masks to identify regions requiring separate processing. Without these masks, fine details are either treated as view-independent colors and undergo part-based coloring, or regarded as sketch lines and suppressed for coloring, which leads to detail loss and color bleeding along region boundaries. Second, the sparsity of input views inherently limits reconstruction quality in occluded regions. Areas not visible in any input view must be inferred through interpolation during texture projection, potentially causing color bleeding and inconsistent appearance. Additionally, our method assumes flat-colored model sheets and may not generalize to heavily shaded or painted styles. The drawing decomposition step specifically targets line art with discrete color regions, limiting applicability to other artistic styles. Future work could explore adaptive decomposition strategies to handle a broader range of inputs.

Beyond these specific artifacts, our framework faces broader geometric constraints. First, the training bias of the generative base

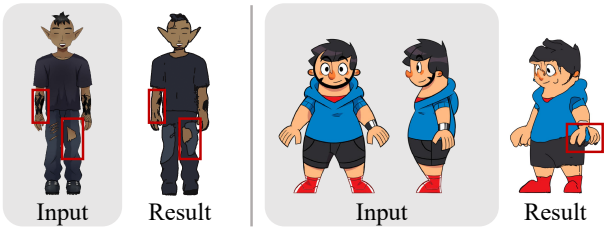

Figure 15: *Limitations. Left: Missing masks for fine-detail decals cause the details to merge with surrounding colors. Right: Input view sparsity produces color bleeding in occluded regions. Input images: ©GSMinerva/DeviantArt, ©Torogoz/DeviantArt, CC BY-NC-ND 3.0.*

model [XLX*25] limits generalization; while robust for human-like characters, it sometimes yields planar, non-volumetric outputs for non-humanoid forms (e.g., ducks, birds, cloud shapes), which are incompatible with our refinement process. Second, prioritizing strict 2D alignment over smoothness creates a trade-off, leading to local geometric irregularities such as jagged edges. Integrating a post-deformation remeshing stage would help recover production-ready topology while maintaining the achieved alignment.

8. Conclusion

We presented a novel framework for reconstructing faithful 3D characters from hand-drawn model sheets that addresses two fundamental challenges: multi-view inconsistency inherent in hand-drawings and view-dependent lines that hinder accurate texture reconstruction. Our approach introduces a deformable per-pixel camera ray representation that enables robust cross-view correspondences despite geometric inconsistencies, and a three-layer decomposition strategy that separates view-dependent lines, view-independent colors, and fine-detail decals for clean appearance reconstruction. Comprehensive experiments demonstrate that our framework consistently outperforms existing texture reconstruction and generation methods across diverse 3D generative models, successfully maintaining both geometric accuracy and visual fidelity of input model sheets. This work bridges 2D character design and 3D modeling workflows, with practical potential applications for animation and game industries.

Acknowledgments

We thank anonymous reviewers for their valuable feedback. This work was supported by NRF grants (RS-2025-02216257, RS-2024-00451947) and IITP grants (RS-2022-II220290, RS-2024-00437866) funded by the Korean government (MSIT).

References

[AGK*22] AIGERMAN, NOAM, GUPTA, KUNAL, KIM, VLADIMIR G, et al. "Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes". *arXiv preprint arXiv:2205.02904* (2022) 5.

[BKR17] BI, SAI, KALANTARI, NIMA KHADEMI, and RAMAMOORTHI, RAVI. "Patch-based optimization for image-based texture mapping." *ACM Trans. Graph.* 36.4 (2017), 106–1 3.

[BMD13] BUCHANAN, PHILIP, MUKUNDAN, RAMAKRISHNAN, and DOGGETT, MICHAEL. "Automatic single-view character model reconstruction". *Proceedings of the international symposium on sketch-based interfaces and modeling*. 2013, 5–14 3.

[BS18] BECK, JOHANNES and STILLER, CHRISTOPH. "Generalized B-spline camera model". *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, 2137–2142 6.

[BVZ02] BOYKOV, YURI, VEKSLER, OLGA, and ZABIH, RAMIN. "Fast approximate energy minimization via graph cuts". *IEEE Transactions on pattern analysis and machine intelligence* 23.11 (2002), 1222–1239 5, 7.

[Can86] CANNY, JOHN. "A computational approach to edge detection". *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698 5.

[CCJJ23] CHEN, RUI, CHEN, YONGWEI, JIAO, NINGXIN, and JIA, KUI. "Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation". *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, 22246–22256 3.

[CCRB24] CHU, EDDY, CHEN, YIYANG, RAISSI, CHEDY, and BHOJAN, ANAND. "CharNeRF: 3D character generation from concept art". *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*. IEEE. 2024, 185–194 3.

[CDI22] CHAN, CAROLINE, DURAND, FRÉDO, and ISOLA, PHILLIP. "Learning to generate line drawings that convey geometry and semantics". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 7915–7925 5.

[CMS13] CAMPELLO, RICARDO JGB, MOULAVI, DAVOUD, and SANDER, JÖRG. "Density-based clustering based on hierarchical density estimates". *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2013, 160–172 5.

[Com24] COMMUNITY, BLENDER ONLINE. *Blender – a 3D Modelling and Rendering Package*. Version 4.3. 2024. URL: http://www.blender.org 9.

[CYW*24] CHEN, MINGLIN, YUAN, WEIHAO, WANG, YUKUN, et al. "Sketch2NeRF: Multi-view Sketch-guided Text-to-3D Generation". *CoRR* (2024) 3.

[CZS*23] CHEN, SHUHONG, ZHANG, KEVIN, SHI, YICHUN, et al. "Panic-3d: Stylized single-view 3d reconstruction from portraits of anime characters". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 21068–21077 3.

[DAI*18] DELANOY, JOHANNA, AUBRY, MATHIEU, ISOLA, PHILLIP, et al. "3d sketching using multi-view deep volumetric prediction". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.1 (2018), 1–22 2.

[Dee25] DEEPMIND, GOOGLE. *Gemini 2.5 Flash*. Online; Release notes. 2025. URL: https://deepmind.google/models/gemini/ 1, 8, 9, 13.

[DFRS03] DECARLO, DOUG, FINKELSTEIN, ADAM, RUSINKIEWICZ, SZYMON, and SANTELLA, ANTHONY. "Suggestive contours for conveying shape". *ACM Trans. Graph.* 22.3 (July 2003), 848–855. ISSN: 0730-0301. DOI: 10.1145/882262.882354. URL: https://doi.org/10.1145/882262.882354 5.

[DHF*20] DU, DONG, HAN, XIAOGUANG, FU, HONGBO, et al. "SAniHead: Sketching animal-like 3D character heads using a view-surface collaborative mesh generative network". *IEEE Transactions on Visualization and Computer Graphics* 28.6 (2020), 2415–2429 3.

[DSC*20] DVOROŽŇÁK, MAREK, SỲKORA, DANIEL, CURTIS, CASSIDY, et al. "Monster mash: a single-view approach to casual 3D modeling and animation". *ACM Transactions on Graphics (ToG)* 39.6 (2020), 1–12 3.

[FYY*18] FU, YANPING, YAN, QINGAN, YANG, LONG, et al. "Texture mapping for 3d reconstruction with rgb-d sensor". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 4645–4653 3.

[GAG*23] GAO, WILLIAM, AIGERMAN, NOAM, GROUEIX, THIBAULT, et al. "Textdeformer: Geometry manipulation using text guidance". *ACM SIGGRAPH 2023 conference proceedings*. 2023, 1–11 5.

[GD00] GEYER, CHRISTOPHER and DANIILIDIS, KOSTAS. "A unifying theory for central panoramic systems and practical implications". *European conference on computer vision*. Springer. 2000, 445–461 6.

[GN01] GROSSBERG, MICHAEL D and NAYAR, SHREE K. "A general imaging model and a method for finding its parameters". *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE. 2001, 108–115 6.

[GN05] GROSSBERG, MICHAEL D and NAYAR, SHREE K. "The raxel imaging model and ray-based calibration". *International Journal of Computer Vision* 61.2 (2005), 119–137 6.

[GSW*22] GAO, JUN, SHEN, TIANCHANG, WANG, ZIAN, et al. "Get3d: A generative model of high quality 3d textured shapes learned from images". *Advances in neural information processing systems* 35 (2022), 31841–31854 3.

[GWO*10] GAL, RAN, WEXLER, YONATAN, OFEK, EYAL, et al. "Seamless Montage for Texturing Models". *Computer Graphics Forum* 29.2 (2010), 479–486 3.

[GYS*22] GAO, CHENJIAN, YU, QIAN, SHENG, LU, et al. "Sketchsampler: Sketch-based 3d reconstruction via view-dependent depth sampling". *European Conference on Computer Vision*. Springer. 2022, 464–479 2.

[HJN22] HÖLLEIN, LUKAS, JOHNSON, JUSTIN, and NIESSNER, MATTHIAS. "Stylemesh: Style transfer for indoor 3d scene reconstructions". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 6198–6208 3.

[HMLZ20] HAN, ZHIZHONG, MA, BAORUI, LIU, YU-SHEN, and ZWICKER, MATTHIAS. "Reconstructing 3D Shapes From Multiple Sketches Using Direct Shape Optimization". *IEEE Transactions on Image Processing* 29 (2020), 8721–8734. DOI: 10.1109/TIP.2020.3018865 2.

[HYY*25] HUNYUAN3D, TEAM, YANG, SHUHUI, YANG, MINGXIN, et al. "Hunyuan3D 2.1: From Images to High-Fidelity 3D Assets with Production-Ready PBR Material". *arXiv preprint arXiv:2506.15442* (2025) 2, 3.

[HZG*23] HONG, YICONG, ZHANG, KAI, GU, JIUXIANG, et al. "Lrm: Large reconstruction model for single image to 3d". *arXiv preprint arXiv:2311.04400* (2023) 3.

[IMT99] IGARASHI, TAKEO, MATSUOKA, SATOSHI, and TANAKA, HIDEHIKO. "Teddy: a sketching interface for 3D freeform design". *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, 409–416. ISBN: 0201485605. DOI: 10.1145/311535.311602. URL: https://doi.org/10.1145/311535.311602 2.

[JAC*21] JEONG, YOONWOO, AHN, SEOKJUN, CHOY, CHRISTOPHER, et al. "Self-calibrating neural radiance fields". *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, 5846–5854 5.

[JJKL16] JEON, JUNHO, JUNG, YEONGYU, KIM, HAEJOON, and LEE, SEUNGYONG. "Texture map generation for 3D reconstructed scenes". *The Visual Computer* 32.6 (2016), 955–965 3.

[KB06] KANNALA, JUHO and BRANDT, SAMI S. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". *IEEE transactions on pattern analysis and machine intelligence* 28.8 (2006), 1335–1340 6.

[KH06] KARPENKO, OLGA A and HUGHES, JOHN F. "Smoothsketch: 3d free-form shapes from complex sketches". *ACM SIGGRAPH 2006 Papers*. 2006, 589–598 2.

[KHW*22] KIM, BYUNGSOO, HUANG, XINGCHANG, WUELFROTH, LAURA, et al. "Deep reconstruction of 3D smoke densities from artist sketches". *Computer Graphics Forum* 41.2 (2022), 97–110 2.

[KPWG23] KNODT, JULIAN, PAN, ZHERONG, WU, KUI, and GAO, XIFENG. "Joint UV optimization and texture baking". *ACM Transactions on Graphics* 43.1 (2023), 1–20 3.

[LCD*23] LUO, ZHONGJIN, CAI, SHENGCAI, DONG, JINGUO, et al. "Rabit: Parametric modeling of 3d biped cartoon characters with a topological-consistent dataset". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 12825–12835 3.

[LGK*17] LUN, ZHAOLIANG, GADELHA, MATHEUS, KALOGERAKIS, EVANGELOS, et al. "3d shape reconstruction from sketches via multi-view convolutional networks". *2017 International Conference on 3D Vision (3DV)*. IEEE. 2017, 67–77 2.

[LGL*24] LONG, XIAOXIAO, GUO, YUAN-CHEN, LIN, CHENG, et al. "Wonder3d: Single image to 3d using cross-domain diffusion". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, 9970–9980 3.

[LHH22] LIN, ZUZENG, HUANG, AILIN, and HUANG, ZHEWEI. "Collaborative neural rendering using anime character sheets". *arXiv preprint arXiv:2207.05378* (2022) 3.

[LHK*20] LAINE, SAMULI, HELLSTEN, JANNE, KARRAS, TERO, et al. "Modular primitives for high-performance differentiable rendering". *ACM Transactions on Graphics (ToG)* 39.6 (2020), 1–14 5.

[LI07] LEMPITSKY, VICTOR and IVANOV, DENIS. "Seamless mosaicing of image-based texture maps". *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, 1–6 3, 7.

[LLC25] LLC, MESHY. *Meshy AI: Image to 3D*. https://www.meshy.ai/. Version –. 2025 1, 13.

[LLZ*23] LIU, YUAN, LIN, CHENG, ZENG, ZIJIAO, et al. "Syncdreamer: Generating multiview-consistent images from a single-view image". *arXiv preprint arXiv:2309.03453* (2023) 3.

[LPBM20] LI, CHANGJIAN, PAN, HAO, BOUSSEAU, ADRIEN, and MITRA, NILOY J. "Sketch2cad: Sequential cad modeling by sketching in context". *ACM Transactions on Graphics (TOG)* 39.6 (2020), 1–14 2.

[LPL*17] LI, CHANGJIAN, PAN, HAO, LIU, YANG, et al. "Bendsketch: Modeling freeform surfaces through 2d sketching". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1–14 2.

[LPL*18] LI, CHANGJIAN, PAN, HAO, LIU, YANG, et al. "Robust flow-guided neural prediction for sketch-based freeform surface modeling". *ACM Transactions on Graphics (TOG)* 37.6 (2018), 1–12 2.

[LZL*25] LI, YANGGUANG, ZOU, ZI-XIN, LIU, ZEXIANG, et al. "Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models". *arXiv preprint arXiv:2502.06608* (2025) 3.

[MR07] MEI, CHRISTOPHER and RIVES, PATRICK. "Single view point omnidirectional camera calibration from planar grids". *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, 3945–3950 6.

[MRP*23] METZER, GAL, RICHARDSON, ELAD, PATASHNIK, OR, et al. "Latent-nerf for shape-guided generation of 3d shapes and textures". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, 12663–12673 3.

[NISA07] NEALEN, ANDREW, IGARASHI, TAKEO, SORKINE, OLGA, and ALEXA, MARC. "Fibermesh: designing freeform surfaces with 3d curves". *ACM SIGGRAPH 2007 papers*. 2007, 41–es 2.

[PMKB23] PUHACHOV, IVAN, MARTENS, CEDRIC, KRY, PAUL G, and BESSMELTSEV, MIKHAIL. "Reconstruction of machine-made shapes from bitmap sketches". *ACM Transactions on Graphics (TOG)* 42.6 (2023), 1–16 2.

[PZG*24] PENG, HAO-YANG, ZHANG, JIA-PENG, GUO, MENG-HAO, et al. "Charactergen: Efficient 3d character generation from single images with multi-view pose canonicalization". *ACM Transactions on Graphics (TOG)* 43.4 (2024), 1–13 3.

[QMH*23] QIAN, GUOCHENG, MAI, JINJIE, HAMDI, ABDULLAH, et al. "Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors". *arXiv preprint arXiv:2306.17843* (2023) 3.

[RMA*23] RICHARDSON, ELAD, METZER, GAL, ALALUF, YUVAL, et al. "Texture: Text-guided texturing of 3d shapes". *ACM SIGGRAPH 2023 conference proceedings*. 2023, 1–11 3, 9, 11.

[Ros16] ROSEBROCK, DENNIS. "The Surface Model: An Uncertain Continuous Representation of the Generic Camera Model and Its Calibration". PhD thesis. Braunschweig University of Technology, Germany, 2016 6.

[SCZ*23] SHI, RUOXI, CHEN, HANSHENG, ZHANG, ZHUOYANG, et al. "Zero123++: a single image to consistent multi-view diffusion base model". *arXiv preprint arXiv:2310.15110* (2023) 3.

[SF*68] SOBEL, IRWIN, FELDMAN, GARY, et al. "A 3x3 isotropic gradient operator for image processing". *a talk at the Stanford Artificial Project in* 1968 (1968), 271–272 5.

[SHY25] SMITH, HARRISON JESSE, HE, NICKY, and YE, YUTING. "Animating Childlike Drawings with 2.5 D Character Rigs". *arXiv preprint arXiv:2502.17866* (2025) 3.

[SKČ*14] SỲKORA, DANIEL, KAVAN, LADISLAV, ČADÍK, MARTIN, et al. "Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters". *ACM Transactions on Graphics (TOG)* 33.2 (2014), 1–15 3.

[SLPS20] SCHOPS, THOMAS, LARSSON, VIKTOR, POLLEFEYS, MARC, and SATTLER, TORSTEN. "Why having 10,000 parameters in your camera model is better than twelve". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 2535–2544 6.

[SWSJ07] SCHMIDT, RYAN, WYVILL, BRIAN, SOUSA, MARIO COSTA, and JORGE, JOAQUIM A. "Shapeshop: Sketch-based solid modeling with blobtrees". *ACM SIGGRAPH 2007 courses*. 2007, 43–es 2.

[SWY*23] SHI, YICHUN, WANG, PENG, YE, JIANGLONG, et al. "MVDream: Multi-view Diffusion for 3D Generation". *ArXiv* abs/2308.16512 (2023). URL: https://api.semanticscholar.org/CorpusID:261395233 3.

[SZL*23] SMITH, HARRISON JESSE, ZHENG, QINGYUAN, LI, YIFEI, et al. "A method for animating children's drawings of the human figure". *ACM Transactions on Graphics* 42.3 (2023), 1–15 3.

[Tau95] TAUBIN, GABRIEL. "A signal processing approach to fair surface design". *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995, 351–358 7.

[TE07] TOEPFER, CHRISTIAN and EHLGEN, TOBIAS. "A unifying omnidirectional camera model and its applications". *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, 1–5 6.

[TRZ*23] TANG, JIAXIANG, REN, JIAWEI, ZHOU, HANG, et al. "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation". *arXiv preprint arXiv:2309.16653* (2023) 3.

[VD] VEKSLER, OLGA and DELONG, ANDREW. *gco-v3.0: Multi-label optimization*. Online; available: http://vision.cs.uwaterloo.ca/code/ 7.

[WLW*23] WANG, ZHENGYI, LU, CHENG, WANG, YIKAI, et al. "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation". *Advances in neural information processing systems* 36 (2023), 8406–8441 3.

[WLY*22] WANG, JIAYUN, LIN, JIERUI, YU, QIAN, et al. "3d shape reconstruction from free-hand sketches". *European Conference on Computer Vision*. Springer. 2022, 184–202 3.

[WLZ*24] WU, SHUANG, LIN, YOUTIAN, ZHANG, FEIHU, et al. "Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer". *Advances in Neural Information Processing Systems* 37 (2024), 121859–121881 3.

[WMG14] WAECHTER, MICHAEL, MOEHRLE, NILS, and GOESELE, MICHAEL. "Let there be color! Large-scale texturing of 3D reconstructions". *European conference on computer vision*. Springer. 2014, 836–850 3, 7, 9, 11.

[WPM*24] WEBER, ETHAN, PETERLINZ, RILEY, MATHUR, ROHAN, et al. "Toon3D: Seeing Cartoons from New Perspectives". *arXiv preprint arXiv:2405.10320* (2024) 3.

[XCS*14] XU, BAOXUAN, CHANG, WILLIAM, SHEFFER, ALLA, et al. "True2Form: 3D curve networks from 2D sketches via selective regularization". *ACM Transactions on Graphics* 33.4 (2014) 2.

[XHY*22] XU, PENG, HOSPEDALES, TIMOTHY M, YIN, QIYUE, et al. "Deep learning for free-hand sketch: A survey". *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022), 285–312 3.

[XLX*25] XIANG, JIANFENG, LV, ZELONG, XU, SICHENG, et al. "Structured 3d latents for scalable and versatile 3d generation". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 21469–21480 1–4, 8–10, 13, 14.

[YHK*24] YEH, YU-YING, HUANG, JIA-BIN, KIM, CHANGIL, et al. "Texturedreamer: Image-guided texture synthesis through geometry-aware diffusion". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 4304–4314 3.

[YKKS24] YOO, SEUNGWOO, KIM, KUNHO, KIM, VLADIMIR G, and SUNG, MINHYUK. "As-plausible-as-possible: Plausibility-aware mesh deformation using 2d diffusion priors". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, 4315–4324 5.

[YLT*25] YU, QIAO, LI, XIANZHI, TANG, YUAN, et al. "Fancy123: One Image to High-Quality 3D Mesh Generation via Plug-and-Play Deformation". *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. June 2025, 595–604 3.

[YYG*24] YU, XIN, YUAN, ZE, GUO, YUAN-CHEN, et al. "Texgen: a generative diffusion model for mesh textures". *ACM Transactions on Graphics (TOG)* 43.6 (2024), 1–14 3.

[ZK14] ZHOU, QIAN-YI and KOLTUN, VLADLEN. "Color map optimization for 3d reconstruction with consumer depth cameras". *ACM Transactions on Graphics (ToG)* 33.4 (2014), 1–10 3.

[ZLY*23] ZHOU, JIE, LUO, ZHONGJIN, YU, QIAN, et al. "GA-Sketching: Shape Modeling from Multi-View Sketching with Geometry-Aligned Deep Implicit Functions". *Computer Graphics Forum* 42.7 (2023), e14948 3.

[ZPW*23] ZHENG, XIN-YANG, PAN, HAO, WANG, PENG-SHUAI, et al. "Locally attentional sdf diffusion for controllable 3d shape generation". *ACM Transactions on Graphics (ToG)* 42.4 (2023), 1–13 2.

[ZQG*20] ZHONG, YUE, QI, YONGGANG, GRYADITSKAYA, YULIA, et al. "Towards practical sketch-based 3d shape generation: The role of professional sketches". *IEEE Transactions on Circuits and Systems for Video Technology* 31.9 (2020), 3518–3528 2, 3.

[ZXLF24] ZHOU, JIE, XIAO, CHUFENG, LAM, MIU-LING, and FU, HONGBO. "Drawingspinup: 3d animation from single character drawings". *SIGGRAPH Asia 2024 Conference Papers*. 2024, 1–10 2, 3.

[ZXW*25] ZHANG, YUQING, XU, HAO, WU, YIQIAN, et al. "AlignTex: Pixel-Precise Texture Generation from Multi-view Artwork". *ACM Transactions on Graphics (TOG)* 44.4 (2025), 1–12 3.

[ZYC*22] ZHANG, CONGYI, YANG, LEI, CHEN, NENGLUN, et al. "Creatureshop: Interactive 3d character modeling and texturing from a single color drawing". *IEEE Transactions on Visualization and Computer Graphics* 29.12 (2022), 4874–4890 3.