DOI: 10.1111/cgf.70392
EUROGRAPHICS 2026
A. Bönsch and N. Umetani
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2
STAR – State of The Art Report

Advances in Neural 3D Mesh Texturing: A Survey

Sai Raj Kishore Perla Hao Zhang Ali Mahdavi-Amiri

Simon Fraser University

Figure 1: ***Neural 3D Mesh Texturing*** *spans diverse settings, with works addressing different aspects of surface appearance generation and control. For instance, (a)* Texture Alignment*, learning class-consistent UV mappings [CYF22]; (b)* Texturing Humans and Garments *[ZWC*24], also demonstrating texture synthesis from an input image; (c)* Texture Variation and Transfer *[MEM24], also illustrating texturing from an input textured mesh; (d)* PBR Material Generation *[XLH*25], also showcasing texturing from a text prompt; and (e)* Scene-level Texturing *[WLX*24]. Figure adapted from [CYF22, ZWC*24, MEM24, XLH*25, WLX*24].*

Abstract

Texturing 3D meshes plays a vital role in determining the visual realism of digital objects and scenes. Although recent generative 3D approaches based on Neural Radiance Fields and Gaussian Splatting can produce textured assets directly, polygonal meshes remain the core representation across modeling, animation, visual effects, and gaming pipelines. Neural 3D mesh texturing therefore continues to be an essential and active area of research. In this survey, we present a comprehensive review of recent advances in neural 3D mesh texturing, covering methods for texture synthesis, transfer, and completion. We first summarize key foundations in mesh geometry, texture mapping, differentiable rendering, and neural generative models, and then organize the literature into a unified taxonomy spanning early GAN-based methods to modern diffusion-based pipelines. We further analyze common architectures and supervision strategies, review datasets and evaluation protocols, and discuss emerging applications, practical/commercial systems, and open challenges. Together, these insights provide a structured perspective on the current landscape and help guide future developments in learning-based 3D mesh texturing.

Project Page: sairajk.github.io/neural-mesh-texturing

CCS Concepts

• ***Computing methodologies*** → ***Texturing;*** *Image processing;* ***Neural networks; Mesh models;***

1. Introduction

Texturing 3D meshes has long been a central task in computer graphics and digital content creation, providing the visual richness that transforms geometric models into realistic and stylistically expressive assets [Hec86, DG01]. In traditional pipelines, this process typically relies on manual UV unwrapping together with artist-driven material setup and texture authoring [SPR06, LPRM02, SLMB05]. These steps usually require artistic skill and extensive labor. While procedural and image/example-based texturing techniques have eased some of this burden [PCOS10, WD97, Tur01, WLKT09], they remain largely constrained by handcrafted priors and limited semantic understanding, making it difficult to synthesize complex object appearances across diverse categories.

Recent advances in neural visual processing have shown great promise and significantly reshaped this landscape (Fig. 1). By leveraging deep learning, differentiable rendering [RRN*20, JSL*19], and large-scale image generative models [RBL*22], recent systems can automatically generate, complete, or refine textures for 3D meshes [CSL*23, PWMAZ24]. Instead of relying solely on manual texture authoring, these methods combine geometric reasoning with learned visual priors to guide texture synthesis from text, images, or sparse observations. This has opened a new paradigm for 3D asset creation, enabling meshes to be textured with greater realism, consistency, and flexibility while substantially reducing manual effort.

An early line of neural texturing methods relied on adversarial and feed-forward generation, learning to synthesize textures from weak 2D supervision through differentiable rendering [YDPT21, STM*22, BTD23]. These works demonstrated that neural networks could directly generate plausible mesh textures, but were often limited in semantic control, generalization, and diversity. Subsequent approaches shifted toward optimization-based pipelines guided by pretrained vision-language models such as CLIP [MBOL*22, MZS*23, CCL*22, R*21], showing that large-scale 2D priors can effectively steer texture synthesis in 3D. Later, diffusion-based optimization methods [CCJJ23, YHK*24, PJBM23] further improved fidelity and diversity through Score Distillation Sampling (SDS) [PJBM23] and related guidance schemes [WLW*23]. More recently, iterative and feed-forward diffusion pipelines [CSL*23, RMA*23, TZF*24] have enabled faster synthesis of high-resolution textures with stronger multi-view coherence. Complementary advances have also targeted human avatar texturing [NKML25, LZT*24], physically based material generation [HWLW25, CCL*22], and local or procedural control [DLAH24], further expanding the automation and editability of neural mesh texturing.

Despite these breakthroughs, challenges remain. Neural texturing still struggles with efficiency, multi-view consistency, and physical realism under varying illumination. Moreover, the scarcity of high-quality textured datasets and standardized evaluation mechanisms complicates benchmarking and comparison across methods. As the field moves toward prompt-driven 3D content generation and broader adoption of neural methods for content creation and manipulation, understanding the assumptions, algorithmic setups, the network architectures underlying these methods, as well as the trade-offs among them, becomes increasingly important.

To the best of our knowledge, while several surveys discuss texture synthesis and mapping, none focus exclusively on *neural 3D mesh texturing* as we do. In 2D, surveys of exemplar- and patch-based texture synthesis focus on image domains rather than 3D surfaces [BZ17, RDDM18]. In 3D, prior surveys have reviewed non-neural texture mapping and texturing topics including parameterization and mapping [Hec86, SPR06], texture mapping from photographs [WD97], example-based texture synthesis on surfaces [Tur01], and 3D texturing more broadly [DG01]. Later surveys consolidated example-based and volumetric texture synthesis, but they do not cover deep-learning-based techniques [WLKT09, PCOS10]. Relevant recent surveys on neural generation typically address broader themes, such as text-guided 3D editing [LLZ*24], neural stylization [CSS*25], or text-to-3D generation [LZC*23].

In this survey, we review recent advances in *neural 3D mesh texturing*, focusing on neural methods that operate directly on 3D meshes. For the purposes of this survey, we view 3D mesh texturing as involving two tightly coupled subproblems: *(i) surface parameterization*, which defines a 2D signal domain on the mesh surface (*e.g.*, UV charts and seams) where appearance can be stored and sampled, and *(ii) texture synthesis*, which generates the actual appearance representation, such as RGB textures, material maps, or learned features, either in that domain or in view space before baking the result back onto the mesh. Given the breadth of mesh parameterization research, this survey focuses primarily on *neural texture synthesis* and discusses parameterization only where it directly affects texture representation and learning; we refer readers to dedicated surveys for a more comprehensive overview [SPR06, HPS08, FH05].

We organize the literature primarily by recurring methodological families, while also reflecting the field's evolution from early foundational neural methods—including GAN-based and weakly supervised pipelines—to optimization-based approaches, such as vision–language-guided and diffusion-guided optimization, and, more recently, accelerated diffusion-based methods that synthesize textures through iterative, synchronized multi-view, or feed-forward generation. We also discuss specialized domains such as 3D human texturing and commercial systems, along with datasets and evaluation metrics, highlighting gaps and open challenges in the current research landscape. Ultimately, this survey aims to provide a comprehensive perspective on how neural models are transforming 3D texture generation by combining artistic creativity, physical realism, and scalable automation.

In the following, we first introduce key preliminaries in Sec. 2, followed by a discussion of guidance signals used in neural mesh texturing in Sec. 3. Sec. 4 surveys texturing methods, including foundational neural pipelines, optimization-based methods, and accelerated diffusion-based approaches, as well as 3D human texturing and commercial systems. We then review datasets and evaluation metrics in Sec. 5, summarize applications in Sec. 6, discuss limitations in Sec. 7, and conclude with future work in Sec. 8.

2. Neural Texturing Overview and Background

This section outlines the background required to understand and compare methods for neural 3D mesh texturing. We review the fun-

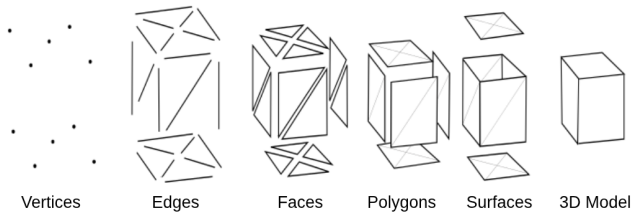

Figure 2: *A 3D mesh is defined by vertices (points), edges (line segments), and faces (surface elements, often triangles or quads), which combine into surface patches that approximate the object's geometry. Figure reproduced from [Lob09].*

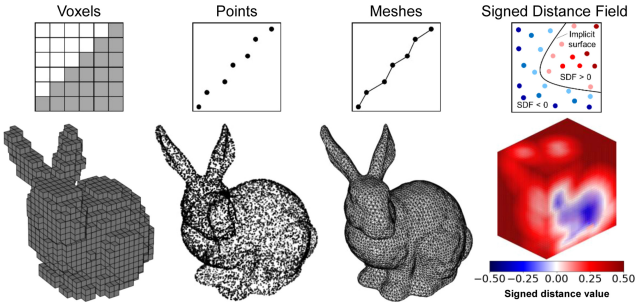

Figure 3: *3D shape representations. Voxels discretize space, point clouds sample points, meshes encode surface connectivity, and SDFs represent shapes implicitly via a continuous signed distance function. Figure reproduced from [AFL23].*

damentals of 3D mesh representation in Sec. 2.1, introduce surface parameterization as a bridge from 3D surfaces to 2D texture domains in Sec. 2.2, summarize rendering and differentiable rendering in Sec. 2.3, discuss physically based rendering (PBR) and PBR materials in Sec. 2.4, and outline key learning paradigms relevant to neural mesh texturing—neural fields [XTS*22], vision–language models [R*21, LLXH22], VAEs [KW19], GANs [GPM*14], and diffusion models [HJA20]—in Sec. 2.5. Together, these topics provide the foundation for the survey that follows.

2.1. Why 3D Meshes?

Triangle meshes are the dominant explicit representation for surfaces in computer graphics and vision [BKP*10, AHH*18, DSS*23, KMJ*19]. A mesh encodes geometry as a finite set of vertex positions together with a connectivity structure of edges and faces, which defines adjacency relations and surface topology (Fig. 2). In practice, triangle meshes support rich per-primitive attributes such as normals, colors, and material identifiers, making them a natural carrier for appearance information [AHH*18]. From a texturing standpoint, this explicit representation is crucial: appearance channels can be deterministically associated with well-defined surface elements, and mesh connectivity supports the consistent propagation of these signals across the surface [BKP*10].

Contrasting meshes with alternative shape encodings (Fig. 3) helps explain why they are the primary surface representation considered in this survey [BKP*10]. Point-based and surfel models

Figure 4: *Mesh parameterization maps a 3D surface to a 2D domain (UV space) by partitioning it into charts and flattening each into a planar region. Example meshes (left) and UV layouts (right) show varying chart counts. Figure adapted from [WWS*25].*

capture geometry without explicit connectivity and are attractive for point- or splat-based rendering, but they do not provide a standard 2D surface parameterization for texture mapping [PZVBG00, RL00]. Volumetric grids—dense or sparse—have been widely used for image-to-shape prediction and 3D reconstruction, but become memory-bound at high resolutions and typically require isosurface extraction to yield deployable surfaces [CXG*16, LC87]. Continuous implicit fields represent shapes as decision boundaries or signed-distance zero sets [MON*19, PFS*19], while radiance fields directly model view-dependent appearance in 3D space [MST*20]. Although these formulations excel at reconstruction and novel-view synthesis, their outputs are often converted to triangle meshes (*e.g.*, via marching cubes [LC87]) to interoperate with texture-map-based tools and rendering engines.

2.2. Mesh Parameterization

Mesh parameterization assigns each point on a surface a coordinate in a two-dimensional domain, defining a mapping from 3D surface space to 2D [SPR06] (Fig. 4). In practice, because general surfaces typically cannot be flattened into a single planar domain without introducing cuts and/or distortion, a triangle mesh is partitioned into a set of charts—connected surface patches—each mapped to a planar region; the collection of these charts, packed into one or more 2D texture images, forms an atlas. The resulting per-vertex UV coordinates define the texture parameterization and are interpolated per fragment during rendering. Charts meet along seams where the UV mapping is discontinuous. Parameterization also enables raster textures to be sampled and filtered during rendering. To improve filtering across scales, textures are typically stored in a mipmap hierarchy, which precomputes progressively downsampled versions of the image to reduce aliasing during minification [Wil83].

From a texturing standpoint, parameterization decouples geometry from appearance. Appearance attributes—such as albedo, normal, and roughness—can be stored in image space at resolutions largely independent of mesh tessellation and sampled consistently during rendering [YKSH19]. Once parameterized, texture resolu-

tion can be increased (*e.g.*, by using higher-resolution UV images) without altering the underlying geometry, allowing appearance fidelity to scale with available storage and bandwidth budgets.

While traditional pipelines represent appearance as one or more raster texture images in UV space (*e.g.*, albedo/normal/roughness maps), some neural methods replace or augment these images with *learned* representations. A representative example is *neural textures*—high-dimensional feature maps stored on a mesh (often still in UV space) and optimized jointly with a neural renderer, enabling the renderer to decode view-dependent appearance from learned features rather than directly from RGB texels [TZN19]. Other works represent texture implicitly as a continuous function over surface coordinates or 3D points (Sec. 2.5.1), but parameterization (UVs, seams, and chart layout) remains central whenever supervision or outputs are baked back into a 2D atlas.

Several alternatives to UV-based parameterization exist. *Per-face texturing* (*e.g.*, Ptex [BL08]) stores signals on individual faces without requiring a global UV layout, reducing seam management and atlas packing overhead, but it can introduce discontinuities when filtering across face boundaries. *Per-vertex colors* are simple and compact, but the achievable detail is constrained by mesh resolution (vertex density) and connectivity (triangle distribution), and interpolation across faces can blur high-frequency signals [AHH*18]. *Volumetric or procedural textures* avoid UV maps entirely, but they introduce challenges in authoring, editing, and manual control. Consequently, in practice, UV-based parameterization remains the most portable representation across offline and real-time rendering pipelines and continues to serve as the standard substrate for current texture libraries and authoring workflows [YKSH19].

Parameterization itself is also an active research problem that can materially affect downstream texturing quality, since distortion, seam placement, and chart fragmentation can influence both downstream learning behavior and the visibility of seams after synthesis. Classical methods target conformality and distortion control (*e.g.*, least-squares conformal maps [LPRM02], angle-based flattening [SLMB05], and boundary-first parameterization [SC17]), often coupled with seam optimization to reduce fold-overs and atlas fragmentation [LLC*18]. More recently, learned and task-adaptive approaches have emerged that automate or adapt parameterization toward data-driven objectives [LL22, LAKH23, WWS*25]: *Flatten Anything* proposes an unsupervised neural architecture that learns free-boundary surface parameterizations and can adaptively infer reasonable cuts and UV boundaries even from unstructured surface samples [ZHWH24], while *Auto-Regressive Surface Cutting* (SeamGPT) formulates seam generation as next-token prediction to produce semantically cleaner, less fragmented seam layouts for UV unwrapping [LCL*25]. Such methods complement robust classical solvers by reducing manual charting effort and by producing atlases better matched to modern texturing pipelines.

For these reasons, this survey considers mesh parameterization a foundational step for neural 3D mesh texturing: it decouples geometry from appearance by defining a 2D signal domain in which textures are stored (as images or learned features), optimized or regularized during learning, and consumed by renderers [YKSH19, SSGH01, Wil83]. Given the breadth of mesh parameterization research, we limit our discussion to aspects most relevant to texture representation and learning, and refer readers to dedicated surveys for a comprehensive overview [SPR06, HPS08, FH05].

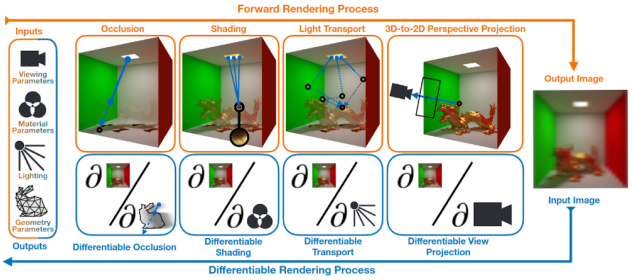

Figure 5: *Differentiable rendering enables backpropagation of image-space losses through the rendering pipeline (occlusion, shading, light transport, and projection) to optimize scene parameters (geometry, materials, lighting, and viewpoint). Figure reproduced from [BN25].*

2.3. Differentiable Mesh Rendering

Rendering converts a 3D scene into a 2D image by simulating what a camera would observe. Given a triangle mesh with associated materials and lighting, a forward rendering pipeline determines which surface points are visible and computes their appearance in the image. In rasterization-based rendering, mesh vertices are transformed from object space to camera space, projected onto the image plane, assembled into screen-space primitives, and resolved for visibility using a depth buffer [AHH*18]. In ray- or path-traced rendering, rays are cast through the scene to determine visible surface points and their light transport interactions [PJH23]. The visible points are then shaded by evaluating a material model under incident illumination [Kaj86], while textures are sampled—with filtering to reduce aliasing—to provide spatially varying parameters such as color or roughness [PJH23]. The resulting pixel values are finally composited with a background (solid color, image, or environment map) and written to an output image or framebuffer.

Differentiable rendering makes parts of the rendering process amenable to gradient-based optimization, enabling end-to-end learning of geometry, materials, and textures from image supervision (Fig. 5). A central challenge is the discontinuities inherent to visibility and rasterization. Classical approaches address this by introducing differentiable approximations or surrogate gradients, as in OpenDR [LB14], or by reformulating rasterization to expose stable derivatives, as in Neural Mesh Renderer [KUH18], Soft Rasterizer [LLCL19], and analytical treatments such as DIB-R [CGL*19]. In path-traced settings, differentiability is achieved via Monte Carlo estimators that handle visibility boundaries and other discontinuities [LADL18, LHJ19, ZMY*20, PJH23]. These advances are embodied in modern research renderers and toolchains: Mitsuba 2/3 [NDVZJ19] with its Dr.Jit compiler [JSRV22] supports forward- and reverse-mode differentiation for full light transport, while rasterization-based back ends such as nvdiffrast [LKA*20] and deep-learning libraries like PyTorch3D [RRN*20] and Kaolin [JSL*19] provide efficient GPU pipelines for differentiable rendering.

Figure 6: *Representative PBR materials rendered under identical geometry and lighting. Varying material map parameters (e.g., base color/albedo, roughness, metallic, normals, and emissive terms) yield distinct, relightable appearances such as leather, cloth, ceramic, wood, marble, microfiber, and chromium, illustrating how PBR materials extend beyond a single RGB texture. Figure reproduced from [wol16].*

2.4. Physically Based Rendering and PBR Material Maps

Physically based rendering (PBR) aims to model light–material interaction using physically grounded scattering and transport principles, so that appearance remains predictable under changing illumination and viewpoints. In the rendering equation formulation [Kaj86], outgoing radiance is determined by emitted radiance together with incident radiance modulated by the surface scattering function. For opaque surfaces, this scattering is typically modeled by a *bidirectional reflectance distribution function* (BRDF), and more generally by a BSDF when transmission is included [PJH23, Vea97]. The BRDF describes how incident light from one direction is reflected into another at a surface point [NRH*77]. In modern PBR systems, this reflectance is commonly modeled using energy-conserving microfacet models [CT81, Bur12, PJH23]. This perspective distinguishes *geometry* (shape and surface normals) from *material* parameters that govern diffuse and specular response, Fresnel effects [Sch94, BW99], and the distribution of micro-surface orientations.

A single RGB texture is often treated as a "color" signal (*e.g.*, diffuse/albedo) and may implicitly bake lighting, specular highlights, or other view-dependent effects, limiting relightability. In contrast, a PBR material is typically represented by a *set of texture maps* in UV space that parameterize a shading model. Modern real-time pipelines commonly adopt a metallic–roughness parameterization (as in Khronos glTF), where `baseColor` serves as diffuse albedo for dielectrics (non-metals) and specular reflectance color for conductors (metals), while `metallic` and `roughness` control reflectance behavior and highlight sharpness [Khrb, GA, Khra]. Additional maps such as normal, ambient occlusion, emissive, and, in broader PBR workflows, height/displacement can further enrich appearance; optional extensions such as clearcoat, sheen, transmission, and volume provide additional control over layered and transmissive effects [GA, Khrb] (Fig. 6). We use *PBR material generation* to refer to methods that explicitly synthesize such multi-map material representations rather than only a single RGB texture.

Generating high-quality PBR materials is harder than predicting a single RGB texture because multiple channels must be both *individually plausible* and *mutually consistent*. First, material maps are tightly coupled: normal detail should be consistent with roughness and specular response, and metallic regions should exhibit physically consistent baseColor/reflectance behavior [Bur12, GA]. Second, disentangling intrinsic material from illumination is ill-posed when supervision is limited to rendered RGB images, often causing "baked-in" shading artifacts that break under relighting. Third, real materials can be spatially varying and layered (paint/clearcoat, fabrics, skin/hair), and their appearance depends on the target shader and renderer conventions (color space, parameter ranges, map packing), making cross-system generalization non-trivial [GA, Khrb]. Finally, large-scale paired datasets of geometry with calibrated, multi-map ground-truth materials remain comparatively scarce relative to RGB imagery, complicating evaluation and often encouraging simplified supervision or incomplete material factorization [DAD*18]. These issues motivate current research directions that incorporate geometry cues, multi-view consistency, priors from image generative models, and renderer-aware training to better support relightable, editable PBR materials.

2.5. Neural Network Paradigms

Many neural mesh texturing pipelines can be understood through a small set of recurring paradigms for representing and generating surface appearance. We briefly introduce these paradigms—predicting textures or material maps in UV space, modeling appearance as a continuous function over the surface, and generating view-space images or latents that are later baked onto the mesh—as background for the methods surveyed in later sections.

2.5.1. Neural Fields

Neural fields, also referred to as coordinate-based neural representations, model signals as continuous functions of space, optionally extended with direction, time, or other conditioning variables, and parameterized by a neural network. Formally, a neural field is a function

$$f_\theta : \mathbb{R}^n \to \mathbb{R}^m$$

that maps a coordinate—such as a 3D point $\mathbf{x} \in \mathbb{R}^3$ or a tuple $(\mathbf{x}, \mathbf{d})$ with $\mathbf{d} \in \mathbb{S}^2$ (*i.e.*, a unit vector in $\mathbb{R}^3$) denoting a viewing direction—to a value representing signed distance, occupancy, color, or material parameters [XTS*22]. Unlike classical, explicitly sampled representations (*e.g.*, images, voxel grids, or vertex-attached attributes on meshes), a neural field encodes content in network weights and can be queried at arbitrary spatial resolution. This formulation yields a compact, differentiable, and resolution-independent representation well suited to inverse problems from images [MON*19, PFS*19]. Neural fields can be optimized end-to-end from image supervision via differentiable rendering. Consequently, Neural Radiance Fields (NeRFs) instantiate this idea for view synthesis by learning a mapping

$$f_\theta(\mathbf{x}, \mathbf{d}) \mapsto (\sigma, \mathbf{c}),$$

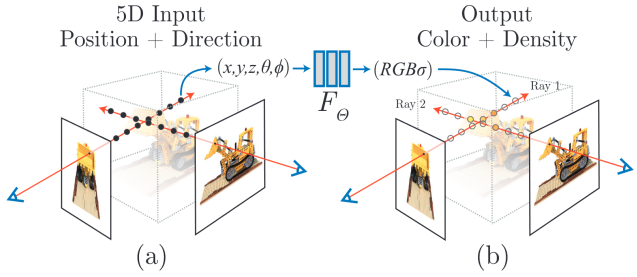

Figure 7: *An overview of Neural Radiance Fields (NeRFs). Images are synthesized by sampling 5D coordinates—3D position (x,y,z) and viewing direction (θ,ϕ)—along camera rays (a) and feeding them to a neural network that outputs color (R,G,B) and volume density σ (b). Figure adapted from [MST*20].*

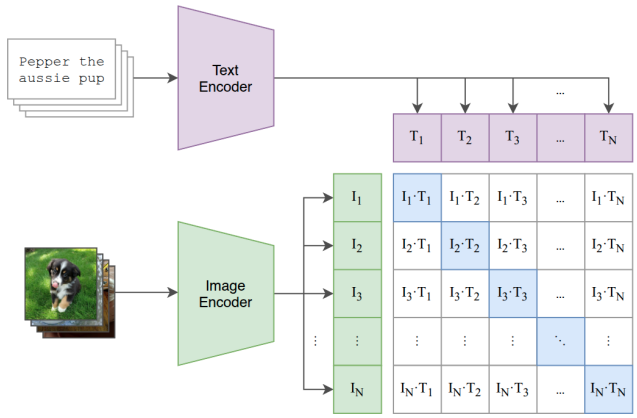

Figure 8: *CLIP jointly trains image and text encoders using a contrastive loss to match paired images and texts within a batch. Figure adapted from [R*21].*

where σ denotes volume density and $\mathbf{c}$ the view-dependent color (Fig. 7). Images are rendered via volumetric integration along camera rays [MST*20].

Despite their strengths, neural fields present several challenges for texturing. Training and optimization can remain computationally expensive, and the learned signal may entangle lighting and material properties unless these factors are explicitly modeled, causing baked textures to capture specular highlights or shadows as albedo. In practice, neural fields and mesh-based textures are often complementary: fields excel at recovering and regularizing fine-scale appearance from images, while meshes and UV maps provide an editable and interoperable substrate for downstream use.

2.5.2. Vision Language Models

Vision–language models (VLMs) are neural networks trained on paired image–text data to align visual content with natural language. In contrast to vision-only encoders (which map images to features) or language-only models (which operate purely over text), VLMs learn shared cross-modal representations that support semantic alignment between modalities. In neural 3D mesh

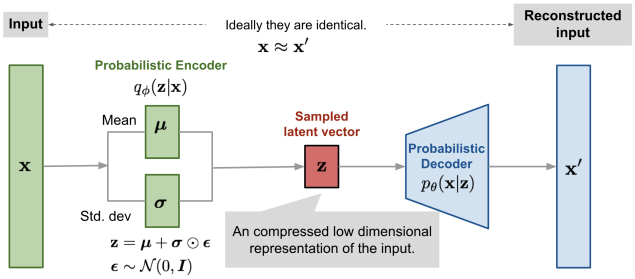

Figure 9: *An overview of variational autoencoders (VAEs). A probabilistic encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ maps an input $\mathbf{x}$ to a Gaussian latent distribution parameterized by (μ,σ); a latent sample $\mathbf{z} = \mu + \sigma \odot \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is then decoded by $p_\theta(\mathbf{x} \mid \mathbf{z})$ to reconstruct $\mathbf{x}' \approx \mathbf{x}$. The model is trained by maximizing the evidence lower bound (ELBO) $\mathcal{L}_{ELBO}$ (see Sec. 2.5.3). Figure reproduced from [Wen18].*

texturing, VLMs are especially valuable because they provide (i) open-vocabulary guidance for text-driven appearance synthesis, (ii) multi-view scoring of renders that links textual intent to visual fidelity, and (iii) region- or phrase-level grounding to target specific mesh parts.

A foundational instance is *CLIP* [R*21], which learns dual encoders for images and text using a symmetric contrastive objective over large-scale web image–text pairs (Fig. 8). By maximizing cosine similarity for matched image–text embeddings and minimizing it for mismatched pairs, CLIP induces a shared embedding space that supports robust zero-shot recognition and has become a widely used source of supervision for text-driven stylization and texture transfer [MBOL*22, CCL*22, KXBP22]. *GLIP* [LZZ*22] extends this cross-modal alignment toward object detection and phrase grounding, producing region-aware, language-conditioned visual representations. *BLIP* [LLXH22] further introduces a multimodal encoder–decoder framework that supports both understanding-oriented objectives (e.g., image–text matching) and generation tasks such as captioning and VQA.

2.5.3. Variational Autoencoders

Variational autoencoders (VAEs) [KW14] are latent-variable generative models that pair an encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ with a decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ and are trained by maximizing the evidence lower bound (ELBO) [JGJS99, NH98, BKM17] (Fig. 9). For a single observation $\mathbf{x} \in \mathcal{X}$ and a latent code $\mathbf{z} \in \mathbb{R}^d$ with prior $p(\mathbf{z})$, the ELBO is

$$\mathcal{L}_{\mathrm{ELBO}}(\theta,\phi;\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x} \mid \mathbf{z})\big] - \mathrm{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big),$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler divergence [KL51]. After training, generation proceeds by ancestral sampling: one first draws $\mathbf{z} \sim p(\mathbf{z})$ (*e.g.*, $\mathcal{N}(\mathbf{0},\mathbf{I})$) and then samples $\mathbf{x} \sim p_\theta(\mathbf{x} \mid \mathbf{z})$.

VAEs offer a continuous latent space that supports interpolation, sampling, and, in conditional variants, controllable generation and editing. However, standard VAEs often produce overly smooth samples and can suffer from posterior collapse. Although more expressive priors and approximate posteriors [RM15, KSJ*16] can improve results, VAEs generally fall short of GANs [GPM*14] and diffusion models [RBL*22] in perceptual fidelity.

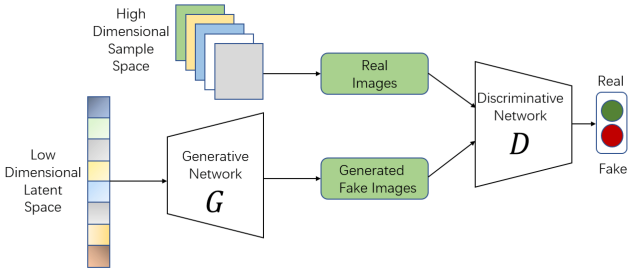

Figure 10: *An overview of generative adversarial networks (GANs). A generator G maps noise samples (latent codes) to synthetic images, while a discriminator D learns to distinguish real images from generated ones; both are trained adversarially using $\mathcal{L}_{GAN}$ (see Sec. 2.5.4). Figure reproduced from [CCC*20].*

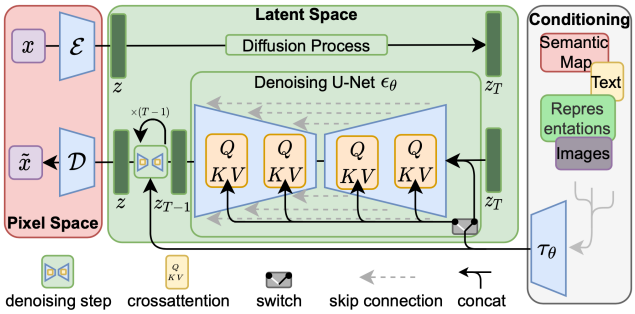

Figure 11: *An overview of latent diffusion models (LDMs). An autoencoder $(\mathcal{E},\mathcal{D})$ encodes images $\mathbf{x}$ into latents $\mathbf{z}$ (and decodes back to $\tilde{\mathbf{x}}$), and a denoising network ϵ_θ iteratively predicts and removes noise; conditioning (e.g., text/images/semantic maps) is injected via cross-attention. Figure reproduced from [RBL*22].*

2.5.4. Generative Adversarial Networks

Generative adversarial networks (GANs) formulate generative modeling as a two-player game between a generator G and a discriminator D [GPM*14]. The classical minimax objective

$$\mathcal{L}_{\text{GAN}} = \min_G \max_D \Big[\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]\Big]$$

trains D to distinguish real samples $\mathbf{x}$ from synthesized ones $G(\mathbf{z})$, while G learns to fool D (Fig. 10). Under an optimal discriminator, this objective can be shown to correspond to minimizing the Jensen–Shannon divergence between the data and model distributions, which motivated later variants based on alternative divergence measures.

In fact, numerous extensions have strengthened the original GAN formulation, improving training stability, sample fidelity, and controllability. For instance, Wasserstein GANs [ACB17,GAA*17] reformulate adversarial training using the Wasserstein-1 (Earth-Mover) distance, yielding smoother gradients and more stable optimization. Complementarily, PatchGAN discriminators classify local image patches rather than entire images, enabling sharper detail in conditional settings such as image-to-image translation [IZZE17].

However, common challenges in training GANs include instability (*e.g.*, oscillations or failure to converge), mode collapse (loss of diversity), and sensitivity to the choice of objective and regularization. While Wasserstein objectives and gradient penalties help mitigate these issues, they do not eliminate them entirely [ACB17, GAA*17]. Nonetheless, GANs remain attractive for high-fidelity, sharp synthesis and for learning class- or image-conditional mappings with controllable outputs.

2.5.5. Diffusion Models

Diffusion models learn to reverse a process that gradually perturbs data into Gaussian noise [SWMG15, HJA20]. The forward process is defined by a variance schedule $\{\beta_t\}_{t=1}^T$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}\big),$$
$$\bar{\alpha}_t = \textstyle\prod_{s=1}^t (1-\beta_s), \quad \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I}).$$

A neural network ϵ_θ is then trained to predict the injected noise from $(\mathbf{x}_t, t)$ [HJA20]. In the commonly used simplified objective, this becomes

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\mathbf{x}_0,t,\epsilon}\Big[\big\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\big\|_2^2\Big].$$

Conditioning variables $\mathbf{c}$ (*e.g.*, text features) can be incorporated in several ways, including guidance at sampling time and conditioning mechanisms within the denoiser such as cross-attention.

Direct diffusion in pixel space is computationally expensive. Latent diffusion models mitigate this by learning an autoencoder $(\mathcal{E},\mathcal{D})$ and performing the diffusion process in a lower-dimensional latent space $\mathbf{z} = \mathcal{E}(\mathbf{x})$, with conditioning (*e.g.*, text) injected via cross-attention inside the denoiser [RBL*22] (Fig. 11). Beyond text prompts, additional conditioning modules have been introduced to provide finer control without retraining the full backbone. ControlNet [ZRA23] adds condition-specific, zero-initialized branches to incorporate structural signals such as edges, depth, or human pose while preserving the pretrained model. IP-Adapter [YZL*23] introduces a lightweight image-prompt pathway through decoupled cross-attention, enabling style and identity control alongside text conditioning.

Diffusion has been applied to a wide range of modalities, including images [RBL*22], video [HSG*22], audio [LCY*23], and 3D content [LGT*23, CCJJ23], as well as mesh-based textures, as discussed in subsequent sections. Collectively, the core formulation, latent variants, and conditioning mechanisms have evolved into a flexible toolkit that has broadened the practicality of diffusion-based generation.

3. Guidance for 3D Mesh Texturing

This section discusses the types of *guidance* used in neural 3D mesh texturing. Here, guidance refers to the signals or conditions that specify aspects of the desired texture and steer the generative model toward a target result. Broadly, we group guidance into two categories: *structural* guidance, which constrains the spatial layout or geometric placement of texture on the 3D surface, and *stylistic* guidance, which specifies the desired appearance of the texture.

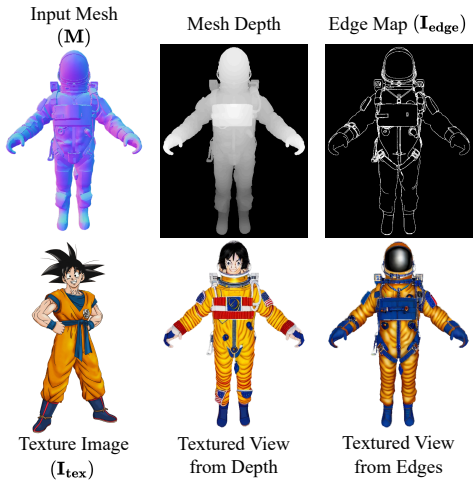

Figure 12: *Depth vs. edges for structural guidance. Depth maps are typically smooth and provide coarse structural cues. In contrast, geometric edges derived from mesh attributes (e.g., normals, depth discontinuities, and connectivity) are more detailed and aligned with mesh geometry, improving mesh–texture consistency. Figure reproduced from [PWMAZ24].*

3.1. Structural Guidance

Structural guidance provides geometric cues that inform the model about *where* specific content should appear on the mesh. It does not define the artistic style of those elements, but helps ensure that generated textures remain consistent with the underlying geometry and part layout of the mesh. Because it is tied to the 3D surface, such guidance is often derived from the input mesh or from mesh-aligned intermediate representations. Regardless of model design or stylistic conditioning, many methods condition the generation process on structural information to avoid textures that conflict with geometry. Several forms of structural guidance have been explored, including geometric edges derived from mesh attributes (normals, depth, connectivity) [PWMAZ24] (Fig. 12), depth maps [RMA*23, CSL*23], normal maps [YHK*24], canonical UV layouts of SMPL/SMPL-X human templates [LMR*15, PCG*19, LZT*24], and silhouettes [YDPT21]. More generally, any representation that provides a correspondence between 3D surface regions and 2D texture locations can serve as structural guidance. The choice and integration of such cues usually depend on the task and model design.

3.2. Stylistic Guidance

Stylistic guidance defines *what* the texture should look like—its color palette, materials, patterns, or overall appearance—independent of geometry. It specifies the desired appearance of the texture and is typically provided through an external conditioning signal, such as text, reference images, or exemplar textures. Texture generation can also proceed without such input, *i.e.*, unconditionally, in which case the style of the output is drawn solely from the learned training distribution. Conversely, when conditioning is provided, the model is guided to match the specified stylistic attributes.

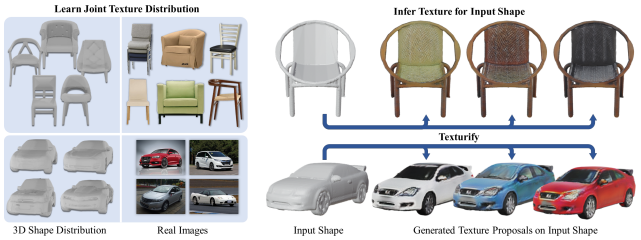

Figure 13: *Unconditional texture generation with Texturify. The model learns a distribution of plausible textures conditioned on 3D shape from real-image supervision, and infers diverse texture proposals for a given input shape. Figure reproduced from [STM*22].*

3.2.1. Unconditional Texture Generation

Unconditional texture generation produces textures without explicit stylistic guidance, instead sampling appearance from a learned distribution, typically from random noise or a latent code. In this setting, the model may still be conditioned on the mesh or other structural inputs, but it is not given user-specified appearance cues. Without such high-level conditioning, specific attributes cannot be directly controlled; however, varying the random seed or latent input allows diverse sampling from the learned distribution, with limited control possible through latent-space manipulations, as in 2D GANs [KLA19, KLA*20] (Fig. 13).

The main advantage of unconditional generation is its simplicity and ability to produce diverse textures without user input. However, the absence of explicit appearance control is a major limitation, making it less practical when a specific target style is required. On the positive side, unconditional generation is often relatively straightforward to integrate, since randomness can be introduced through latent or noise inputs without requiring additional conditioning signals.

3.2.2. Conditional Texture Generation

Conditional texture generation provides explicit control over the style or appearance of the output through additional input cues. The model receives external conditioning signals that specify the desired texture and uses them to guide generation. Stylistic guidance can take various forms—such as text, reference images, exemplar textures, or textured 3D assets. Conditioning greatly expands user control and broadens the range of possible tasks (*e.g.*, texture transfer, domain-specific texturing, and style interpolation), but also introduces challenges: the model must faithfully reproduce the specified style while maintaining coherence with the underlying mesh structure. The choice of conditioning modality often influences the model architecture, training strategy, and target applications.

Conditional texture generation is often evaluated along two dimensions: *(i)* the intrinsic quality and realism of the generated texture, and *(ii)* its faithfulness to the input guidance. High-quality textures that deviate from the prompt, or prompt-faithful results that lack realism, are both inadequate; achieving both plausibility and guidance adherence remains a central goal of conditional methods.

Figure 14: *Text-conditioned texture generation. Given untextured meshes (left), Text2Tex synthesizes textures guided by text prompts (right). Figure reproduced from [CSL*23].*

Text-based Generation. Using natural language as guidance is one of the most popular and convenient ways to control 3D texture synthesis, owing to the accessibility of text as an interface. In text-guided texturing, a user provides a description (prompt) of the desired texture, and the model conditions generation on this input to produce a semantically matching result (Fig. 14). Text is typically incorporated via a pretrained encoder that converts the prompt into a conditioning representation, which is then injected into the generation pipeline. Many systems use vision–language models such as CLIP [R*21] or dedicated text encoders like T5 [RSR*20], which are usually pretrained on large text or image–text datasets and may either be used as is or fine-tuned for the task at hand.

Text guidance provides high-level, intuitive control, allowing users to specify abstract concepts and attributes that are otherwise difficult to quantify. Its drawback is ambiguity; text can be underspecified or open to multiple interpretations. Diffusion-based methods can help mitigate this, as large text-to-image models like Stable Diffusion [RBL*22] are trained on vast datasets and can translate rich textual cues into detailed imagery. Leveraging such models allows text-guided 3D texturing to inherit this semantic knowledge.

Image-based Generation. Using an image example as guidance provides a more concrete and precise specification of texture style. Instead of describing the appearance in words, the user provides one or more reference images, and the model generates a texture that matches their visual characteristics. Such guidance can capture details that are difficult to express textually, *e.g.*, specific patterns, artistic styles, or complex color gradients. In image-based texture generation, the reference image is typically encoded into a conditioning representation using a visual encoder [R*21, SZ15], or incorporated through image-conditioning modules [YZL*23], which then guide texture generation toward the desired style/appearance.

The advantage of image guidance lies in its specificity, meaning that it is generally less ambiguous than text. A reference image can convey fine texture details (for instance, the floral pattern on a dress or the wood grain of a chair) that would be tedious to describe verbally, making this a highly practical form of guidance.

Textured 3D Shape-based Generation. A more structured form of stylistic guidance uses a textured 3D shape, or a mesh-aligned appearance representation, to guide the texturing of another mesh. In this setting, appearance is specified directly in 3D or in an aligned surface domain, rather than through a text prompt or a single reference image. Such guidance is particularly useful for tasks such as texture alignment and transfer across related shapes [CYF22], as well as generating variations of an existing textured asset and, in some cases, transferring learned appearance statistics to new geometries [MEM24].

Overall, using textured 3D shapes as guidance is less common than text or image inputs, mainly because such exemplars are relatively scarce and often task-specific. When available, however, they can provide detailed and geometrically aligned appearance information. Future systems may combine multiple forms of guidance, *e.g.*, text prompts, reference images, and partial or complete 3D textures, to provide users with finer control over 3D texture creation.

4. Neural 3D Mesh Texturing

In this section, we review works on *Neural 3D Mesh Texturing* and organize them according to recurring methodological families, while also reflecting the field's evolution over time. We begin with foundational neural mesh texturing methods that laid the groundwork for later approaches by using differentiable rendering, weak 2D supervision, and adversarial learning to generate textures from limited 2D supervision. We then discuss optimization-based methods, which iteratively refine textures using pretrained priors such as vision–language or diffusion models. Finally, we review accelerated diffusion-based methods, which can be further categorized by inference-time strategy into iterative view-by-view pipelines, synchronized multi-view approaches, and feed-forward methods. This taxonomy reflects a trade-off space between quality, speed, and controllability, and highlights the evolution from slower but flexible optimization to more scalable generation while maintaining consistency across the 3D surface. We provide a summary of representative works in *Neural 3D Mesh Texturing* in Tab. 1.

4.1. Foundational Neural Mesh Texturing

Early works on neural 3D mesh texturing demonstrated that neural networks can learn to synthesize realistic textures on 3D surfaces or in surface-aligned representations from limited or indirect supervision. Many of these methods rely on differentiable rendering to bridge the 2D–3D gap, often leveraging unpaired or weakly paired 2D images to learn how to assign realistic textures to 3D meshes. Broadly, these approaches can be grouped into image-driven texturing, texture super-resolution and completion, neural texture representations in function space, and adversarial or GAN-based synthesis pipelines.

Image-driven Texture Transfer and Reconstruction. *PhotoShape* [PRFS18] is an early large-scale effort in this direction, automatically assigning photorealistic appearance to collections of untextured 3D shapes by mining product photographs and material exemplars. Its system retrieves images and materials with appearance similar to a target mesh and aligns them to the shape, enabling large-scale creation of textured ShapeNet-style [CFG*15] assets. Around the same time, Kanazawa *et al.* [KTEM18] introduced a category-specific mesh predictor that jointly infers 3D geometry and a corresponding UV texture map from a single image via differentiable rendering, without relying on ground-truth 3D supervision. The model deforms a fixed template mesh with a shared

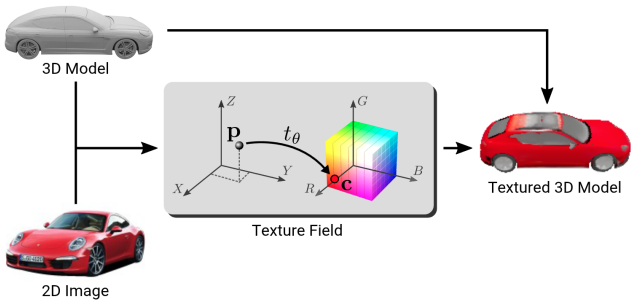

Figure 15: *An overview of TextureFields. Given a 3D shape (and optionally a reference image), the method learns a continuous texture field t_θ mapping surface points* **p** *to colors* **c**, *producing a textured mesh. Figure reproduced from [OMN*19].*

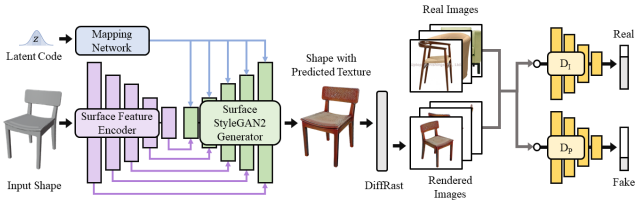

Figure 16: *Training pipeline of Texturify. Given an untextured input mesh, a surface-feature encoder and a StyleGAN2-based [KLA*20] generator predict a texture for the mesh. The textured mesh is then differentiably rendered, and the resulting images are evaluated by two discriminators (a global discriminator D_I and a patch-based discriminator D_P) against real images; the adversarial loss trains all networks. Figure reproduced from [STM*22].*

UV atlas to match each target's geometry while preserving consistent UV coordinates. This canonical UV space gives texels a more stable semantic correspondence across instances, helping disentangle texture from shape and pose. As a result, the network can learn category-level appearance priors efficiently, using the shared template as a strong inductive bias for texture prediction. Building on this idea of category-level canonicalization, Henderson *et al.* [HTL20] proposed one of the first fully generative models that learns both shape and texture directly from collections of unpaired 2D images. Unlike Kanazawa *et al.* [KTEM18], they omit UV maps entirely, representing texture as piecewise-constant per-face colors on a fixed-topology mesh while predicting vertex positions for geometry. This yields class-consistent face correspondences that enable joint sampling of plausible geometry and appearance from the image distribution, demonstrating that coherent textured shapes can emerge from 2D supervision alone. These methods collectively established that realistic texture generation can be achieved by learning to invert the rendering process using only 2D supervision.

Texture Super-Resolution and Completion. Another challenge for early neural texturing pipelines was the limited resolution of reconstructed textures. Richard *et al.* [RCO*19] and Li *et al.* [LTT*19] addressed this problem through learned texture upsampling and refinement networks. Their CNN-based models fuse multi-view imagery and low-resolution atlases to synthesize sharper, high-frequency texture details that conventional texture-fusion pipelines often fail to recover. In parallel, Chibane and Pons-Moll [CPM20] proposed to complete missing texture regions using an implicit representation: their Implicit Feature Network (IF-Net) predicts per-point texture values conditioned on partial scans, enabling plausible inpainting of unseen surfaces consistent with the geometry. These works treat texture refinement as a learnable process, moving beyond simple interpolation (*e.g.*, view-dependent texture sampling [DTM96]) or seam-hiding blends and global color optimization [ZK14, WMG14, PGB03].

Neural Texture Representations in Function Space. Oechsle *et al.* [OMN*19] proposed *Texture Fields* to represent texture as a continuous neural function that maps spatial queries to RGB color (Fig. 15). This implicit formulation reduces dependence on fixed UV atlases and allows resolution-independent texture definition. Texture Fields can be trained jointly with shape representations and, in generative settings, combined with adversarial training [GPM*14] to produce high-quality renderings. Recent successors have extended this idea toward neural material representations. For example, TexGaussian [XLH*25] proposed an octree-based 3D Gaussian representation for feed-forward PBR material generation, predicting albedo, roughness, and metallic parameters from geometry-consistent 3D features.

Adversarial and GAN-based Texture Synthesis. Generative Adversarial Networks (GANs) became a prominent direction in early neural texturing pipelines. Huang *et al.* [HTD*20] introduced an adversarial optimization framework for RGB-D scans, refining noisy vertex colors by enforcing photorealism through a learned patch discriminator. By jointly optimizing texture and geometry under differentiable rendering, they achieved sharper and more consistent results than traditional photometric blending. Yu *et al.* [YDPT21] learned texture generators for ShapeNet [CFG*15] object collections using only untextured meshes and unaligned internet photos, relying on a common UV layout per class and adversarial 2D supervision. Their network learns to populate the UV map such that renders of the textured meshes match real photographs, establishing an early class-level neural texture generation framework. 3DStyleNet [YGS*21] disentangles geometry and texture style in latent space, allowing independent control of shape and appearance and enabling style transfer across objects. SPSG [DST*21] extends these ideas to full indoor RGB-D scenes through a self-supervised approach that hallucinates missing geometry and surface color for large 3D scans, demonstrating neural texturing at the scene level.

Direct Surface Texture Generation. Later works shifted toward generating textures directly on mesh surfaces without relying on explicit UV alignment. *Texturify* [STM*22] is a surface-based GAN that predicts per-face colors directly on the mesh (Fig. 16). It leverages a hierarchical 4-RoSy parameterization [HZY*19] to define orientation-consistent face convolutions, combining a mesh-face encoder with a StyleGAN2-inspired [KLA*20] decoder. Training uses multi-view differentiable rendering from random cameras and two discriminators: an image discriminator for overall realism and

a patch-consistency discriminator to enforce cross-view consistency. This framework avoids explicit UV atlases and the associated seam/distortion issues, learns geometry-aware textures from unpaired 2D images and untextured meshes, and captures high-frequency detail, with its effective resolution primarily limited by the number of faces at the finest 4-RoSy level.

Mesh2Tex [BTD23] extends this line of work by replacing direct per-face color prediction with a *hybrid mesh–neural-field* representation: a face-convolution encoder–decoder produces coarse per-face features, while a shared neural field maps these features to RGB. This formulation enables high-resolution texture generation beyond coarse face-wise color prediction. Mesh2Tex also supports *image-guided* texturing through inference-time optimization, aligning renders of the target mesh with a reference image in a perceptual sense.

ShaDDR [CCZZ23] developed an example-based deep generative framework that, given a coarse voxel shape, detailizes geometry and generates textures in a style learned from a small set of textured exemplars. For texture synthesis, differentiable rendering compares multi-view renders of the generated shape against exemplar texture images, while style is controlled through learned latent codes. These methods mark a transition from 2D-conditioned texturing toward more direct neural texture generation on 3D surfaces.

While these neural texturing approaches demonstrated that realistic and controllable 3D texture generation is feasible, they also faced important constraints. Their reliance on limited object categories and small datasets restricted generalization (*e.g.*, category-specific canonical UVs or class-specific shape collections) [KTEM18, YDPT21], and GAN-based objectives often suffered from instability or limited diversity relative to later generative models [LKM*18]. Many architectures also required meshes with specific structural or parameterization assumptions (*e.g.*, 4-RoSy-based surface representations or class-aligned UV layouts), limiting applicability to arbitrary models [STM*22, YDPT21]. Furthermore, text-based or open-vocabulary control was essentially absent in this era—capabilities that only emerged with subsequent vision–language and diffusion frameworks [CCL*22, RMA*23, CSL*23]. Even methods that achieved high realism, such as *Texturify* [STM*22] and *Mesh2Tex* [BTD23], remained limited by the fidelity/diversity trade-offs of pre-diffusion generative priors and lacked the broader semantic control and robustness introduced by these newer paradigms. Nonetheless, these pioneering works established the foundations of modern neural texturing by demonstrating that neural networks, coupled with differentiable rendering, can learn to synthesize textures directly on 3D meshes from weak 2D supervision [KUH18, STM*22, KTEM18].

4.2. Optimization-based Texturing

In *optimization-based texturing*, a mesh's texture (and sometimes geometry) is directly refined to satisfy objectives derived from powerful pre-trained models, leveraging their prior knowledge to generate diverse, detailed textures, often without task-specific training. Such methods offer several benefits, including improved texture consistency: optimizing a shared texture map across many rendered views can help mitigate discontinuities that often affect multi-view feed-forward approaches.

Vision–Language Model Guidance. A prominent optimization-based line of work uses vision–language models such as CLIP [R*21] to guide texture synthesis. As a precursor, Neural 3D Mesh Renderer [KUH18] introduced a differentiable renderer and demonstrated that mesh textures (and even vertices) can be optimized using image-based objectives. In particular, it showed that a mesh can be iteratively updated to minimize a 2D *style loss* [GEB16], enabling style transfer from an image onto a 3D asset. Building on this idea, later methods replaced style images with natural-language descriptions using CLIP's joint vision–language embedding. In these approaches, the mesh is rendered from multiple viewpoints, and the CLIP loss between rendered images and the target text is used to update the texture. For instance, Text2Mesh [MBOL*22] optimizes mesh appearance and local geometric detail to match a given prompt in CLIP space, enabling compelling text-driven stylization through optimization alone. Subsequent works improved efficiency and fidelity: X-Mesh [MZS*23] introduced a text-guided dynamic attention mechanism that improves stylization accuracy and convergence speed while refining both geometry and texture. When geometry deformation is disabled, the method reduces to a purely text-guided texture optimization pipeline.

Several related methods also use CLIP-based objectives for text-driven 3D appearance optimization [KXBP22, Jet21, JMB*22]. While CLIP-guided texture optimization can produce broadly plausible and semantically relevant results without task-specific 3D training data, its visual quality is often limited by the coarse supervision provided by CLIP embeddings. To push toward photorealism, TANGO [CCL*22] extends this line of work to optimize spatially varying material properties, local geometric variation, and lighting under a differentiable renderer. By predicting material, normal map, and lighting condition from a text prompt, TANGO produces more realistic appearances, including shiny or metallic finishes, and enables photorealistic text-driven stylization. Overall, CLIP-based methods established the feasibility of text-driven texturing and stylization, but were later surpassed in fidelity by diffusion-based approaches.

Diffusion-Driven Optimization. Instead of relying on CLIP embedding alignment, some methods optimize textured meshes using guidance from a pre-trained diffusion model, *e.g.*, Stable Diffusion [RBL*22], conditioned on the same text prompt. The key idea, introduced by DreamFusion [PJBM23], is *Score Distillation Sampling* (SDS): rather than comparing against a single target image, SDS uses the score function of a frozen text-to-image diffusion model to provide gradients that drive the 3D representation toward the prompt-conditioned image distribution. In practice, this is often written as an SDS objective $\mathcal{L}_{\text{SDS}}$ whose gradients encourage rendered views of the mesh to be interpreted by the diffusion model as matching the prompt. Optimizing this objective directly on a mesh's texture can synthesize more complex and higher-frequency details than earlier CLIP-based methods.

A limitation of basic SDS, however, is oversmoothing or oversaturation of textures, partly due to how the guidance gradients are obtained. To address this, ProlificDreamer [WLW*23] proposed *Variational Score Distillation* (VSD). VSD treats the desired 3D texture as a random variable and optimizes a variational bound, which in-

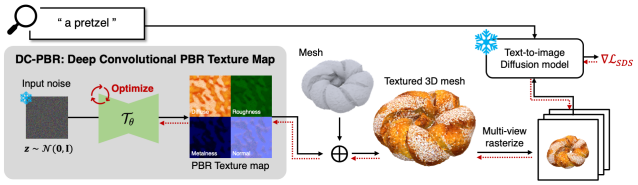

Figure 17: *An overview of Paint-It. Given an untextured mesh and a text prompt, the method optimizes PBR texture maps (e.g., albedo, roughness, metalness, normals) using SDS guidance [PJBM23] from a frozen text-to-image diffusion model over multi-view renderings. Figure reproduced from [YJPMO24].*

troduces diversity and reduces bias in score estimation. As a result, VSD can yield sharper and more diverse details than basic SDS. Although DreamFusion and ProlificDreamer optimize NeRF-based representations, the same principles can transfer to explicit mesh textures. In fact, *Latent-NeRF* [MRP*23] showed that latent-space score distillation can be applied directly to a mesh's UV texture map. In their scheme, the texture is represented in the latent space of a pretrained autoencoder; SDS is applied in that compact space to generate a coarse latent texture, which is then decoded to RGB and further refined. This yields high-resolution textures more efficiently than pixel-space optimization.

Subsequent works extended this paradigm to physically based material generation. Fantasia3D [CCJJ23] demonstrated text-to-3D content creation with disentangled geometry and appearance. By optimizing an explicit mesh for shape and a neural appearance representation for material under SDS guidance, Fantasia3D generates textured meshes with relightable PBR material properties from a text prompt. Building on this direction, Paint-It [YJPMO24] targets high-fidelity physically based texture generation (Fig. 17). The authors observed that applying SDS directly to pixel-wise texture maps leads to noisy and blurry results due to uneven gradient coverage. To address this, they represent each texture map (diffuse, specular, roughness) with a lightweight convolutional network instead of raw pixels. This re-parameterization acts as an implicit regularizer, filtering high-frequency noise from diffusion gradients and enabling coarse-to-fine optimization. Paint-It achieves faster convergence and finer details (e.g., text and engravings) than prior distillation-based methods, while also allowing explicit lighting control during texture generation. FlashTex [DOW*24] introduces LightControlNet, an illumination-aware diffusion model conditioned on user-specified lighting (e.g., HDRI or text). Its pipeline first generates lighting-consistent multi-view images of the mesh and then refines the texture through optimization so that the recovered appearance remains relightable and consistent under arbitrary illumination. Going a step further, DreamMat [ZLX*24] trains a diffusion model that is explicitly geometry- and light-aware. The model conditions on geometric cues such as normal and depth maps together with lighting and text, and learns to generate physically plausible material maps. Relatedly, DreamPBR [XZP*25] targets text-driven, high-resolution SVBRDF generation with multimodal guidance, producing relightable PBR parameter maps that can complement mesh-based material texturing.

More recently, video diffusion models have been explored for texturing because they often provide stronger inter-frame, and thus cross-view, consistency than image-based generators. For example, VideoMat [MWL*25] extracts relightable PBR material maps for a given 3D shape by first generating multi-view, view-consistent renderings with a fine-tuned video diffusion model, then recovering base color, roughness, and metallic through intrinsic decomposition, and finally applying differentiable path-traced refinement to output standard PBR maps.

Beyond text prompts, optimization-based texturing has been adapted to other forms of guidance. TextureDreamer [YHK*24] addresses image-driven texture synthesis. Given only 3–5 reference photos of a real object or scene, it personalizes a diffusion model to capture the reference appearance, in a manner inspired by DreamBooth [RLJ*23], and then optimizes the target mesh's relightable texture maps using VSD [WLW*23] so that rendered views match the references.

In a similar spirit, StyleTex [XZT*24] focuses on style transfer from a single 2D image. It decouples the image's style from its content in CLIP space, then injects the style features via cross-attention during diffusion-guided, multi-view optimization of the mesh's UV texture, while using the content features as negative guidance to suppress content leakage. This yields RGB (non-PBR) textures that faithfully inherit the reference image's artistic style (color palettes, brushstrokes, *etc.*) without distorting the mesh structure. Another intriguing direction is optimizing procedural material parameters instead of raw textures. MaPa [ZPX*24] segments a 3D model and assigns each part a procedural material graph, as used in tools like *Blender* [Ble25] or *Substance 3D* [ado25]. The parameters of these graphs (*e.g.*, noise scale, color, roughness) are optimized using a segment-controlled text-to-image diffusion model that synthesizes part-aligned target images, bridging text descriptions to material parameters without paired training data. This produces high-quality, tileable textures with correct reflectance and, crucially, editable procedural materials rather than baked bitmaps.

There have also been efforts to texturize an entire scene. SceneTex [CLL*24] addresses this challenge for large indoor 3D scenes with many objects from a text prompt. Instead of optimizing a single explicit scene-level UV atlas, it represents appearance using a multi-resolution texture field and optimizes it with a score-distillation-based objective. To ensure global coherence, SceneTex introduces view-dependent refinement through a cross-attention decoder that propagates appearance information across viewpoints, yielding fully textured rooms in which walls, floors, and furniture share a consistent style.

Finally, optimization can also be constrained to *local* regions of a mesh. 3D Paintbrush [DLAH24] enables localized text-driven texturing using a novel Cascaded Score Distillation (CSD) loss that combines guidance from multiple diffusion stages. Given a mesh and a text prompt such as *"superman emblem"*, it jointly learns a localization map to identify the target region and a texture map, both represented as neural fields, confining the edit to the predicted area while leaving the rest of the texture unchanged (Fig. 18).

Optimization-based texturing has significantly advanced the quality and flexibility of 3D asset creation, but it remains limited by runtime. Unlike feed-forward networks that generate textures

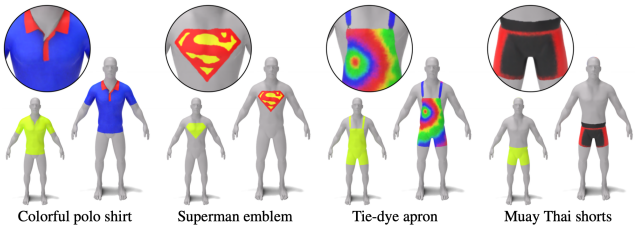

Figure 18: *Given a text prompt, 3D Paintbrush predicts both the target region and corresponding texture on the input mesh, enabling spatially controlled texturing. Figure adapted from [DLAH24].*

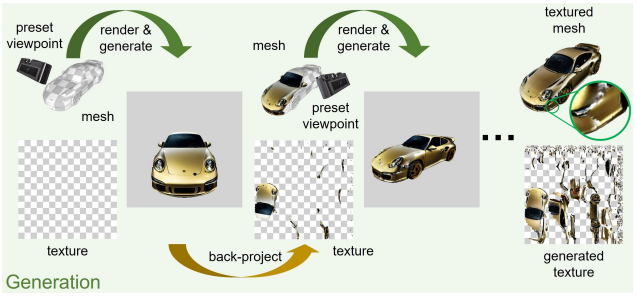

Figure 19: *An overview of an iterative texturing pipeline. The mesh is rendered from preset viewpoints, a text-to-image model generates view-consistent appearances, and the results are back-projected to update the texture map, iterating across views. Figure adapted from [CSL*23].*

in a single pass, these methods typically require many optimization and denoising steps per shape. Even with accelerations such as *latent-space distillation/optimization* [MRP*23] or *efficient differentiable rendering* [LKA*20,NDVZJ19,JSRV22,RRN*20], texturing a single model can still take several minutes, making these approaches costly, especially for time-sensitive or large-scale applications without substantial compute. Nonetheless, optimization-based texturing opened a new frontier: by leveraging powerful 2D priors, it enables automatic texture generation at a level of detail and semantic richness that was previously difficult to achieve. Ongoing research seeks to reduce runtimes through *better initialization and staged pipelines* [LGT*23], *parallelism across views*, and *hybrid learning–optimization designs* that combine generative priors with render-and-backprop refinement [DOW*24, YHK*24]. At the same time, work on *creative control—e.g.*, localized edits and procedural/material parameter optimization—has expanded user control over where and how edits are applied [DLAH24, ZPX*24, GZD*23]. In this sense, combining differentiable rendering with powerful vision–language and diffusion priors remains a compelling paradigm for high-quality 3D mesh texturing.

4.3. Accelerated Diffusion-based Texturing

Early works demonstrated that 2D diffusion models can be used to optimize 3D textures (*e.g.*, by score distillation [PJBM23]). While these optimization-based methods achieve impressive results, they come at the cost of lengthy per-object optimization (often tens of minutes). The need for faster, more scalable approaches motivated a new class of methods that avoid costly test-time optimization by instead using diffusion models in one of three ways: (1) iterative view-by-view painting with a frozen 2D diffusion prior, (2) synchronized multi-view generation that diffuses all views concurrently, or (3) feed-forward texture synthesis in which a custom diffusion model directly outputs a full texture map.

Iterative Texturing. A widely adopted pipeline "paints" the mesh one view at a time in a loop: (i) render the current viewpoint with geometric cues (*e.g.*, depth or normals); (ii) use a pretrained image diffusion model to synthesize an image consistent with the current view and prompt; (iii) back-project the synthesized colors onto the UV map using visibility; and (iv) update a per-texel progression mask before moving to the next view [RMA*23, CSL*23]. Dynamic keep/refine/generate masks guide denoising toward unpainted or low-confidence regions while protecting already generated high-quality textures, progressively covering the UV space across multiple views [RMA*23, CSL*23] (Fig. 19). Many implementations also include a final seam-refinement pass to clean residual discontinuities along chart boundaries [CSL*23]. In practice, this iterative scheme can yield high-quality textures without fine-tuning the diffusion backbone, although minor seams, color drift, or incomplete coverage may remain across view boundaries [RMA*23, CSL*23].

Several works build on the iterative painting paradigm for specialized scenarios. EASI-Tex [PWMAZ24] targets single-image texture transfer: given a reference photo, it uses an IP-Adapter [YZL*23] to diffuse the photo's texture onto a 3D shape. It also showed that geometric edges yield more structurally accurate textures than depth or normals. Paint3D [ZCQ*24] follows a coarse-to-fine strategy: it first samples a pretrained diffusion model to obtain a coarse texture, then uses custom UV-space diffusion models to upsample, inpaint missing regions, and remove baked-in lighting. In the interest of speed, Make-A-Texture [XGF*25] optimizes the diffusion model and introduces a specialized backprojection algorithm that generates a full texture in only 3 seconds, trading some detail for significant speed gains. Iterative pipelines have also been adapted to complex scenes: InstanceTex [YGC*24] textures multi-object scenes instance by instance while aiming to maintain global coherence, whereas RoomTex [WLX*24] unwraps indoor scenes into panoramas for an initial global pass, then iteratively inpaints each object with panoramic guidance to mitigate inter-object style inconsistencies. Despite these advances, iterative methods can still exhibit minor seams or blur due to view-wise generation, though they remain far faster and more practical than per-mesh optimization, requiring a fixed number of diffusion denoising steps per view instead of long gradient-based optimization.

Synchronized Texturing. Another approach is to generate all views simultaneously. Synchronized multi-view texturing methods perform a joint diffusion process across multiple camera views, sampling all views together at each denoising step and aggregating their latent updates onto a common texture representation to promote globally consistent results [CKF*23, LXLW24, SXSX25]. By

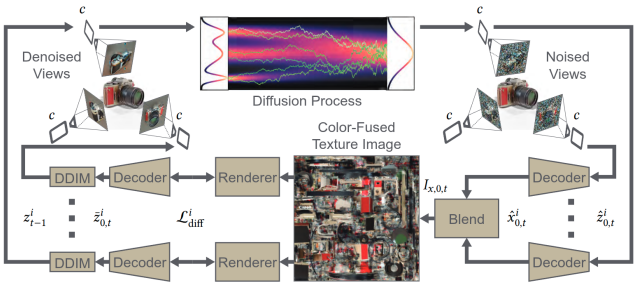

Figure 20: *An overview of a synchronized multi-view texturing pipeline. Per-view diffusion denoising is coupled through a* Blend *module that fuses per-view textures into a shared texture image, improving cross-view consistency. Figure reproduced from [ZPZ*24].*

parallelizing view synthesis, such methods can reduce runtime and improve consistency, since texture evolves jointly across views and overlapping boundary regions are encouraged to agree [CMZ*25]. One of the first such approaches, TexFusion [CKF*23], introduced multi-view latent fusion in which the diffusion denoiser operates on several rendered views concurrently, and the predicted 2D latent features are aggregated onto a shared latent texture map. The final RGB texture is then recovered by optimizing an intermediate neural color field on decoded renders of that latent texture, yielding globally coherent results without the long per-object optimization used in earlier SDS-based pipelines.

Building on this idea, GenesisTex [GJL*24] performs diffusion directly in texture space, maintaining one latent texture map per viewpoint during denoising and dynamically aligning them to ensure cross-view coherence. It further introduces style-consistency mechanisms and latent alignment to enforce a unified appearance across viewpoints. A related method, MVPaint [CMZ*25], also generates all views simultaneously using synchronized multi-view diffusion, but follows this with 3D inpainting and UV-space refinement to fill unobserved regions and reduce cross-view inconsistencies, producing seamless high-resolution textures.

Many other recent works adopt synchronized diffusion to improve multi-view consistency. TexPainter [ZPZ*24] enforces multi-view consistent text-to-texture synthesis using a pre-trained latent diffusion model by jointly denoising multiple views. At each DDIM step, the predicted noiseless states are decoded to images, fused in color space into a common texture, and the fusion objective is back-propagated to the view latents, relaxing sequential dependencies and improving cross-view consistency and quality (Fig. 20). RomanTex [FYY*25] improves geometry- and image-aligned multi-view diffusion for texturing via a 3D-aware rotary positional embedding and a decoupled attention design (reference attention plus multi-view attention), strengthening cross-view coherence before baking to UV space. VCD-Texture [LYC*24] goes further by introducing a 3D–2D collaborative denoising framework that aggregates multi-view 2D latent features into 3D space and rasterizes them back to produce more consistent 2D predictions. This coupling yields highly stable textures, as the 2D diffusion is continuously guided by a holistic 3D representation and vice versa. MaterialMVP [HYY*25a] extends synchronized multi-view diffusion to *PBR* outputs by jointly generating view-consistent albedo and metallic–roughness maps, using consistency-regularized training to suppress illumination artifacts and multi-channel aligned attention to keep the predicted maps spatially aligned. Meanwhile, GenesisTex2 [LZZ*25] extends GenesisTex [GJL*24] with improved stability and quality by introducing local attention reweighting in the diffusion model's self-attention layers, ensuring that spatially corresponding patches across views remain strongly correlated. It further merges latent features across views at multiple stages to enforce consistency without compromising diversity.

More recently, SeqTex [YYS*25] leverages video diffusion priors to strengthen cross-view coherence in synchronized texturing. It formulates mesh texturing as *sequence generation* and jointly models multi-view renderings and UV textures using geometry-informed attention. By coupling view-space and UV-space generation within a unified framework, it produces complete UV texture maps end-to-end and reduces reliance on post-hoc baking or fusion.

Overall, synchronized approaches demonstrate that parallel multi-view generation greatly enhances texture consistency. By jointly denoising all views through shared latent variables or cross-view interactions, these methods substantially reduce seams. However, UV-space fusion can behave like an averaging operation: if applied too strongly throughout the full denoising trajectory, it can suppress high-frequency details and yield over-smoothed (blurry) textures [LXLW24]. As a result, synchronization is often emphasized in early (high-noise) timesteps to align global layout and color, and relaxed in later timesteps to preserve fine details; but once coupling is weakened, per-view updates can drift and residual inconsistencies may still arise, especially in low-overlap or occluded regions, often motivating downstream 3D/UV refinement (*e.g.*, MVPaint [CMZ*25]). The main trade-off is higher memory consumption—since multiple views are diffused simultaneously—and, in some cases, the need for custom synchronization mechanisms or additional guidance modules (*e.g.*, GenesisTex [GJL*24], FlexiTex [JYZ*25]). Nevertheless, synchronization leverages the diffusion model's global coherence to produce seamless textures that are difficult to achieve with strictly view-by-view strategies [CSL*23, RMA*23].

Feed-Forward Texturing. To avoid per-instance optimization, one solution is to train a diffusion model that directly generates the full texture representation in a single pass, without requiring view-by-view rendering during inference. Many current feed-forward texturing methods operate in UV space, allowing occluded regions to be textured since the full surface is visible to the model. This design avoids view-by-view processing and stitching, but typically requires large datasets of textured 3D objects [DSS*23, D*23], which have only recently become available (Fig. 21).

TEXGen [YYG*24] is a 700M-parameter diffusion model trained to generate 1024×1024 UV maps for meshes across diverse categories. Its architecture interleaves 2D convolutions on the UV map with self-attention over a 3D point-cloud representation of the mesh, injecting 3D structural awareness into 2D texture generation. Once trained, TEXGen can synthesize diverse, high-resolution textures in a single forward pass conditioned on text or reference images. Related feed-forward diffusion models also couple 3D and UV-space reasoning: Point-UV Diffusion [YDL*23]

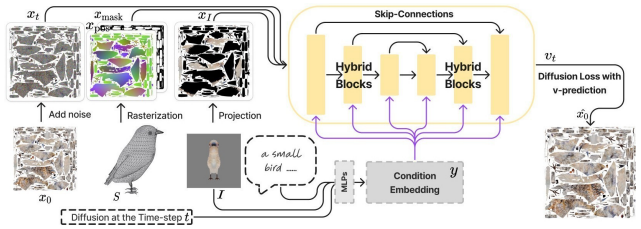

Figure 21: *A feed-forward texturing pipeline (TEXGen). A diffusion U-Net denoises a latent UV texture atlas conditioned on rasterized mesh cues and text embeddings to generate a UV texture map in a single pass. Figure reproduced from [YYG*24].*

adopts a coarse-to-fine design that first diffuses appearance in surface point space to obtain a globally consistent prior, then projects it into UV space and refines a high-resolution texture atlas, helping reduce seams and UV-fragmentation artifacts. More recent feed-forward models continue to broaden this design space. For example, UniTEX [LLC*25] moves beyond purely UV-based processing by lifting texture generation into a unified 3D functional space, while Material Anything [HWLW25] targets physically based material generation, predicting not only albedo but full PBR material maps for a given 3D object.

Feed-forward models have also explored alternative representations. UV-free Texture Diffusion [FZB24] bypasses UV maps entirely by operating on colored point clouds and using heat diffusion over the mesh surface. Its UV3-TeD framework models each surface as a point set with a learnable latent field, enabling texture generation directly on the mesh without a fixed UV parameterization and thereby avoiding seam- and distortion-related issues associated with UV maps. Likewise, Single-Mesh Diffusion [MEM24] learns field latents attached to mesh vertices and trains a diffusion model to generate these latent codes, which are then decoded to colors. Under a feed-forward generation setting for a given trained asset, this enables rapid, view-consistent synthesis of many texture variants when the geometry is known. Another notable work, AlignTex [ZXW*25], addresses the challenge of generating textures that precisely match concept art or reference images of a 3D asset. It employs a diffusion-based two-stage pipeline with alignment losses and transformer-based conditioning to enforce pixel-level correspondence between multi-view artwork and the generated UV map, yielding pixel-precise rather than style-consistent texture synthesis.

Consequently, diffusion-based texturing has rapidly advanced from slow per-mesh optimization [PJBM23, WLW*23] to fast feed-forward generation [HWLW25]. Iterative methods built on the idea of leveraging 2D diffusion models for 3D texture painting [CSL*23, RMA*23], and they remain attractive for their simplicity (no model retraining) and controllability (one can intervene at each view). However, they may still produce seams or inconsistent details, since each view is processed sequentially. Synchronized multi-view diffusion [LXLW24, CKF*23] significantly alleviates this issue: by treating all views (or the UV map) as a joint diffusion task, these methods encourage the texture to evolve coherently across the surface. The cost is higher memory usage and, in some cases, the need for custom diffusion pipelines, but the benefit is improved quality on challenging cases (intricate objects, full scenes) where view-by-view approaches often struggle with misalignment. Feed-forward approaches [ZXW*25] push this further by directly mapping text (and other inputs) to the texture domain, often without explicit view-by-view rendering during inference. This yields substantial speedups and opens the door to zero-shot and generative applications, such as texturing large numbers of objects with diversity.

A core challenge for feed-forward models is the distribution of training data. Large diffusion-based texture models are often trained on synthetic datasets or broad 3D asset collections (*e.g.*, ShapeNet [CFG*15] or Objaverse [DSS*23, D*23]), which may not fully reflect the visual richness of real-world textures. As a result, their outputs may still fall short of the photorealism of real imagery. For instance, a model trained purely on synthetic stylized assets might produce textures with a telltale "CGI" look. Current feed-forward models such as Material Anything [HWLW25] and TEXGen [HGZ*24] already exhibit coherent results, but slight domain gaps remain in high-frequency detail and material realism. One promising direction is to combine the strengths of synthetic and real data, *e.g.*, using large synthetic datasets to learn geometry-to-texture alignment while leveraging real image data to improve appearance realism and material properties. Some recent works have moved in this direction by incorporating real-image losses or conditioning signals: Material Anything [HWLW25] integrates rendering-based losses to encourage physical realism, while FabricDiffusion [ZWC*24] uses real fashion photographs as inputs.

4.4. 3D Humans Texturization

Texturing human avatars merits separate discussion, since the underlying assumptions, constraints, and objectives differ from those of general object texturing. First, representation priors differ: many modern human texturing methods build on parametric body models (*e.g.*, SMPL/SMPL-X [LMR*15, PCG*19]) with fixed topology and canonical UV atlases, providing consistent surface correspondences across poses [LMR*15, PCG*19]. In contrast, general object texturing often lacks such canonical parameterization.

Second, human identity and anatomy impose distinctive constraints: faces, hands, and skin carry fine-grained, identity-defining cues (freckles, lip color, *etc.*) that must be preserved for realism and recognition. Accordingly, many methods incorporate face-specific priors or losses (*e.g.*, a facial UV refinement network or an identity loss) to capture such high-frequency details [OLY*17, DCX*18, WZL*19]. Moreover, several works segment the texture or mesh into semantic parts and apply part-specific models or losses [CSST21].

Third, the data and evaluation protocols are domain-specific: human texturing methods typically train on specialized datasets such as artist-modeled digital humans, 3D scans, or multi-view captures [LZT*24, LIPM19], since fully paired ground-truth textures remain scarce. Evaluation also relies on human-centric metrics, including identity preservation (*e.g.*, face-ID or person re-identification [WZL*19]) and part-aware image fidelity measures such as PSNR or LPIPS computed on rendered views or aligned texture regions [CSST21].

Finally, applications in digital humans demand distinct capabilities: avatars for gaming, virtual try-on, and related interactive settings require textures that are high-fidelity and semantically editable (*e.g.*, "change the shirt logo to #10") [MAPM20, CSST21]. In dynamic settings, even minor misalignments can cause perceptible flicker under motion, breaking realism. Accordingly, human texturing methods often place greater emphasis on semantic controllability and, when animation is involved, temporal consistency of attributes such as clothing patterns [CSST21, MAPM20].

Earlier neural pipelines. Earlier neural pipelines aim to generate full-body textures from sparse inputs, often a single image, typically leveraging 2D supervision through *UV mapping*. Lazova *et al.* [LIPM19] reconstruct a full 360° human texture from a single view by fitting an SMPL mesh to the image, projecting visible pixels into its UV atlas, and training a network to inpaint occluded regions. Grigorev *et al.* [GSVL19] address pose variation through *coordinate-based inpainting*: they map an input person image into UV space using DensePose [GNK18] correspondences and train a GAN to fill missing UV regions, helping preserve fine clothing details such as logos and prints under re-posing. A related idea appears in DensePose Transfer [NGK19], which extracts a source UV texture, reprojects it onto a target pose, and refines seams and missing details via a neural network. These UV completion methods—including UV-GAN [DCX*18] for faces—leverage the maturity of 2D image synthesis in UV space, reducing 3D consistency to a 2D inpainting problem. AUV-Net [CYF22] formalizes this further by learning an *aligned* UV parameterization for an entire shape class, mapping semantically corresponding surface points to shared UV coordinates across instances. By training a network to deform each mesh into a common UV template, it achieves dataset-wide texture alignment without manual unwrapping, enabling generative models (GANs or diffusion) trained in this UV space to produce textures transferable across meshes. TexturePose [PKD19] does not output an explicit texture map but enforces *texture consistency* during training of a human mesh estimator: if consistent UV textures are predicted across viewpoints, the underlying 3D shape is more likely to be accurate. This is implemented by projecting the image from one view onto the predicted mesh, rendering it to a second view, and comparing against the true image. Although geometry-focused, its core idea—*consistent textures imply consistent geometry*—is conceptually related to later methods that jointly reason about shape and texture.

Supervised texture generation with human priors. Some methods treat human texturing as a direct translation problem from images of people or clothing into a UV texture, often leveraging human parsing or identity cues to handle misalignment. Zhao *et al.* [ZLZS20] propose a human-parsing-guided texture transfer model in which a single image's semantic segmentation provides pose and shape cues. During training, they enforce *cross-view consistency* by predicting textures from two views and *exchanging* them to render the opposite view, optimizing a loss between the rendered and input images. This formulation requires no ground-truth 3D textures and enables plausible completion of invisible regions during inference. Wang *et al.* [WZL*19] address the task using an identity objective, with a person re-identification network serving as a perceptual metric for texture generation. From an input image, they extract a partial texture from visible pixels, render a full-body image, and use a pretrained re-ID CNN to compare this render with the real image. The texture generator is then optimized so that the rendered avatar matches the input identity in feature space. Mir *et al.* [MAPM20] address a practical scenario: texturing 3D garments from catalog photos. Given front- and back-view clothing images, their model learns dense correspondences from 2D silhouette coordinates to the garment surface in UV space. Trained on synthetic data with known correspondences, it warps clothing patterns onto a garment template to generate a texture map and supports real-time virtual try-on.

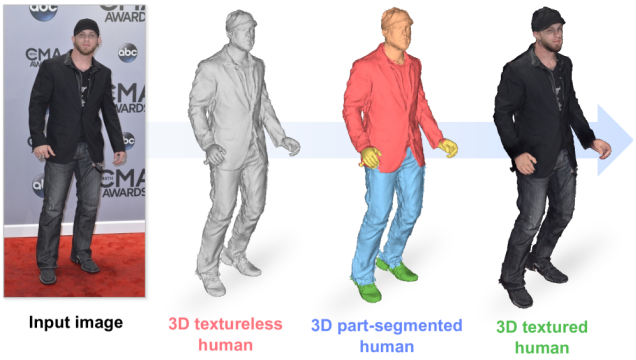

Figure 22: *Part-aware human texturing. Given a single input image (left), the method segments a textureless human mesh into semantic parts and textures them to produce a detailed 3D human with clean region boundaries (right). Figure adapted from [NKML25].*

Later works extend part-specific texturing. PARTE [NKML25] segments the human mesh into semantic parts (torso, arms, pants, *etc.*) and generates textures for each part to prevent cross-region bleeding. A *PartSegmenter* labels each vertex by part, while a diffusion-based *PartTexturer* fills the corresponding UV regions conditioned on part descriptions, part labels, and neighboring parts. This produces sharper transitions and cleaner textures, improving downstream reconstruction (Fig. 22). Chaudhuri *et al.* [CSST21] propose a semi-supervised framework for generating high-resolution (1024^2) editable textures. Their region-adaptive VAE (ReAVAE) learns a latent *style code* for each semantic region (face, shirt, pants, shoes, *etc.*) on the UV map. At generation time, a segmentation mask and optional style vectors guide the synthesis of a consistent texture map, enabling localized edits such as changing only the shirt's pattern or color. To overcome limited paired data, single-view images are projected into texture space and used as partial supervision. These supervised and weakly supervised methods leverage human-specific cues—such as parsing, keypoints, and identity features—to infer plausible textures from sparse inputs, enabling more flexible generative approaches.

High-fidelity generation via diffusion-based models. Recent methods move toward more flexible human texture generation using diffusion-based models, with a key focus on maintaining *3D consistency* and *semantic control*. TexDreamer [LZT*24] exemplifies this trend as a zero-shot multimodal 3D human texturing sys-

tem. By adapting a large text-to-image model to a semantic human UV structure using the ATLAS dataset, it generates detailed skin and clothing textures directly in a canonical UV layout, bypassing view rendering. A feature translator maps text or reference-image inputs into the diffusion latent space while conditioning on UV geometry. TexGarment [LWG*25] targets standalone garments, combining a pre-trained text-to-image diffusion Transformer with 3D structural guidance to generate diverse, 3D-consistent textures. It injects UV position maps and 3D shape cues (*e.g.*, point clouds) to enforce alignment across seams and improve 3D consistency.

Beyond generation, transfer methods also recover reusable materials from real photos. FabricDiffusion [ZWC*24] recovers a distortion-free, tileable fabric texture from a single in-the-wild garment photo by "unwarping" it into a flat texture map. Trained on synthetic garments with known projected patterns, it infers the underlying tileable material, which can then be mapped to garment UVs and coupled with existing PBR material generation pipelines for high-fidelity rendering and virtual try-on. Make-It-Vivid [TZF*24] explores text-driven texture generation for cartoon characters by adapting a pretrained diffusion model to paired UV maps and text descriptions, and further employs adversarial learning to sharpen details and reduce the synthetic-to-real domain gap.

Finally, methods such as HumanRef [ZLZ*24] bridge generative texturing and 3D reconstruction via Ref-SDS, a reference-guided score distillation approach that optimizes a textured 3D human model from a single image. By incorporating image guidance and region-aware attention, HumanRef preserves fine appearance details from the reference image while maintaining cross-view consistency. SiTH [HSH24] takes a related but feed-forward route: it trains an image-conditioned diffusion model to hallucinate unseen back-view appearance from a front view, then feeds the generated views into a mesh reconstruction and texturing pipeline to recover full-body textures. These works highlight how diffusion models and large-scale generative priors support both open-ended texture generation from text or images and strong inpainting of occluded regions where deterministic methods often struggle.

All in all, human texturing has progressed from UV-space completion and transfer [LIPM19, GSVL19, MAPM20] to powerful diffusion- and transformer-based generation/transfer systems [LZT*24, LWG*25, ZWC*24], with part-aware designs improving seam handling and editability [CSST21, NKML25]. Yet, several domain-specific hurdles remain. In dynamic settings, temporal consistency under motion remains challenging [HSH24, ZYY24], and generalization across poses, garments, and body shapes is still limited by scarce paired supervision [ZLZS20, PKD19, WZL*19]. Facial fidelity and material heterogeneity (skin, hair, fabric) continue to challenge unified modeling [OLY*17, DCX*18, ZWC*24], while some diffusion-guided pipelines improve realism at increased computational cost [LZT*24, LWG*25].

Looking forward, we foresee progress toward more transferable canonical UV representations across subjects [CYF22, LZT*24]; part- and seam-aware generators with geometry guidance for complex garments [NKML25, LWG*25]; reference- and text-conditioned editing with identity preservation [WZL*19, ZLZ*24]; and scalable appearance and material modeling that extends beyond RGB [ZWC*24]. Achieving these at real-time or near-interactive rates, while maintaining temporal stability, remains a central challenge for neural 3D human texturing.

4.5. Commercial Systems and Technical Reports

In addition to the peer-reviewed literature surveyed above, numerous commercial systems, technical reports, and recent preprints have emerged for end-to-end 3D asset generation, often including mesh texturing as a key component [ZLL*25, HYY*25b, FLL*25, ZWZ*24, LZS*25, BKA*24, Hyp, Tri, Mes]. These systems are typically released as hosted products, API services, preprints, or technical write-ups, and therefore differ from academic work in their disclosure level and evaluation practices. We include them here to contextualize research progress with real-world deployment; unless noted otherwise, many sources cited in this subsection are non-peer-reviewed project pages, documentation, or technical reports.

A common theme across these systems is that *texturing is only one component of a larger pipeline* that jointly targets geometry, UVs, and material appearance. For example, Hunyuan3D [ZLL*25, HYY*25b] describes a two-stage design that first produces a 3D shape and then "paints" appearance to obtain a high-fidelity textured asset, with later iterations explicitly incorporating production-oriented material outputs (*e.g.*, PBR-oriented variants) alongside geometry generation. Similarly, Seed3D [FLL*25] presents a modular pipeline in which geometry generation is paired with multi-view texture/material synthesis and a subsequent UV completion stage to support downstream use in standard 3D toolchains. CLAY [ZWZ*24] is a controllable end-to-end 3D asset generator that couples a large 3D geometry model with a multi-view material diffusion module to produce 2K PBR material maps, illustrating how production-oriented systems integrate texturing into the full asset pipeline.

Commercial services such as Rodin [Hyp], Tripo [Tri], and Meshy [Mes] illustrate how these ideas are operationalized in practice: they emphasize a streamlined user experience (single-image or text-conditioned generation, export-ready textured assets, and direct export to common asset formats) and prioritize robustness across diverse user inputs. While implementation details vary and are not always fully disclosed, these systems broadly align with research trends highlighted throughout this survey: leveraging strong 2D generative priors (diffusion/transformers) for appearance, enforcing multi-view agreement to reduce view-dependent artifacts, and adopting standardized material conventions to ease integration into digital content creation (DCC) tools and real-time renderers.

From a research perspective, these deployments highlight several practical pressures that are likely to shape future texturing work. First, *production constraints*—such as predictable UV conventions, stable material parameterizations, and renderer compatibility—can matter as much as perceptual texture realism. Second, *system-level objectives* such as latency, scalability, failure modes, and controllability become central when texturing is embedded in an end-to-end generation stack rather than studied in isolation. Finally, because many commercial systems are distributed as services, their training data curation, preprocessing, and evaluation protocols are often less transparent than in academic releases, motivating reproducible benchmarks and standardized reporting for fair comparison.

Table 1: ***A summary of representative works in*** **Neural 3D Mesh Texturing**. *Each work is characterized by* ***Model*** *(the type of neural network being used),* ***Guidance*** *(the type of Stylistic Guidance which controls the texture appearance),* ***Model Type*** *(if the model is Pre-trained, Fine-tuned, or Custom trained for the texturing task),* ***Generation Strategy*** *(the way textures are generated: Optimization / Iterative / Synchronized / Feed-forward), and the output* ***Texture Type*** *(RGB textures, with baked-in lighting effects, or disentangled PBR materials).*

Methods	Model	Guidance	Model Type	Generation Strategy	Texture Type
TextureFields [OMN*19]	Neural fields	Uncond./Image	Custom	Feed-forward	RGB textures
Huang *et al.* [HTD*20]	GAN	Image	Custom	Optimization	RGB textures
LTG [YDPT21]	GAN	Unconditional	Custom	Feed-forward	RGB textures
Texturify [STM*22]	GAN	Unconditional	Custom	Feed-forward	RGB textures
Mesh2Tex [BTD23]	GAN+Neu. fields	Uncond./Image	Custom	Feed-forward+Opt.	RGB textures
ShaDDR [CCZZ23]	GAN	3D shape	Custom	Feed-forward	RGB textures
3DStyleNet [YGS*21]	GAN	3D shape	Pre-trained+Custom	Optimization	RGB textures
SPSG [DST*21]	GAN	Image	Custom	Feed-forward	RGB textures
AUV-Net [CYF22]	GAN	Uncond./Image/3D shape	Custom	Feed-forward+Opt.	RGB textures
Chaudhuri *et al.* [CSST21]	VAE+GAN	Uncond./Image	Pre-trained+Custom	Feed-forward	RGB textures
Lazova *et al.* [LIPM19]	GAN	Image	Custom	Feed-forward	RGB textures
Grigorev *et al.* [GSVL19]	GAN	Image	Custom	Feed-forward	RGB textures
Neverova *et al.* [NGK19]	GAN	Image	Custom	Feed-forward	RGB textures
UV-GAN [DCX*18]	GAN	Image	Custom	Feed-forward	RGB textures
NMR [KUH18]	VGG	Image	Pre-trained	Optimization	RGB textures
Dream Fields [JMB*22]	CLIP	Text	Pre-trained+Custom	Optimization	RGB textures
Text2Mesh [MBOL*22]	CLIP	Text	Pre-trained+Custom	Optimization	RGB textures
X-Mesh [MZS*23]	CLIP	Text	Pre-trained+Custom	Optimization	RGB textures
TANGO [CCL*22]	CLIP	Text	Pre-trained+Custom	Optimization	PBR materials
CLIP-Mesh [KXBP22]	CLIP	Text	Pre-trained	Optimization	RGB textures
TEXTure [RMA*23]	Diffusion (2D)	Text	Pre-trained	Iterative	RGB textures
Text2Tex [CSL*23]	Diffusion (2D)	Text	Pre-trained	Iterative	RGB textures
TexFusion [CKF*23]	Diffusion (2D)+Neu. fields	Text	Pre-trained+Custom	Synchronized + Opt.	RGB textures
Paint3D [ZCQ*24]	Diffusion (2D+UV)	Text/Image	Pre-trained+Custom	Iterative (+UV refine)	RGB textures
SyncMVD [LXLW24]	Diffusion (2D)	Text	Pre-trained	Synchronized	RGB textures
MVPaint [CMZ*25]	Diffusion (2D+UV)	Text	Pre-trained+Custom	Synchronized (+refine)	RGB textures
TexPainter [ZPZ*24]	Diffusion (2D)	Text	Pre-trained	Synchronized+Opt.	RGB textures
VCD-Texture [LYC*24]	Diffusion (2D)	Text	Pre-trained	Synchronized (+refine)	RGB textures
TexGen [HGZ*24]	Diffusion (2D)	Text	Pre-trained	Synchronized	RGB textures
GenesisTex [GJL*24]	Diffusion (2D)	Text	Pre-trained	Synchronized (+refine)	RGB textures
GenesisTex2 [LZZ*25]	Diffusion (2D)	Text	Pre-trained	Synchronized	RGB textures
FlexiTex [JYZ*25]	Diffusion (2D)	Text/Image	Pre-trained	Synchronized	RGB textures
RoomPainter [HYC*25]	Diffusion (2D)	Text	Pre-trained	Synchronized	RGB textures
InstanceTex [YGC*24]	Diffusion (2D)+Neu. fields	Text	Fine-tuned+Custom	Iterative+Opt.	RGB textures
RoomTex [WLX*24]	Diffusion (2D)	Text	Pre-trained	Iterative	RGB textures
Make-A-Texture [XGF*25]	Diffusion (2D)	Text	Pre-trained	Iterative	RGB textures
AlignTex [ZXW*25]	Diffusion (2D)	Image	Fine-tuned	Feed-forward+Sync.	RGB textures
TEXGen (700M) [YYG*24]	Diffusion (UV)	Text/Image	Custom	Feed-forward	RGB textures
Single-Mesh DM [MEM24]	Diffusion (on surface)	Unconditional	Custom	Feed-forward	RGB textures
Diffu. Tex. Paint. [HDAA*24]	Diffusion (2D)	Image/Brush	Fine-tuned	Feed-forward (interactive)	RGB textures
TextureDreamer [YHK*24]	Diffusion (2D)	Image	Fine-tuned+Custom	Optimization	PBR materials
StyleTex [XZT*24]	Diffusion (2D)	Image	Pre-trained	Optimization	RGB textures
FlashTex [DOW*24]	Diffusion (2D)	Text	Fine-tuned+Custom	Synchronized+Opt.	RGB textures
DreamMat [ZLX*24]	Diffusion (2D)	Text	Fine-tuned+Custom	Optimization	PBR materials
MaPa [ZPX*24]	Diffusion (2D)	Text	Fine-tuned+Custom	Optimization	PBR materials
Decorate3D [GZD*23]	Diffusion (2D)	Text/Image	Pre-trained+Custom	Optimization	RGB textures
3D Paintbrush [DLAH24]	Diffusion (2D)	Text	Pre-trained+Custom	Optimization	RGB textures
EASI-Tex [PWMAZ24]	Diffusion (2D)	Image	Pre-trained	Iterative	RGB textures
Paint-it [YJPMO24]	Diffusion (2D)	Text	Pre-trained+Custom	Optimization	PBR materials
Fantasia3D [CCJJ23]	Diffusion (2D)	Text	Pre-trained+Custom	Optimization	PBR materials
Material Anything [HWLW25]	Diffusion (2D/UV)	Text/3D Mesh	Feed-forward	Feed-forward	PBR materials
TexDreamer [LZT*24]	Diffusion (2D/UV)	Text/Image	Fine-tuned	Feed-forward	RGB textures
Make-It-Vivid [TZF*24]	Diffusion (2D/UV)	Text	Fine-tuned	Feed-forward	RGB textures
FabricDiffusion [ZWC*24]	Diffusion (2D)	Image	Fine-tuned	Feed-forward (UV/tile)	RGB textures
TexGarment [LWG*25]	Diffusion (2D/UV)	Text	Fine-tuned+Custom	Feed-forward	RGB textures
Point-UV Diffusion [YDL*23]	Diffusion (Point+UV)	Uncond./Text/Image	Custom	Feed-forward	RGB textures
SeqTex [YYS*25]	Diffusion (Video+UV)	Text/Image	Fine-tuned	Synchronized	RGB textures
RomanTex [FYY*25]	Diffusion (2D)	Image	Fine-tuned	Synchronized	RGB textures
MaterialMVP [HYY*25a]	Diffusion (2D)	Image	Fine-tuned	Synchronized	PBR materials
CLAY [ZWZ*24]	Diffusion (2D)	Text/Image	Custom	Synchronized	PBR materials
VideoMat [MWL*25]	Diffusion (Video+PBR)	Text/Image	Fine-tuned	Synchronized	PBR materials
DreamPBR [XZP*25]	Diffusion (2D+PBR)	Text+Multimodal	Fine-tuned+Custom	Feed-forward	PBR materials

5. Datasets and Evaluation Metrics

This section reviews the datasets and evaluation metrics commonly used in neural methods that *directly* texture 3D meshes. We focus on datasets that offer textured meshes or enable reliable supervision for mesh texturing, as well as standard metrics for evaluating the fidelity, consistency, and usability of textured assets.

5.1. Datasets

High-quality textured 3D meshes remain relatively scarce compared to geometry-only repositories. As a result, most works rely on one or more of three data sources: (i) curated datasets of textured meshes; (ii) geometry-only collections combined with external image datasets or procedural materials; and (iii) large-scale, web-scraped 3D asset libraries with varying texture quality and licensing. Below, we summarize representative datasets grouped by content type.

5.1.1. Mesh Datasets

General Object Mesh Datasets. Early repositories such as ModelNet [WSK*15] and ShapeNet [CFG*15] contain thousands of 3D CAD models but are mostly untextured or only sparsely textured. Similarly, Thingi10K [ZJ16] includes 10,000 3D-printable models, emphasizing geometric diversity over material realism. The Princeton COSEG dataset [WAvK*12] provides segmented 3D models across object categories for shape analysis, but with little or no texture information. PASCAL-3D+ [XMS14] bridges 2D and 3D domains by aligning 3D CAD models with real images for object detection and pose estimation, but it does not provide high-quality textured meshes. In contrast, PhotoShape [PRFS18] augmented a subset of ShapeNet [CFG*15] with photorealistic materials inferred from internet photos, yielding a valuable resource for learning-based rendering and texturing research.

Recent datasets explicitly emphasize diverse, high-quality object textures. 3D-FUTURE [FJG*21] contains about 10,000 furniture CAD models with high-resolution textures and rich annotations, targeting household objects in indoor scenes. It is often paired with the 3D-FRONT scene dataset [FCG*21], which provides complete room layouts furnished with 3D-FUTURE assets to support indoor-scene synthesis and related texture-transfer research. The Amazon-Berkeley Objects (ABO) dataset [CGD*22] likewise offers thousands of product models with realistic geometry and physically based materials, helping bridge synthetic and real-world object understanding. At larger scale, Objaverse [DSS*23] and its successor Objaverse-XL [D*23] aggregate millions of 3D models across highly diverse categories; although not all models feature high-quality textures (Fig. 23), these web-scale collections have become valuable resources for pre-training and large-scale generative 3D/texturing research. The recently introduced TexVerse dataset [ZZMC25] further advances both scale and quality, curating over 850K unique 3D objects with high-resolution textures, including over 158K with full PBR maps, as well as specialized subsets of rigged and animated textured models, making it a promising resource for learning neural texturing across object types.

For research requiring scenes, the Matterport3D dataset [CDF*17] provides 90 real indoor scenes with RGB-D captures, surface reconstructions, and textured meshes, making it a valuable resource for methods that leverage scene-level geometry and appearance. Some synthetic datasets focus on specialized use cases; for example, Houses3K [PCN*20] contains 3,000 procedurally generated house models with multiple texture variants, originally created to train next-best-view policies for 3D reconstruction. In summary, a broad spectrum of mesh datasets now exists, from early geometry-only repositories to modern collections with curated textures, which can be combined to train and evaluate neural texturing methods.

Figure 23: *Objaverse-XL provides over 10M 3D objects spanning diverse categories, enabling large-scale training and benchmarking for mesh texturing. Figure reproduced from [D*23].*

3D Humans and Garments. Textured 3D human models present unique challenges, and relatively few large public datasets are available. Several commercial or restricted 3D human resources exist, including RenderPeople [Ren25], Triplegangers [Tri25], Treedy [Tre25], and Twindom [Twi25]. These resources provide high-fidelity scanned humans or avatar assets with realistic textures, but they are typically proprietary or only partially accessible. Recent academic efforts have instead focused on larger-scale human texture datasets. For example, TexDreamer's ATLAS dataset [LZT*24], described as the "largest high-resolution 3D human texture dataset", offers diverse human UV textures for zero-shot generative modeling. Although ATLAS has not been fully released, it enabled TexDreamer's high-quality results in 3D human texturing. For clothed virtual characters, the 3DBiCar dataset [LCD*23] provides 1,500 fully textured cartoon-style human meshes spanning 15 character species.

Many works also leverage 2D data to compensate for the scarcity of textured 3D human scans. For instance, single- or multi-view images of people, as well as curated internet photos of clothing, are often used to texture 3D human models [MAPM20, CSAN23]. Therefore, while large-scale public datasets of textured 3D humans remain limited, the community is gradually assembling both realistic (scanned) and synthetic (artistic or generated) texture collections to advance learning-based texturing of people and garments.

5.1.2. Image Datasets

In addition to 3D datasets, many methods rely on 2D image datasets for learning or evaluation, especially when only geometry is available. For example, image-guided texture generation for specific object categories uses object image collections such as BrnoCompSpeed [SJŠ*19], which includes over 20k real car images with ground-truth annotations, and CompCars [YLLT15], which provides 214k car images spanning make and model variations, to synthesize realistic car paint or decal textures. Bird texture transfer methods likewise employ the CUB-200-2011 dataset [WBW*11] to map fine-grained feather patterns onto 3D bird models. Generic texture descriptors are often learned from the Describable Textures Dataset (DTD) [CMK*14], containing 5,640 images of patterned textures labeled with human-descriptive attributes. For texture super-resolution or recovering high-frequency details, high-resolution datasets such as DIV2K [AT17], with 1,000 2K-resolution images, are commonly used as benchmarks [RCO*19].

Several large-scale human-centric image datasets have been instrumental in learning to texture people and clothing. The DeepFashion dataset [LLQ*16], with over 800k clothing images annotated with attributes and landmarks, and Market-1501 [ZST*15], comprising 32,000+ multi-camera person images, serve as abundant 2D sources of in-the-wild clothing appearance and are often used as supervision or priors for methods that transfer garment appearance onto 3D human meshes. The MVC dataset [LCC16] provides multi-view photographs of about 37,000 clothing items with attribute labels, enabling learning of view-invariant clothing texture representations. For faces, the FFHQ dataset [KLA19] (70,000 high-quality human face images) is often used to learn realistic facial appearance priors, including for facial texture or albedo synthesis. Two additional datasets link images to human UV texture maps: DensePose-COCO [GNK18] augments COCO images [LMB*14] with dense body-surface correspondences and part-specific UV coordinates, facilitating supervised texture transfer and completion, while the CMU Multi-PIE dataset [GMC*08] provides multiview facial images under varying illumination, supporting the learning of consistent UV facial textures across viewpoints.

Finally, beyond established datasets, many works curate task-specific data from internet imagery to suit particular applications—*e.g.*, assembling custom clothing-image collections for texture exemplars or constructing custom 2D–3D correspondences [ZWC*24, YDPT21]. Such one-off datasets highlight the community's continuing need to fill gaps in data availability.

5.1.3. Dataset Preprocessing and Filtering

Meshes. Large-scale mesh repositories collected from the web (*e.g.*, Objaverse/Objaverse-XL [DSS*23, D*23]) exhibit substantial variation in geometry, UV parameterizations, texture conventions, and asset completeness, making careful preprocessing and filtering important for stable training and fair evaluation. For instance, even widely used collections contain many assets with missing or low-resolution textures, and reported cleaning pipelines often enforce minimum texture-resolution thresholds, exclude assets with restrictive usage tags (*e.g.*, terms related to "NoAI"), and restrict the dataset to redistributable Creative Commons licenses before standardizing formats such as `.glb` for downstream use [ZZMC25, LZS*25, SXSX25]. Beyond basic sanity checks, practical mesh filtering commonly includes: (i) *geometry quality* checks (degenerate faces, corrupted topology, extreme scale/units, missing normals/material assignments); and (ii) *UV validity* checks (presence of UVs, invalid/NaN coordinates, severe overlaps, excessive fragmentation, or extremely low texel density), since UV pathologies can directly translate into artifacts or unstable supervision when learning in UV space.

Recent texturing pipelines provide concrete examples of such curation. TexGen [HGZ*24] reports that web meshes come with inconsistent "texture structures" and quality; it filters poor texture cases, re-unwraps meshes to a new parameterization, and bakes diffuse color into the new UVs to obtain a consistent training representation. For PBR-oriented supervision, Material Anything [HWLW25] constructs Material3D by filtering Objaverse [DSS*23] meshes to retain only assets with a sufficiently complete material-map set (*e.g.*, base color, roughness, metallic, and bump), then re-unwrapping and consolidating parts to produce UV-ready training data. Such steps also reduce confounds in evaluation: improvements in a learned model should not be attributable to inconsistent UV layouts, missing maps, or broken assets.

A key additional consideration is *PBR availability and imbalance*. High-quality multi-map materials are substantially less common than albedo-only textures or shaded renders; for example, TexVerse [ZZMC25] curates 858K high-resolution textured models, but only a subset (158K) contains PBR materials under standard metalness–roughness or specular–glossiness workflows with the requisite roughness/glossiness and metalness/specular channels. This scarcity creates an inherent data imbalance between RGB-only supervision and fully relightable PBR supervision, and motivates reporting dataset statistics (#assets with UVs, #with albedo-only, #with full PBR sets, map resolutions) as well as clearly stating conversion choices (*e.g.*, metalness–roughness vs. specular–glossiness, map packing, and color-space conventions). Overall, documenting preprocessing decisions helps readers interpret results and improves reproducibility across datasets and renderers.

Images. When methods rely on reference or conditioning images (*e.g.*, as references [PWMAZ24]), basic curation helps prevent the image domain from becoming a hidden source of artifacts and bias. Common steps include (i) foreground isolation via segmentation/matting (to avoid background leakage into the synthesized texture), (ii) cropping to a region of interest and resizing to a consistent resolution, and (iii) filtering for excessive occlusion, extreme viewpoints, or low sharpness [PWMAZ24, STM*22]. Some pipelines additionally leverage region-level crops or masks to isolate the relevant appearance signal (*e.g.*, material/print patterns) and reduce interference from background clutter or occlusions; for example, FabricDiffusion [ZWC*24] conditions on a clothing image together with region captures of its fabric materials and prints to extract normalized textures/prints, which are then mapped onto the target garment UVs. Overall, while image preprocessing is typically lighter-weight than mesh filtering, these steps improve robustness and reduce spurious texture transfer driven by backgrounds or irrelevant context.

5.2. Evaluation Metrics

For neural 3D mesh texturing, ground-truth data are often unavailable. As a result, acceptable outputs vary by goal (*e.g.*, transfer, alignment, or conditioned synthesis), leading to task-specific and non-standardized evaluations. In practice, most works report a combination of image-based and asset-level criteria, assessing *distribution-level realism*, *per-instance fidelity*, and *semantic alignment*, often complemented by user studies. For controlled comparison, evaluations are typically conducted on multi-view renderings with fixed camera sets and lighting or environment maps, with background masking when comparing to real photographs to isolate object appearance. Robustness is also tested across mesh types (organic *vs.* CAD-like, generated *vs.* artist-authored) and is often accompanied by runtime analysis for practical relevance [CKF*23, RMA*23, XZT*24, LXLW24]. Below, we mark ↑ where higher metric values indicate better performance and ↓ where lower values are better.

Appearance Alignment. To measure how closely the distribution of generated textured images matches real images or ground-truth textures, authors often rely on GAN-style distribution metrics. The Fréchet Inception Distance (FID; ↓) [HRU*17] computes the 2-Wasserstein distance [Vil09, DL82, Gel90] between Gaussian approximations of deep feature distributions (typically extracted using an Inception-V3 network [SVI*16]) for rendered results and reference images. Lower FID indicates that the generated distribution is closer to the reference distribution. Similarly, the Kernel Inception Distance (KID; ↓) [BSAG18] is an unbiased MMD-based [GBR*12] metric comparing sets of Inception features; it is often preferred for smaller sample sizes. The Inception Score (IS; ↑) [SGZ*16] measures both confidence and diversity by feeding generated images into a pretrained classifier: sharp, class-consistent predictions for individual images together with high entropy over the marginal label distribution yield a higher IS. These distribution-level metrics are most common when a unique ground-truth texture is unavailable (*e.g.*, style transfer) and realism or diversity should match a target distribution. We note, however, that their usefulness can vary: *e.g.*, FID and IS are less sensitive to spatial alignment or fine geometric detail, and reliable estimates generally require sufficiently large sample sizes to reduce statistical noise. Therefore, they are usually supplemented by more granular metrics as described next.

Instance-Level Reconstruction Fidelity. When a specific target texture or image is available for each 3D model (*e.g.* in texture super-resolution or image-based texturing tasks), evaluation can treat the problem as an image reconstruction task, and standard pixel-wise or structural metrics from image processing can be employed. Peak Signal-to-Noise Ratio (PSNR; ↑) [ZIE*18] and the Structural Similarity Index (SSIM; ↑) [WBSS04, WSB03] are frequently reported to quantify low-level fidelity—higher PSNR or SSIM on rendered views indicates that the synthesized texture preserves more detail and structure from the ground truth. Because these pointwise metrics often fail to reflect perceptual quality, many works also use learned perceptual distances; for example, the Learned Perceptual Image Patch Similarity (LPIPS; ↓) [ZIE*18], which measures distance using deep features, is a popular gauge of human-perceived closeness (lower LPIPS means more perceptually alike), especially in texture super-resolution. If the task involves semantic or part-aware accuracy, and ground-truth regions are available (such as in Texture Alignment [CYF22]), Intersection-over-Union (IoU; ↑) or other overlap measures can be used to evaluate how well the predicted textures align with those regions.

Prior-Guided Semantic Alignment. In many neural texturing scenarios, the goal is for the output texture to match some input descriptions such as a text prompt, a style image, or the visual characteristics of a source domain. Here, evaluation leverages pretrained models as "priors" to score alignment. A common choice is CLIP [R*21]: the CLIP Score (image–text cosine similarity; ↑) [HHF*21] measures how well a rendered image of the textured mesh matches a given text prompt in CLIP's joint embedding space, while CLIP-Var (image–image cosine similarity; ↑) [LFW*24] evaluates the consistency of multi-view renderings of the textured mesh. Likewise, for text-to-texture tasks, authors also report CLIP R-Precision (↑) [PAL*21], which measures whether the correct text prompt is retrieved among a set of distractors using CLIP embeddings. In practice, this retrieval-based metric serves as a reasonable proxy for human judgments of text–image correspondence [LYC*24]. Additionally, some works report the *CLIP–Aesthetic Score* (↑), which is a no-reference regressor over CLIP embeddings trained to predict perceived visual appeal. This complements semantic alignment metrics but does not directly assess alignment with a prompt [Sch22].

Beyond CLIP, other perceptual or semantic priors also inform evaluation. Some works compute feature-space distances such as Feature-ℓ_1/ℓ_2 (↓) between rendered outputs and target images using pretrained CNN features (*e.g.*, VGG [SZ15]). These distances provide a perceptual or semantic fidelity measure and are widely used in settings such as style transfer and image-based optimization [GEB16, KUH18]. No-reference image quality models are also repurposed to assess textures: the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE; ↓) [MMB12] measures deviations from natural scene statistics, so lower BRISQUE generally indicates more natural-looking outputs. Similarly, Generative Image Quality Assessment (GIQA; ↑) [GBCW20] uses a learned model to predict the quality of each generated image. A higher GIQA score indicates better per-image quality, complementing FID's dataset-level view.

Very recently, researchers have begun leveraging multimodal large language models for evaluation. One example is the MLLM Score (↑) [HSX*23, HDS*25], which prompts a vision–language model to assess how well an image satisfies a textual instruction or description. Finally, we note that certain metrics target specific texture properties. For instance, TexTile (↓) [RPCGLM24] is a learned metric designed to quantify the tileability of a texture (*i.e.*, whether it can repeat without visible seams) [ZWC*24]. In summary, prior-guided metrics enable evaluation of semantic correctness, style consistency, proxy visual realism, and multi-view consistency, using powerful pretrained models to capture what simple pixel metrics cannot.

Human Perceptual Evaluation (User Studies). Given the inherent ambiguities of the task, the ultimate assessment of texture qual-

ity often relies on human perception. Many surveyed works include user studies to compare perceptual quality and user preference [PWMAZ24, RMA*23, CSL*23]. Despite the variety of quantitative metrics, user studies remain essential for evaluating perceptual realism and semantic satisfaction, though they are subjective and not perfectly reproducible. Common strategies to reduce bias include diversifying participants and evaluation tasks, randomizing presentation order, and carefully controlling comparison protocols. Typical study designs include pairwise A/B preference tests reporting the *win rate* (↑), ranking tasks reporting the *average rank* (↓), and absolute rating studies yielding a *mean opinion score (MOS)* (↑) [ITU19].

User studies are typically reported across multiple aspects: *overall quality*; *structural guidance alignment* (respecting mesh semantics during texturing) [XZT*24]; *stylistic guidance alignment*; *level of detail* and *presence of artifacts* [CKF*23]; *seam visibility*; *diversity*; and *3D consistency* across views [LXLW24]. In practice, user studies are most informative when interpreted alongside quantitative metrics.

In a field where no single metric is sufficient, a multifaceted evaluation strategy that combines distribution-level statistics, per-instance accuracy, semantic alignment, user studies, runtime analysis, and robustness tests across different inputs offers the most holistic view of a method's performance. Each class of metrics captures a distinct aspect of the textured output, and together they reveal the complementary strengths and trade-offs of neural 3D texturing approaches in the literature.

6. Applications

Neural 3D mesh texturing supports a broad range of task settings and downstream applications. Research has progressively addressed complementary problem settings, including synthesizing textures from text [RMA*23, CSL*23] or images [PWMAZ24], estimating physically based (PBR) materials [CCL*22, ZLX*24], aligning category-level UVs [CYF22], completing missing appearance on partial scans [CPM20], localized texturing [DLAH24], and scene-scale texturing [WLX*24, HYC*25, YGC*24]. Beyond benchmarking, these advances map naturally to practical use cases across content creation, telepresence, visualization, simulation, gaming, and fabrication, where the final output is often a UV-parameterized mesh—and increasingly a full PBR material stack—whose appearance can be edited, relit, and rendered efficiently (Fig. 24):

- **Asset creation for games, XR, and robotics.** Text- and image-guided pipelines generate high-quality textures on fixed meshes from high-level prompts or exemplars, accelerating look development, style exploration, and re-skinning; representative systems include prompt-driven and diffusion-guided UV synthesis [RMA*23, CSL*23, CKF*23, LXLW24]. Outputs integrate with real-time engines via PBR material maps (albedo/roughness/normal, *etc.*), enabling fast relighting and consistent deployment across platforms [CCL*22, CCJJ23]. In practice, teams can use these models to produce on-brand variants (*e.g.*, seasonal skins) while preserving decals and layout fidelity inherited from the mesh UVs [RMA*23, CSL*23]. More broadly, diversified textures on scene meshes are often used in simulation to improve sim-to-real transfer for perception and control in robotics [TFR*17].
- **Digital humans and telepresence.** Neural textures for deformable meshes offer temporally stable, high-fidelity avatar rendering, reenactment, and appearance editing, making them relevant to telepresence and virtual production [TZN19]. Related pipelines transfer garment patterns and prints to clothed-body meshes while respecting canonical UVs or part correspondences, supporting virtual try-on and fashion prototyping [NKML25, ZWC*24]. When PBR is required, texturing methods that estimate SVBRDF channels enable realistic relighting and integration into real or virtual environments [CCL*22, ZLX*24].
- **E-commerce and product visualization.** For product visualization, texturing techniques enable fabric, leather, or finish swaps on a single mesh, allowing customized previews and photoreal variants. Diffusion- and image-conditioned approaches can reduce manual authoring time while preserving brand constraints and UV semantics [PWMAZ24, YHK*24].
- **3D stylization and look development.** Optimization and feed-forward methods that modify appearance directly on 3D surfaces or meshes enable consistent, multi-view stylization (*e.g.*, painterly, toon, or brand-specific styles) with artist-in-the-loop control; coupling semantic guidance (text/image) with structural cues (normals/depth) improves fidelity and geometry awareness for interactive authoring [MBOL*22, HJN22, PWMAZ24].

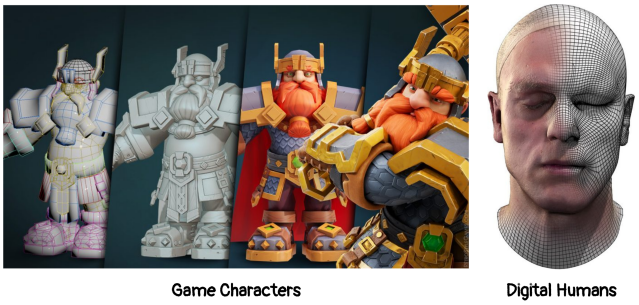

Figure 24: *Applications of 3D mesh texturing in games and VFX. Textures add material detail and realism to production assets such as game characters (left) and digital humans (right). Figure adapted from [Dem24, Wol17].*

7. Limitations

While 3D mesh texturing has advanced rapidly with the advent of diffusion-driven generative pipelines, several important challenges remain:

- **Incomplete texturing from multi-view projection.** Methods that texture a mesh by iteratively projecting and inpainting from multiple rendered views—e.g., TEXTure, Text2Tex, and TexFusion—are effective and convenient because they leverage off-the-shelf, pre-trained text-to-image diffusion models and require little or no task-specific training. However, they inherently cover only camera-visible regions in each step; self-occluded or rarely visible areas may remain underspecified and require post-

hoc fixes. Even with heuristics for next-best-view scheduling and partial texture masks, view-to-view inconsistency and residual holes are common failure modes [RMA*23, CSL*23, CKF*23, LXLW24]. Recent formulations that synchronize multi-view denoising improve global consistency, but do not fully eliminate occlusion-driven gaps on challenging geometry [LXLW24].

- **Small and imperfect open-source datasets for *textured* assets.** Large-scale 3D repositories (e.g., Objaverse/Objaverse-XL) provide breadth and diversity, yet textures are heterogeneous in quality, licensing, and parameterization, and rarely include physically based (PBR) channels at scale [DSS*23, D*23]. Dedicated material datasets exist—from measured SVBRDFs to modern CC0 PBR collections—but they typically provide *2D* material assets (planar patches) rather than curated, high-quality, per-*mesh* UV textures suitable for training texturing models end-to-end [MXZ*23, VD24]. As a result, supervision remains limited for learning high-fidelity, mesh-aligned appearance.
- **Generalization and factorization of appearance.** Models trained on small or weakly curated datasets can bake view-dependent effects (specularities, shadows) into albedo and struggle to predict physically meaningful material maps (albedo/roughness/metalness) that generalize across objects and lighting. Progress in neural inverse rendering and reflectance decomposition underscores both the promise and the difficulty of robust, disentangled estimation under unknown illumination and complex geometry [ZSD*21, BBJ*21, MXZ*23]. Foundational, broadly generalizable material estimators for in-the-wild meshes remain an open problem.
- **Computational cost and memory footprint.** High-resolution UVs (multi-UDIM, 8–16 K) and multi-view diffusion sampling are expensive in time and GPU memory. Although distillation and consistency-model techniques offer a path to reducing sampling steps, pipelines that couple high-resolution UV baking with multi-view regularization are still resource-intensive, especially in academic settings [LTH*23, CKF*23, CSL*23].

8. Conclusion and Future Work

Neural 3D mesh texturing has rapidly developed into a vibrant research area, with growing impact on 3D asset creation for VFX, e-commerce, advertising, gaming, and related industries. Recent advances in diffusion models, vision–language priors, and differentiable rendering have significantly expanded the capabilities of automated texture generation, enabling workflows that are more scalable, expressive, and less reliant on manual intervention.

In this survey, we presented a comprehensive overview of this field, categorizing methods into foundational neural approaches, optimization-based methods, and accelerated diffusion-based pipelines, while analyzing their design choices in terms of supervision, guidance, and architectural patterns. Alongside a review of current datasets, evaluation strategies, and practical applications, we identified key limitations related to occlusion-aware synthesis, dataset quality, the generalization and factorization of physical appearance, and computational cost.

Looking ahead, future work in neural mesh texturing will likely expand along multiple directions, including greater scale, dynamic and deformable settings, and continued improvements in quality, controllability, and interactivity, alongside related advances in part-aware 3D understanding [PVN*25]. We outline a few concrete problems below:

- **Texturing 3D scenes.** Compared with individual 3D objects, scenes exhibit greater geometric diversity and looser structural relationships among constituent objects. This added complexity makes scene-level texturing more challenging, particularly when enforcing appearance consistency across objects, materials, and spatial context.
- **Geometry-aware view planning and coverage.** Beyond heuristic camera schedules, optimization- or learning-based view planning that explicitly maximizes surface coverage and uncertainty reduction could mitigate self-occlusion while minimizing redundant renderings. Integrating coverage metrics with synchronized multi-view denoising may further reduce inconsistency [CSL*23, LXLW24].
- **Dynamic textures and materials.** With video diffusion models maturing, generating temporally coherent animated textures (e.g., flowing patterns or wear over time) is a natural extension. Ensuring temporal stability across UV seams and under motion, including deforming meshes, poses new challenges in conditioning, regularization, and evaluation [KKW*23].
- **Deformation- and correspondence-aware texturing for dynamic meshes.** Extending static pipelines to articulated or topology-varying meshes requires correspondence-robust mapping across frames or poses and appearance transport that respects stretch, compression, and contact. Combining deformation-aware parameterization with multi-view diffusion priors is a promising direction.
- **Scalability and speed.** Practical adoption benefits from faster sampling (few-step or distilled samplers), tile- and patch-based UV generation with seamless blending, and memory-efficient training and inference for multi-UDIM assets. Recent latent consistency approaches suggest a promising path toward substantial speedups while preserving fidelity [LTH*23].
- **Faithful material decomposition and relightability.** Joint estimation of mesh-aligned albedo, normal, roughness, metallicity, and environment illumination remains difficult at scale. Hybrid pipelines that couple diffusion priors with physically motivated inverse rendering, and that train against measured or high-quality synthetic SVBRDF corpora, could yield more robust and relightable assets [ZSD*21, BBJ*21, MXZ*23].

Last but not least, there is still a pressing need for curated, rights-cleared benchmarks of textured *meshes* with high-resolution UVs and, ideally, PBR channels, paired with standardized metrics for multi-view consistency, seam visibility, and perceptual quality under novel lighting and viewpoints. Recent dataset efforts provide important building blocks, but they do not yet provide mesh-aligned supervision at scale [DSS*23, MXZ*23, VD24].

References

[ACB17] ARJOVSKY M., CHINTALA S., BOTTOU L.: Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning* (2017), vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 214–223. URL: https://proceedings.mlr.press/v70/arjovsky17a.html. 7

[ado25] Adobe Substance 3D Painter, 2025. Accessed: 2025-09-03. URL: https://www.adobe.com/products/substance3d/apps/painter.html. 12

[AFL23] ATTAR H. R., FOSTER A., LI N.: Implicit Neural Representations of Sheet Stamping Geometries with Small-Scale Features. *Engineering Applications of Artificial Intelligence 123* (2023), 106482. doi:10.1016/j.engappai.2023.106482. 3

[AHH*18] AKENINE-MÖLLER T., HAINES E., HOFFMAN N., PESCE A., IWANICKI M., HILLAIRE S.: *Real-Time Rendering, Fourth Edition*. A K Peters/CRC Press, 2018. doi:10.1201/b22086. 3, 4

[AT17] AGUSTSSON E., TIMOFTE R.: NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2017), pp. 1122–1131. doi:10.1109/CVPRW.2017.150. 20

[BBJ*21] BOSS M., BRAUN R., JAMPANI V., BARRON J. T., LENSCH H. P.: NeRD: Neural Reflectance Decomposition from Image Collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12684–12694. doi:10.1109/ICCV48922.2021.01249. 23

[BKA*24] BENSADOUN R., KLEIMAN Y., AZURI I., HAROSH O., VEDALDI A., NEVEROVA N., GAFNI O.: Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430* (2024). URL: https://arxiv.org/abs/2407.02430, doi:10.48550/arXiv.2407.02430. 17

[BKM17] BLEI D. M., KUCUKELBIR A., MCAULIFFE J. D.: Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association 112*, 518 (2017), 859–877. doi:10.1080/01621459.2017.1285773. 6

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. A K Peters/CRC Press, 2010. doi:10.1201/b10688. 3

[BL08] BURLEY B., LACEWELL D.: Ptex: Per-Face Texture Mapping for Production Rendering. *Computer Graphics Forum (Proc. EGSR) 27*, 4 (2008), 1155–1164. doi:10.1111/j.1467-8659.2008.01253.x. 4

[Ble25] BLENDER FOUNDATION: *Blender Reference Manual*, 2025. Accessed: 2025-09-03. URL: https://docs.blender.org/manual/en/latest/. 12

[BN25] BANNISTER J. J., NOWROUZEZAHRAI D.: Differentiable Visual Computing: Bridging 2D and 3D in Machine Learning Applications. mila.quebec, Jan. 2025. Blog article, Mila – Quebec AI Institute. 4

[BSAG18] BIŃKOWSKI M., SUTHERLAND D. J., ARBEL M., GRETTON A.: Demystifying MMD GANs. In *International Conference on Learning Representations* (2018). URL: https://openreview.net/forum?id=r1lUOzWCW. 21

[BTD23] BOKHOVKIN A., TULSIANI S., DAI A.: Mesh2Tex: Generating Mesh Textures from Image Queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 8918–8928. doi:10.1109/ICCV51070.2023.00819. 2, 11, 18

[Bur12] BURLEY B.: Physically-Based Shading at Disney. SIGGRAPH Course Notes, 2012. Walt Disney Animation Studios. URL: https://disneyanimation.com/publications/physically-based-shading-at-disney/. 5

[BW99] BORN M., WOLF E.: *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7 ed. Cambridge University Press, 1999. 5

[BZ17] BARNES C., ZHANG F.-L.: A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media 3*, 1 (2017), 3–20. doi:10.1007/s41095-016-0064-2. 2

[CCC*20] CAI L., CHEN Y., CAI N., CHENG W., WANG H.: Utilizing Amari-Alpha Divergence to Stabilize the Training of Generative Adversarial Networks. *Entropy 22* (04 2020), 410. doi:10.3390/e22040410. 7

[CCJJ23] CHEN R., CHEN Y., JIAO N., JIA K.: Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 22246–22256. doi:10.1109/ICCV51070.2023.02033. 2, 7, 12, 18, 22

[CCL*22] CHEN Y., CHEN R., LEI J., ZHANG Y., JIA K.: TANGO: Text-driven Photorealistic and Robust 3D Stylization via Lighting Decomposition. In *Advances in Neural Information Processing Systems* (2022), pp. 30923–30936. URL: https://openreview.net/forum?id=zbuq101sCNV. 2, 6, 11, 18, 22

[CCZZ23] CHEN Q., CHEN Z., ZHOU H., ZHANG H.: ShaDDR: Interactive Example-Based Geometry and Texture Generation via 3D Shape Detailization and Differentiable Rendering. In *Proceedings of SIGGRAPH Asia* (2023). doi:10.1145/3610542.3626131. 11, 18

[CDF*17] CHANG A., DAI A., FUNKHOUSER T., HALBER M., NIESSNER M., SAVVA M., SONG S., ZENG A., ZHANG Y.: Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)* (2017), 667–676. doi:10.1109/3DV.2017.00081. 19

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: *ShapeNet: An Information-Rich 3D Model Repository*. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. URL: https://arxiv.org/abs/1512.03012, arXiv:1512.03012. 9, 10, 15, 19

[CGD*22] COLLINS J., GOEL S., DENG K., LUTHRA A., XU L., GUNDOGDU E., ZHANG X., YAGO VICENTE T. F., DIDERIKSEN T., ARORA H., GUILLAUMIN M., MALIK J.: ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. *CVPR* (2022), 21094–21104. doi:10.1109/CVPR52688.2022.02045. 19

[CGL*19] CHEN W., GAO J., LING H., SMITH E. J., LEHTINEN J., JACOBSON A., FIDLER S.: Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer. In *Advances in Neural Information Processing Systems* (2019), vol. 32. URL: https://papers.nips.cc/paper_files/paper/2019/hash/f5ac21cd0ef1b88e9848571aeb53551a-Abstract.html. 4

[CKF*23] CAO T., KREIS K., FIDLER S., SHARP N., YIN K.: TexFusion: Synthesizing 3D Textures with Text-Guided Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4191–4202. doi:10.1109/ICCV51070.2023.00385. 13, 14, 15, 18, 21, 22, 23

[CLL*24] CHEN D. Z., LI H., LEE H.-Y., TULYAKOV S., NIESSNER M.: SceneTex: High-Quality Texture Synthesis for Indoor Scenes via Diffusion Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.01992. 12

[CMK*14] CIMPOI M., MAJI S., KOKKINOS I., MOHAMED S., VEDALDI A.: Describing Textures in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2014), pp. 3606–3613. doi:10.1109/CVPR.2014.461. 20

[CMZ*25] CHENG W., MU J., ZENG X., CHEN X., PANG A., ZHANG C., WANG Z., FU B., YU G., LIU Z., ET AL.: Mvpaint: Synchronized multi-view diffusion for painting anything 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025), pp. 585–594. doi:10.1109/CVPR52734.2025.00063. 14, 18

[CPM20] CHIBANE J., PONS-MOLL G.: Implicit Feature Networks for Texture Completion from Partial 3D Data. In *European Conference on Computer Vision Workshops* (2020), Springer, pp. 717–730. doi:10.1007/978-3-030-65414-6_43. 10, 22

[CSAN23] CHA S., SEO K., ASHTARI A., NOH J.: Generating Texture for 3D Human Avatar from a Single Image Using Sampling and Refinement Networks. *Computer Graphics Forum 42*, 2 (2023), 385–396. doi:10.1111/cgf.14769. 19

[CSL*23] CHEN D. Z., SIDDIQUI Y., LEE H.-Y., TULYAKOV S., NIESSNER M.: Text2Tex: Text-driven Texture Synthesis via Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 18558–18568. doi:10.1109/ICCV51070.2023.01701. 2, 8, 9, 11, 13, 14, 15, 18, 22, 23

[CSS*25] CHEN Y., SHAO G., SHUM K. C., HUA B.-S., YEUNG S.-K.: Advances in 3D Neural Stylization: A Survey. *International Journal of Computer Vision 133*, 17 (2025), 5026–5061. doi:10.1007/s11263-025-02403-9. 2

[CSST21] CHAUDHURI B., SARAFIANOS N., SHAPIRO L., TUNG T.: Semi-Supervised Synthesis of High-Resolution Editable Textures for 3D Humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7987–7996. doi:10.1109/CVPR46437.2021.00790. 15, 16, 17, 18

[CT81] COOK R. L., TORRANCE K. E.: A Reflectance Model for Computer Graphics. *ACM SIGGRAPH Computer Graphics 15*, 3 (1981), 307–316. doi:10.1145/965161.806819. 5

[CXG*16] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *European Conference on Computer Vision* (2016), pp. 628–644. doi:10.1007/978-3-319-46484-8_38. 3

[CYF22] CHEN Z., YIN K., FIDLER S.: AUV-Net: Learning Aligned UV Maps for Texture Transfer and Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 1465–1474. doi:10.1109/CVPR52688.2022.00152. 1, 9, 16, 17, 18, 21, 22

[D*23] DEITKE M., ET AL.: Objaverse-XL: A Universe of 10M+ 3D Objects. In *Advances in Neural Information Processing Systems* (2023). doi:10.5555/3666122.3667676. 14, 15, 19, 20, 23

[DAD*18] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-Image SVBRDF Capture with a Rendering-Aware Deep Network. *ACM Transactions on Graphics 37*, 4 (2018), 128:1–128:15. doi:10.1145/3197517.3201378. 5

[DCX*18] DENG J., CHENG S., XUE N., ZHOU Y., ZAFEIRIOU S.: UV-GAN: Adversarial Facial UV Map Completion for Pose-Invariant Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 7093–7102. doi:10.1109/CVPR.2018.00741. 15, 16, 17, 18

[Dem24] DEMIANENKO Y.: How to Make 3D Models for Games: 10 Quick Tips. RetroStyle Games Blog, July 2024. Published July 9, 2024. Accessed January 25, 2026. URL: https://retrostylegames.com/blog/how-to-make-3d-models-for-games/. 22

[DG01] DISCHLER J.-M., GHAZANFARPOUR D.: A survey of 3D texturing. *Computers & Graphics 25*, 1 (2001), 135–151. doi:10.1016/S0097-8493(00)00113-8. 2

[DL82] DOWSON D. C., LANDAU B. V.: The Fréchet Distance Between Multivariate Normal Distributions. *Journal of Multivariate Analysis 12*, 3 (1982), 450–455. doi:10.1016/0047-259X(82)90077-X. 21

[DLAH24] DECATUR D., LANG I., ABERMAN K., HANOCKA R.: 3D Paintbrush: Local Stylization of 3D Shapes with Cascaded Score Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.00428. 2, 12, 13, 18, 22

[DOW*24] DENG K., OMERNICK T., WEISS A., RAMANAN D., ZHU J.-Y., ZHOU T., AGRAWALA M.: FlashTex: Fast Relightable Mesh Texturing with LightControlNet. In *European Conference on Computer Vision* (2024), Lecture Notes in Computer Science, Springer. doi:10.1007/978-3-031-73383-3_6. 12, 13, 18

[DSS*23] DEITKE M., SCHWENK D., SALVADOR J., WEIHS L., MICHEL O., VANDERBILT E., SCHMIDT L., EHSANI K., KEMBHAVI A., FARHADI A.: Objaverse: A Universe of Annotated 3D Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 13142–13153. doi:10.1109/CVPR52729.2023.01263. 3, 14, 15, 19, 20, 23

[DST*21] DAI A., SIDDIQUI Y., THIES J., VALENTIN J., NIESSNER M.: SPSG: Self-Supervised Photometric Scene Generation from RGB-D Scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 1636–1645. doi:10.1109/CVPR46437.2021.00171. 10, 18

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach. In *Proceedings of SIGGRAPH* (1996), pp. 11–20. doi:10.1145/237170.237191. 10

[FCG*21] FU H., CAI B., GAO L., ZHANG L.-X., WANG J., LI C., ZENG Q., SUN C., JIA R., ZHAO B., ET AL.: 3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10913–10922. doi:10.1109/ICCV48922.2021.01075. 19

[FH05] FLOATER M. S., HORMANN K.: Surface Parameterization: a Tutorial and Survey. In *Advances in Multiresolution for Geometric Modelling*. Springer, 2005, pp. 157–186. doi:10.1007/3-540-26808-1_9. 2, 4

[FJG*21] FU H., JIA R., GAO L., GONG M., ZHAO B., MAYBANK S., TAO D.: 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision 129*, 12 (2021), 3313–3337. 19

[FLL*25] FENG J., LI X., LIN J., LIU J., LIU G., LOU W., MA S., SHI G., WANG Q., WANG J., XU Z., YI X., YU Z., ZHANG J., ZHU Y., CHEN R., CHI J., DU Z., HAN L., HUANG L., JIANG K., LI Y., LUO G., WANG S., WU Q., YANG F., ZHANG J., ZHANG X.: Seed3D 1.0: From Images to High-Fidelity Simulation-Ready 3D Assets, 2025. URL: https://arxiv.org/abs/2510.19944, arXiv:2510.19944. 17

[FYY*25] FENG Y., YANG M., YANG S., ZHANG S., YU J., ZHAO Z., LIU Y., JIANG J., GUO C.: Romantex: Decoupling 3d-aware rotary positional embedded multi-attention network for texture synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025), pp. 17203–17213. 14, 18

[FZB24] FOTI S., ZAFEIRIOU S., BIRDAL T.: UV-free Texture Generation with Denoising and Geodesic Heat Diffusion. *Advances in Neural Information Processing Systems 37* (2024), 128053–128081. URL: https://proceedings.neurips.cc/paper_files/paper/2024/hash/e70ffb3f05096e62b5077d8e1b62e668-Abstract-Conference.html, doi:10.52202/079017-4066. 15

[GA] GUY R., AGOPIAN M.: Filament Materials Guide. Online documentation. URL: https://google.github.io/filament/Materials.md.html. 5

[GAA*17] GULRAJANI I., AHMED F., ARJOVSKY M., DUMOULIN V., COURVILLE A.: Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems* (2017), vol. 30. URL: https://papers.nips.cc/paper_files/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html. 7

[GBCW20] GU S., BAO J., CHEN D., WEN F.: GIQA: Generated Image Quality Assessment. In *European Conference on Computer Vision* (2020), pp. 369–385. doi:10.1007/978-3-030-58621-8_22. 21

[GBR*12] GRETTON A., BORGWARDT K. M., RASCH M. J., SCHÖLKOPF B., SMOLA A.: A Kernel Two-Sample Test. *Journal of Machine Learning Research 13* (2012), 723–773. URL: https://www.jmlr.org/papers/v13/gretton12a.html. 21

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2016), pp. 2414–2423. doi:10.1109/CVPR.2016.265. 11, 21

[Gel90] GELBRICH M.: On a Formula for the L^2 Wasserstein Metric Between Measures on Euclidean and Hilbert Spaces. *Mathematische Nachrichten 147*, 1 (1990), 185–203. doi:10.1002/mana.19901470121. 21

[GJL*24] GAO C., JIANG B., LI X., ZHANG Y., YU Q.: Genesistex: adapting image denoising diffusion to texture space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4620–4629. doi:10.1109/CVPR52733.2024.00442. 14, 18

[GMC*08] GROSS R., MATTHEWS I., COHN J., KANADE T., BAKER S.: Multi-PIE. In *2008 8th IEEE International Conference on Automatic Face and Gesture Recognition* (2008), pp. 1–8. doi:10.1109/AFGR.2008.4813399. 20

[GNK18] GÜLER R. A., NEVEROVA N., KOKKINOS I.: DensePose: Dense Human Pose Estimation in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 7297–7306. doi:10.1109/CVPR.2018.00762. 16, 20

[GPM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (2014), pp. 2672–2680. doi:10.5555/2969033.2969125. 3, 6, 7, 10

[GSVL19] GRIGOREV A., SEVASTOPOLSKY A., VAKHITOV A., LEMPITSKY V.: Coordinate-Based Texture Inpainting for Pose-Guided Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 12126–12135. doi:10.1109/CVPR.2019.01241. 16, 17, 18

[GZD*23] GUO Y., ZUO X., DAI P., LU J., WU X., CHENG L., YAN Y., XU S., WU X.: Decorate3D: Text-Driven High-Quality Texture Generation for Mesh Decoration in the Wild. In *Advances in Neural Information Processing Systems* (2023), vol. 36, pp. 36664–36676. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/73af055566f5514b9863315133b84eda-Abstract-Conference.html. 13, 18

[HDAA*24] HU A., DESAI N., ABU ALHAIJA H., KIM S. W., SHUGRINA M.: Diffusion Texture Painting. In *ACM SIGGRAPH 2024 Conference Papers* (2024). doi:10.1145/3641519.3657458. 18

[HDS*25] HUANG K., DUAN C., SUN K., XIE E., LI Z., LIU X.: T2I-CompBench++: An Enhanced and Comprehensive Benchmark for Compositional Text-to-Image Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 47*, 5 (2025), 3563–3579. doi:10.1109/TPAMI.2025.3531907. 21

[Hec86] HECKBERT P. S.: Survey of texture mapping. *IEEE Computer Graphics and Applications 6*, 11 (1986), 56–67. doi:10.1109/MCG.1986.276672. 2

[HGZ*24] HUO D., GUO Z., ZUO X., SHI Z., LU J., DAI P., XU S., CHENG L., YANG Y.-H.: Texgen: Text-guided 3d texture generation with multi-view sampling and resampling. In *European Conference on Computer Vision* (2024), Springer, pp. 352–368. doi:10.1007/978-3-031-72920-1_20. 15, 18, 20

[HHF*21] HESSEL J., HOLTZMAN A., FORBES M., LE BRAS R., CHOI Y.: CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2021), pp. 7514–7528. doi:10.18653/v1/2021.emnlp-main.595. 21

[HJA20] HO J., JAIN A., ABBEEL P.: Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems* (2020), pp. 6840–6851. doi:10.48550/arXiv.2006.11239. 3, 7

[HJN22] HÖLLEIN L., JOHNSON J., NIESSNER M.: StyleMesh: Style Transfer for Indoor 3D Scene Reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 6198–6208. doi:10.1109/CVPR52688.2022.00610. 22

[HPS08] HORMANN K., POLTHIER K., SHEFFER A.: Mesh Parameterization: Theory and Practice. In *ACM SIGGRAPH ASIA 2008 Courses* (New York, NY, USA, 2008), SIGGRAPH Asia '08, Association for Computing Machinery. URL: https://doi.org/10.1145/1508044.1508091, doi:10.1145/1508044.1508091. 2, 4

[HRU*17] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., KLAMBAUER G., HOCHREITER S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems* (2017), pp. 6626–6637. URL: https://papers.nips.cc/paper_files/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html. 21

[HSG*22] HO J., SALIMANS T., GRITSENKO A., CHAN W., NOROUZI M., FLEET D. J.: Video Diffusion Models. In *Advances in Neural Information Processing Systems* (2022). doi:10.5555/3600270.3600898. 7

[HSH24] HO H.-I., SONG J., HILLIGES O.: SiTH: Single-View Textured Human Reconstruction With Image-Conditioned Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.00058. 17

[HSX*23] HUANG K., SUN K., XIE E., LI Z., LIU X.: T2I-CompBench: A Comprehensive Benchmark for Open-world Compositional Text-to-image Generation. *Advances in Neural Information Processing Systems 36* (2023), 78723–78747. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/f8ad010cdd9143dbb0e9308c093aff24-Abstract-Datasets_and_Benchmarks.html. 21

[HTD*20] HUANG J., THIES J., DAI A., KUNDU A., JIANG C. M., GUIBAS L., NIESSNER M., FUNKHOUSER T.: Adversarial Texture Optimization from RGB-D Scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 1550–1559. doi:10.1109/CVPR42600.2020.00162. 10, 18

[HTL20] HENDERSON P., TSIMINAKI V., LAMPERT C. H.: Leveraging 2D Data to Learn Textured 3D Mesh Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 7498–7507. doi:10.1109/CVPR42600.2020.00753. 10

[HWLW25] HUANG X., WANG T., LIU Z., WANG Q.: Material anything: Generating materials for any 3d object via diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025), pp. 26556–26565. doi:10.1109/CVPR52734.2025.02473. 2, 15, 18, 20

[HYC*25] HUANG Z., YU W., CHENG X., ZHAO C., GE Y., GUO M., YUAN L., TIAN Y.: Roompainter: View-integrated diffusion for consistent indoor scene texturing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025), pp. 574–584. doi:10.1109/CVPR52734.2025.00062. 18, 22

[Hyp] HYPER3D: Rodin / Hyper3D API Documentation. Online documentation and API. Non–peer-reviewed documentation. URL: https://hyper3d.ai/. 17

[HYY*25a] HE Z., YANG M., YANG S., TANG Y., WANG T., ZHANG K., CHEN G., LIU Y., JIANG J., GUO C., ET AL.: MaterialMVP: Illumination-Invariant Material Generation via Multi-view PBR Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025), pp. 26294–26305. 14, 18

[HYY*25b] HUNYUAN3D T., YANG S., YANG M., FENG Y., HUANG X., ZHANG S., HE Z., LUO D., LIU H., ZHAO Y., ET AL.: Hunyuan3D 2.1: From Images to High-Fidelity 3D Assets with Production-Ready PBR Material. *arXiv preprint arXiv:2506.15442* (2025). URL: https://github.com/Tencent-Hunyuan/Hunyuan3D-2.1, doi:10.48550/arXiv.2506.15442. 17

[HZY*19] HUANG J., ZHANG H., YI L., FUNKHOUSER T., NIESSNER M., GUIBAS L. J.: TextureNet: Consistent Local Parametrizations for Learning From High-Resolution Signals on Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4435–4444. doi:10.1109/CVPR.2019.00457. 10

[ITU19] ITU-R: *Methodologies for the Subjective Assessment of the Quality of Television Images*. Tech. Rep. Recommendation ITU-R

BT.500-14, International Telecommunication Union, 2019. Accessed 2025-10-09. 22

[IZZE17] ISOLA P., ZHU J., ZHOU T., EFROS A. A.: Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), pp. 5967–5976. doi:10.1109/CVPR.2017.632. 7

[Jet21] JETCHEV N.: ClipMatrix: Text-controlled Creation of 3D Textured Meshes. *arXiv preprint arXiv:2109.12922* (2021). URL: https://arxiv.org/abs/2109.12922, doi:10.48550/arXiv.2109.12922. 11

[JGJS99] JORDAN M. I., GHAHRAMANI Z., JAAKKOLA T. S., SAUL L. K.: An Introduction to Variational Methods for Graphical Models. *Machine Learning 37*, 2 (1999), 183–233. doi:10.1023/A:1007665907178. 6

[JMB*22] JAIN A., MILDENHALL B., BARRON J. T., ABBEEL P., POOLE B.: Zero-Shot Text-Guided Object Generation with Dream Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 867–876. doi:10.1109/CVPR52688.2022.00094. 11, 18

[JSL*19] JATAVALLABHULA K. M., SMITH E., LAFLECHE J.-F., TSANG C. F., ROZANTSEV A., CHEN W., XIANG T., LEBAREDIAN R., FIDLER S.: Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research. arXiv:1911.05063, 2019. doi:10.48550/arXiv.1911.05063. 2, 4

[JSRV22] JAKOB W., SPEIERER S., ROUSSEL N., VICINI D.: Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering. *ACM Transactions on Graphics 41*, 4 (2022), 124:1–124:19. doi:10.1145/3528223.3530099. 4, 13

[JYZ*25] JIANG D., YANG X., ZHAO Z., ZHANG S., YU J., LAI Z., YANG S., GUO C., ZHOU X., KE Z.: FlexiTex: Enhancing Texture Generation via Visual Guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 3967–3975. doi:10.1609/aaai.v39i4.32415. 14, 18

[Kaj86] KAJIYA J. T.: The Rendering Equation. *ACM SIGGRAPH Computer Graphics 20*, 4 (1986), 143–150. doi:10.1145/15886.15902. 4, 5

[Khra] KHRONOS GROUP: glTF 2.0 Specification (Materials / Metallic-Roughness Model). Online specification. URL: https://kcoley.github.io/glTF/specification/2.0/. 5

[Khrb] KHRONOS GROUP: PBR (Physically Based Rendering) in glTF. Online documentation. URL: https://www.khronos.org/gltf/pbr. 5

[KKW*23] KHACHATRYAN L., KNYAZEV B., WANG R., ET AL.: Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 15954–15964. doi:10.1109/ICCV51070.2023.01473. 23

[KL51] KULLBACK S., LEIBLER R. A.: On Information and Sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86. doi:10.1214/aoms/1177729694. 6

[KLA19] KARRAS T., LAINE S., AILA T.: A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410. doi:10.1109/CVPR.2019.00453. 8, 20

[KLA*20] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8107–8116. doi:10.1109/CVPR42600.2020.00813. 8, 10

[KMJ*19] KOCH S., MATVEEV A., JIANG Z., WILLIAMS F., ARTEMOV A., BURNAEV E., ALEXA M., ZORIN D., PANOZZO D.: ABC: A Big CAD Model Dataset for Geometric Deep Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 9601–9611. doi:10.1109/CVPR.2019.00983. 3

[KSJ*16] KINGMA D. P., SALIMANS T., JOZEFOWICZ R., CHEN X., SUTSKEVER I., WELLING M.: Improved Variational Inference with Inverse Autoregressive Flow. arXiv:1606.04934, 2016. doi:10.48550/arXiv.1606.04934. 6

[KTEM18] KANAZAWA A., TULSIANI S., EFROS A. A., MALIK J.: Learning Category-Specific Mesh Reconstruction from Image Collections. In *European Conference on Computer Vision* (2018), pp. 371–389. doi:10.1007/978-3-030-01267-0_23. 9, 10, 11

[KUH18] KATO H., USHIKU Y., HARADA T.: Neural 3D Mesh Renderer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 3907–3916. doi:10.1109/CVPR.2018.00411. 4, 11, 18, 21

[KW14] KINGMA D. P., WELLING M.: Auto-Encoding Variational Bayes. arXiv:1312.6114, 2014. doi:10.48550/arXiv.1312.6114. 6

[KW19] KINGMA D. P., WELLING M.: An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning 12*, 4 (2019), 307–392. doi:10.1561/2200000056. 3

[KXBP22] KHALID N. M., XIE T., BELILOVSKY E., POPA T.: CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 1–8. doi:10.1145/3550469.3555392. 6, 11, 18

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Transactions on Graphics 37*, 6 (2018), 222:1–222:11. doi:10.1145/3272127.3275109. 4

[LAKH23] LIU R., AIGERMAN N., KIM V. G., HANOCKA R.: DA Wand: Distortion-Aware Selection Using Neural Mesh Parameterization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 20575–20584. doi:10.1109/CVPR52729.2023.01606. 4

[LB14] LOPER M., BLACK M. J.: OpenDR: An Approximate Differentiable Renderer. In *European Conference on Computer Vision* (2014), Springer, pp. 154–169. doi:10.1007/978-3-319-10584-0_11. 4

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 163–169. URL: https://doi.org/10.1145/37402.37422, doi:10.1145/37402.37422. 3

[LCC16] LIU K.-H., CHEN T.-Y., CHEN C.-S.: Mvc: A dataset for view-invariant clothing retrieval and attribute prediction. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (2016), pp. 313–316. 20

[LCD*23] LUO Z., CAI S., DONG J., MING R., QIU L., ZHAN X., HAN X.: RaBit: Parametric Modeling of 3D Biped Cartoon Characters with a Topological-Consistent Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12825–12835. doi:10.1109/CVPR52729.2023.01233. 19

[LCL*25] LI Y., CHEUNG V., LIU X., CHEN Y., LUO Z., LEI B., WENG H., ZHAO Z., HUANG J., CHEN Z., GUO C.: Auto-Regressive Surface Cutting. arXiv preprint, 2025. URL: https://arxiv.org/abs/2506.18017. 4

[LCY*23] LIU H., CHEN Z., YUAN Y., MEI X., LIU X., MANDIC D., WANG W., PLUMBLEY M. D.: AudioLDM: Text-to-Audio Generation with Latent Diffusion Models. In *International Conference on Machine Learning* (2023), vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 21450–21474. URL: https://proceedings.mlr.press/v202/liu23f.html. 7

[LFW*24] LI K., FAN Y., WU Y., SUN Z., YANG W., JI X., YUAN L., CHEN J.: Learning pseudo 3d guidance for view-consistent texturing with 2d diffusion. In *European Conference on Computer Vision* (2024), Springer, pp. 18–34. doi:10.1007/978-3-031-73016-0_2. 21

[LGT*23] LIN C., GAO J., TANG L., TAKIKAWA T., ZENG X., HUANG

X., KREIS K., FIDLER S., LIU M., LIN T.: Magic3D: High-Resolution Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023). doi:10.1109/CVPR52729.2023.00037. 7, 13

[LHJ19] LOUBET G., HOLZSCHUCH N., JAKOB W.: Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Transactions on Graphics 38*, 6 (2019), 228:1–228:14. doi:10.1145/3355089.3356510. 4

[LIPM19] LAZOVA V., INSAFUTDINOV E., PONS-MOLL G.: 360-Degree Textures of People in Clothing From a Single Image. In *International Conference on 3D Vision* (2019), pp. 643–653. doi:10.1109/3DV.2019.00076. 15, 16, 17, 18

[LKA*20] LAINE S., KARRAS T., AITTALA M., HELLSTEN A., LEHTINEN J.: Modular Differentiable Rendering with a Monolithic Architecture. *ACM Transactions on Graphics 39*, 4 (2020), 140:1–140:14. doi:10.1145/3386569.3392409. 4, 13

[LKM*18] LUCIC M., KURACH K., MICHALSKI M., GELLY S., BOUSQUET O.: Are GANs Created Equal? A Large-Scale Study. In *Advances in Neural Information Processing Systems* (2018), vol. 31. URL: https://papers.nips.cc/paper_files/paper/2018/hash/e46de7e1bcaaced9a54f1e9d0d2f800d-Abstract.html. 11

[LL22] LOW W. F., LEE G. H.: Minimal Neural Atlas: Parameterizing Complex Surfaces with Minimal Charts and Distortion. In *European Conference on Computer Vision* (2022), pp. 475–492. doi:10.1007/978-3-031-20086-1_27. 4

[LLC*18] LÉVY B., LI W., CHEN Y., PANOZZO D., BOMMES D., JAKOB W., SORKINE-HORNUNG O., SHEFFER A.: OptCuts: Joint Optimization of Surface Cuts and Parameterization. *ACM Transactions on Graphics 37*, 6 (2018), 259:1–259:15. doi:10.1145/3272127.3275042. 4

[LLC*25] LIANG Y., LUO K., CHEN X., CHEN R., YAN H., LI W., LIU J., TAN P.: UniTEX: Universal High Fidelity Generative Texturing for 3D Shapes. *arXiv preprint arXiv:2505.23253* (2025). URL: https://arxiv.org/abs/2505.23253, doi:10.48550/arXiv.2505.23253. 15

[LLCL19] LIU S., LI W., CHEN W., LI X.: Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7708–7717. doi:10.1109/ICCV.2019.00780. 4

[LLQ*16] LIU Z., LUO P., QIU S., WANG X., TANG X.: DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2016). 20

[LLXH22] LI J., LI D., XIONG C., HOI S. C. H.: BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning* (2022), vol. 162 of *Proceedings of Machine Learning Research*, PMLR, pp. 12888–12900. URL: https://proceedings.mlr.press/v162/li22n.html. 3, 6

[LLZ*24] LU L., LI R., ZHANG X., WEI H., DU G., WANG B.: Advances in text-guided 3D editing: A survey. *Artificial Intelligence Review* (2024). doi:10.1007/s10462-024-10937-6. 2

[LMB*14] LIN T.-Y., MAIRE M., BELONGIE S., HAYS J., PERONA P., RAMANAN D., DOLLÁR P., ZITNICK C. L.: Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision* (2014), Springer, pp. 740–755. doi:10.1007/978-3-319-10602-1_48. 20

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics 34*, 6 (2015), 248:1–248:16. doi:10.1145/2816795.2818013. 8, 15

[Lob09] LOBSTERBAKE: Mesh overview.svg. Wikimedia Commons, June 2009. Derivative of an image originally uploaded by Rchoetzlein; licensed under CC BY–SA 3.0. URL: https://commons.wikimedia.org/wiki/File:Mesh_overview.svg. 3

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions on Graphics 21*, 3 (2002), 362–371. doi:10.1145/566654.566590. 2, 4

[LTH*23] LUO S., TAN Y., HUANG L., LI J., ZHAO H.: Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference. *arXiv preprint arXiv:2310.04378* (2023). doi:10.48550/arXiv.2310.04378. 23

[LTT*19] LI Y., TSIMINAKI V., TIMOFTE R., POLLEFEYS M., VAN GOOL L.: 3D Appearance Super-Resolution with Deep Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 9671–9680. doi:10.1109/CVPR.2019.00989. 10

[LWG*25] LIU J., WU J., GAO X., HU J., XIONG B., LIU X., ZHAO C., PEI H., FENG H., LI Y., DING E., WANG J.: TexGarment: Consistent Garment UV Texture Generation via Efficient 3D Structure-Guided Diffusion Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025). doi:10.1109/CVPR52734.2025.02474. 17, 18

[LXLW24] LIU Y., XIE M., LIU H., WONG T.-T.: Text-Guided Texturing by Synchronized Multi-View Diffusion. In *SIGGRAPH Asia 2024 Conference Papers* (2024). doi:10.1145/3680528.3687621. 13, 14, 15, 18, 21, 22, 23

[LYC*24] LIU S., YU C., CAO C., QIAN W., WANG F.: VCD-Texture: Variance alignment based 3D-2D co-denoising for text-guided texturing. In *European Conference on Computer Vision* (2024), Springer, pp. 373–389. doi:10.1007/978-3-031-72640-8_21. 14, 18, 21

[LZC*23] LI C., ZHANG C., CHO J., WAGHWASE A., LEE L.-H., RAMEAU F., YANG Y., BAE S.-H., HONG C.-S.: Generative AI meets 3D: A Survey on Text-to-3D in AIGC Era. arXiv:2305.06131, 2023. doi:10.48550/arXiv.2305.06131. 2

[LZS*25] LI W., ZHANG X., SUN Z., QI D., LI H., CHENG W., CAI W., WU S., LIU J., WANG Z., ET AL.: Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747* (2025). URL: https://arxiv.org/abs/2505.07747, doi:10.48550/arXiv.2505.07747. 17, 20

[LZT*24] LIU Y., ZHU J., TANG J., ZHANG S., ZHANG J., CAO W., WANG C., WU Y., HUANG D.: TexDreamer: Towards Zero-Shot High-Fidelity 3D Human Texture Generation. In *European Conference on Computer Vision* (2024), Spencer P. et al., (Eds.), Lecture Notes in Computer Science, Springer. doi:10.1007/978-3-031-72970-6_11. 2, 8, 15, 16, 17, 18, 19

[LZZ*22] LI L. H., ZHANG P., ZHANG H., YANG J., LI C., ZHONG Y., WANG L., YUAN L., ZHANG L., HWANG J., CHANG K., GAO J.: Grounded Language-Image Pre-Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10965–10975. doi:10.1109/CVPR52688.2022.01069. 6

[LZZ*25] LU J., ZHANG Y., ZHAO Z., WANG H., ZHOU K., SHAO T.: Genesistex2: Stable, consistent and high-quality text-to-texture generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 5820–5828. doi:10.1609/aaai.v39i6.32621. 14, 18

[MAPM20] MIR A., ALLDIECK T., PONS-MOLL G.: Learning to Transfer Texture From Clothing Images to 3D Humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 7021–7032. doi:10.1109/CVPR42600.2020.00705. 16, 17, 19

[MBOL*22] MICHEL O., BAR-ON R., LIU R., BENAIM S., HANOCKA R.: Text2Mesh: Text-Driven Neural Stylization for Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 13492–13502. doi:10.1109/CVPR52688.2022.01313. 2, 6, 11, 18, 22

[MEM24] MITCHEL T. W., ESTEVES C., MAKADIA A.: Single Mesh

Diffusion Models with Field Latents for Texture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 7953–7963. doi:10.1109/CVPR52733.2024.00760. 1, 9, 15, 18

[Mes] MESHY: Meshy API Documentation. Online documentation and API. Non–peer-reviewed documentation. URL: https://docs.meshy.ai/. 17

[MMB12] MITTAL A., MOORTHY A. K., BOVIK A. C.: No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Trans. Image Processing 21*, 12 (2012), 4695–4708. doi:10.1109/TIP.2012.2214050. 21

[MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4460–4470. doi:10.1109/CVPR.2019.00459. 3, 5

[MRP*23] METZER G., RICHARDSON E., PATASHNIK O., GIRYES R., COHEN-OR D.: Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023). doi:10.1109/CVPR52729.2023.01218. 12, 13

[MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision* (2020), pp. 405–421. doi:10.1007/978-3-030-58452-8_24. 3, 6

[MWL*25] MUNKBERG J., WANG Z., LIANG R., SHEN T., HASSELGREN J.: VideoMat: Extracting PBR Materials from Video Diffusion Models. In *Computer Graphics Forum* (2025), vol. 44, Wiley Online Library, p. e70180. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.70180, doi:10.1111/cgf.70180. 12, 18

[MXZ*23] MA X., XU X., ZHANG L., ZHOU K., WU H.: OpenSVBRDF: A Database of Measured Spatially-Varying Reflectance. *ACM Transactions on Graphics 42*, 6 (2023), 1–14. doi:10.1145/3618358. 23

[MZS*23] MA Y., ZHANG X., SUN X., JI J., WANG H., JIANG G., ZHUANG W., JI R.: X-Mesh: Towards Fast and Accurate Text-driven 3D Stylization via Dynamic Textual Guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 2749–2760. doi:10.1109/ICCV51070.2023.00258. 2, 11, 18

[NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Transactions on Graphics 38*, 6 (2019), 203:1–203:17. doi:10.1145/3355089.3356498. 4, 13

[NGK19] NEVEROVA N., GULER R. A., KOKKINOS I.: Dense Pose Transfer. In *European Conference on Computer Vision Workshops* (2019), Leal-Taixé L., Roth S., (Eds.), vol. 11131 of *Lecture Notes in Computer Science*, Springer, pp. 123–137. doi:10.1007/978-3-030-01219-9_8. 16, 18

[NH98] NEAL R. M., HINTON G. E.: A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In *Learning in Graphical Models*, Jordan M. I., (Ed.). Springer, 1998, pp. 355–368. doi:10.1007/978-94-011-5014-9_12. 6

[NKML25] NAM H., KIM D., MOON G., LEE K. M.: PARTE: Part-Guided Texturing for 3D Human Reconstruction From a Single Image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025). 2, 16, 17, 22

[NRH*77] NICODEMUS F. E., RICHMOND J. C., HSIA J. J., GINSBERG I. W., LIMPERIS T.: *Geometrical Considerations and Nomenclature for Reflectance*. Tech. Rep. NBS Monograph 160, National Bureau of Standards, 1977. doi:10.6028/NBS.MONO.160. 5

[OLY*17] OLSZEWSKI K., LI Z., YANG C., ZHOU Y., YU R., HUANG Z.: Realistic Dynamic Facial Textures From a Single Image Using GANs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2017), pp. 5429–5438. doi:10.1109/ICCV.2017.580. 15, 17

[OMN*19] OECHSLE M., MESCHEDER L., NIEMEYER M., STRAUSS T., GEIGER A.: Texture Fields: Learning Texture Representations in Function Space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 4530–4539. doi:10.1109/ICCV.2019.00463. 10, 18

[PAL*21] PARK D. H., AZADI S., LIU X., DARRELL T., ROHRBACH A.: Benchmark for Compositional Text-to-Image Synthesis. In *Advances in Neural Information Processing Systems* (2021). 21

[PCG*19] PAVLAKOS G., CHOUTAS V., GHORBANI N., BOLKART T., OSMAN A. A. A., TZIONAS D., BLACK M. J.: Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10975–10985. doi:10.1109/CVPR.2019.01123. 8, 15

[PCN*20] PERALTA D., CASIMIRO J., NILLES A. M., AGUILAR J. A., ATIENZA R., CAJOTE R.: Next-Best View Policy for 3D Reconstruction. *arXiv preprint arXiv:2008.12664* (2020). URL: https://arxiv.org/abs/2008.12664, doi:10.48550/arXiv.2008.12664. 19

[PCOS10] PIETRONI N., CIGNONI P., OTADUY M. A., SCOPIGNO R.: Solid-texture synthesis: A survey. *IEEE Computer Graphics and Applications 30*, 4 (2010), 74–89. doi:10.1109/MCG.2009.153. 2

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 165–174. doi:10.1109/CVPR.2019.00025. 3, 5

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson Image Editing. In *ACM SIGGRAPH 2003 Conference Papers* (2003), pp. 313–318. doi:10.1145/1201775.882269. 10

[PJBM23] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: DreamFusion: Text-to-3D using 2D Diffusion. In *International Conference on Learning Representations* (2023). URL: https://openreview.net/forum?id=FjNys5c7VyY. 2, 11, 12, 13, 15

[PJH23] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*, 4 ed. MIT Press, 2023. URL: https://pbr-book.org/4ed/. 4, 5

[PKD19] PAVLAKOS G., KOLOTOUROS N., DANIILIDIS K.: TexturePose: Supervising Human Mesh Estimation With Texture Consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 803–812. doi:10.1109/ICCV.2019.00089. 16, 17

[PRFS18] PARK K., REMATAS K., FARHADI A., SEITZ S. M.: PhotoShape: Photorealistic Materials for Large-Scale Shape Collections. In *SIGGRAPH Asia Conference Papers* (2018). doi:10.1145/3272127.3275066. 9, 19

[PVN*25] PERLA S. R. K., VORA A., NAG S., MAHDAVI-AMIRI A., ZHANG H.: ASIA: Adaptive 3d segmentation using few image annotations. *SIGGRAPH Asia Conference Papers* (2025). URL: https://github.com/sairajk/asia, doi:10.1145/3757377.3763821. 23

[PWMAZ24] PERLA S. R. K., WANG Y., MAHDAVI-AMIRI A., ZHANG H.: EASI-Tex: Edge-Aware Mesh Texturing from Single Image. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 43*, 4 (2024), 40:1–40:16. doi:10.1145/3658222. 2, 8, 13, 18, 20, 22

[PZVBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: Surface Elements as Rendering Primitives. In *Proceedings of SIGGRAPH* (2000), pp. 335–342. doi:10.1145/344779.344936. 3

[R*21] RADFORD A., ET AL.: Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020* (2021). doi:10.48550/arXiv.2103.00020. 2, 3, 6, 9, 11, 21

[RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10684–10695. doi:10.1109/CVPR52688.2022.01042. 2, 6, 7, 9, 11

[RCO*19] RICHARD A., CHERABIER I., OSWALD M. R., TSIMINAKI V., POLLEFEYS M., SCHINDLER K.: Learned Multi-View Texture Super-Resolution. In *International Conference on 3D Vision* (2019), pp. 570–579. doi:10.1109/3DV.2019.00068. 10, 20

[RDDM18] RAAD L., DAVY A., DESOLNEUX A., MOREL J.-M.: A survey of exemplar-based texture synthesis. *Annals of Mathematical Sciences and Applications 3*, 1 (2018), 89–148. doi:10.4310/AMSA.2018.v3.n1.a4. 2

[Ren25] RENDERPEOPLE GMBH: Renderpeople: Scanned 3D People Models. renderpeople.com, 2025. Commercial library of scanned 3D humans; accessed 2025-10-09. URL: https://renderpeople.com/. 19

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A Multiresolution Point Rendering System for Large Meshes. In *Proceedings of SIGGRAPH* (2000), pp. 343–352. doi:10.1145/344779.344940. 3

[RLJ*23] RUIZ N., LI Y., JAMPANI V., PRITCH Y., RUBINSTEIN M., ABERMAN K.: DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 22500–22510. doi:10.1109/CVPR52729.2023.02155. 12

[RM15] REZENDE D., MOHAMED S.: Variational Inference with Normalizing Flows. arXiv:1505.05770, 2015. doi:10.48550/arXiv.1505.05770. 6

[RMA*23] RICHARDSON E., METZER G., ALALUF Y., GIRYES R., COHEN-OR D.: TEXTure: Text-Guided Texturing of 3D Shapes. In *ACM SIGGRAPH 2023 Conference Papers* (2023), pp. 54:1–54:11. doi:10.1145/3588432.3591503. 2, 8, 11, 13, 14, 15, 18, 21, 22, 23

[RPCGLM24] RODRÍGUEZ-PARDO C., CASAS D., GARCÉS E., LÓPEZ-MORENO J.: TexTile: A Differentiable Metric for Texture Tileability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4439–4449. doi:10.1109/CVPR52733.2024.00425. 21

[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. doi:10.48550/arXiv.2007.08501. 2, 4, 13

[RSR*20] RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W., LIU P. J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research 21*, 140 (2020), 1–67. URL: https://jmlr.org/papers/v21/20-074.html. 9

[SC17] SAWHNEY R., CRANE K.: Boundary First Flattening. *ACM Transactions on Graphics 37*, 1 (2017), 5:1–5:14. doi:10.1145/3132705. 4

[Sch94] SCHLICK C.: An Inexpensive BRDF Model for Physically-based Rendering. *Computer Graphics Forum 13*, 3 (1994), 233–246. doi:10.1111/1467-8659.1330233. 5

[Sch22] SCHUHMANN C.: Improved Aesthetic Predictor: CLIP+MLP Aesthetic Score Predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor, 2022. GitHub repository, accessed 2026-03-30. 21

[SGZ*16] SALIMANS T., GOODFELLOW I., ZAREMBA W., CHEUNG V., RADFORD A., CHEN X.: Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems* (2016), vol. 29. URL: https://papers.nips.cc/paper_files/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html. 21

[SJŠ*19] SOCHOR J., JURÁNEK R., ŠPAŇHEL J., MARŠÍK L., ŠIROKÝ A., HEROUT A., ZEMČÍK P.: Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement. *IEEE Transactions on Intelligent Transportation Systems 20*, 5 (2019), 1633–1643. doi:10.1109/TITS.2018.2825609. 20

[SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: ABF++: Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics 24*, 2 (2005), 311–330. doi:10.1145/1073204.1073228. 2, 4

[SPR06] SHEFFER A., PRAUN E., ROSE K.: Mesh Parameterization Methods and Their Applications. *Foundations and Trends in Computer Graphics and Vision 2*, 2 (2006), 105–171. doi:10.1561/0600000011. 2, 3, 4

[SSGH01] SANDER P. V., SNYDER J., GORTLER S. J., HOPPE H.: Texture Mapping Progressive Meshes. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 409–416. doi:10.1145/383259.383307. 4

[STM*22] SIDDIQUI Y., THIES J., MA F., SHAN Q., NIESSNER M., DAI A.: Texturify: Generating Textures on 3D Shape Surfaces. In *European Conference on Computer Vision* (2022), Springer, Cham, pp. 72–88. doi:10.1007/978-3-031-20062-5_5. 2, 8, 10, 11, 18, 20

[SVI*16] SZEGEDY C., VANHOUCKE V., IOFFE S., SHLENS J., WOJNA Z.: Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2016), pp. 2818–2826. doi:10.1109/CVPR.2016.308. 21

[SWMG15] SOHL-DICKSTEIN J., WEISS E. A., MAHESWARANATHAN N., GANGULI S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *International Conference on Machine Learning* (2015), vol. 37 of *Proceedings of Machine Learning Research*, PMLR, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html. 7

[SXSX25] SHAO M., XIONG F., SUN Z., XU M.: MVPainter: Accurate and Detailed 3D Texture Generation via Multi-View Diffusion with Geometric Control. *arXiv preprint arXiv:2505.12635* (2025). URL: https://arxiv.org/abs/2505.12635, doi:10.48550/arXiv.2505.12635. 13, 20

[SZ15] SIMONYAN K., ZISSERMAN A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations* (2015). URL: https://arxiv.org/abs/1409.1556. 9, 21

[TFR*17] TOBIN J., FONG R., RAY A., SCHNEIDER J., ZAREMBA W., ABBEEL P.: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017), pp. 23–30. doi:10.1109/IROS.2017.8202133. 22

[Tre25] TREEDY'S SA: Treedy's: 3D Body Scanning and Digital Avatars. treedys.com, 2025. Full-body scanning hardware/software for avatars and sizing; accessed 2025-10-09. URL: https://treedys.com/. 19

[Tri] TRIPO: Tripo API Documentation. Online documentation and API. Non–peer-reviewed documentation. URL: https://studio.tripo3d.ai/. 17

[Tri25] TRIPLEGANGERS LLC: Triplegangers: Human 3D Scan Library. triplegangers.com, 2025. Large-scale scans of faces, hands, and full bodies; accessed 2025-10-09. URL: https://triplegangers.com/. 19

[Tur01] TURK G.: Texture synthesis on surfaces. In *Proceedings of SIGGRAPH 2001* (2001), ACM, pp. 347–354. doi:10.1145/383259.383297. 2

[Twi25] TWINDOM: Twindom: Full-Body 3D Scanners and Avatars. web.twindom.com, 2025. Turnkey scanners and avatar products; accessed 2025-10-09. URL: https://web.twindom.com/. 19

[TZF*24] TANG J., ZENG Y., FAN K., WANG X., DAI B., CHEN K., MA L.: Make-It-Vivid: Dressing Your Animatable Biped Cartoon

Characters From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 6243–6253. doi:10.1109/CVPR52733.2024.00597. 2, 17, 18

[TZN19] THIES J., ZOLLHÖFER M., NIESSNER M.: Deferred Neural Rendering: Image Synthesis Using Neural Textures. *ACM Transactions on Graphics 38*, 4 (2019), 66:1–66:12. doi:10.1145/3306346.3323035. 4, 22

[VD24] VECCHIO G., DESCHAINTRE V.: MatSynth: A Modern PBR Materials Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 21766–21777. doi:10.1109/CVPR52733.2024.02087. 23

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Dec. 1997. URL: https://graphics.stanford.edu/papers/veach_thesis/. 5

[Vil09] VILLANI C.: *Optimal Transport: Old and New*, vol. 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-71050-9. 21

[WAvK*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *ACM Trans. Graph. 31*, 6 (Nov. 2012). doi:10.1145/2366145.2366184. 19

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861. 21

[WBW*11] WAH C., BRANSON S., WELINDER P., PERONA P., BELONGIE S.: The Caltech-UCSD Birds-200-2011 Dataset. 20

[WD97] WEINHAUS F. M., DEVARAJAN V.: Texture mapping 3D models of real-world scenes. *ACM Computing Surveys 29*, 4 (1997), 325–365. doi:10.1145/267580.267583. 2

[Wen18] WENG L.: From Autoencoder to Beta-VAE. *lilianweng.github.io* (2018). URL: https://lilianweng.github.io/posts/2018-08-12-vae/. 6

[Wil83] WILLIAMS L.: Pyramidal Parametrics. In *Proceedings of ACM SIGGRAPH 1983* (1983), pp. 1–11. doi:10.1145/800059.801126. 3, 4

[WLKT09] WEI L.-Y., LEFEBVRE S., KWATRA V., TURK G.: State of the Art in Example-based Texture Synthesis. In *Eurographics 2009 – State of the Art Reports (STAR)* (2009), Eurographics Association, pp. 93–117. doi:10.2312/egst.20091063. 2

[WLW*23] WANG Z., LU C., WANG Y., BAO F., LI C., SU H., ZHU J.: ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Advances in Neural Information Processing Systems* (2023), vol. 36, pp. 8406–8441. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/1a87980b9853e84dfb295855b425c262-Abstract-Conference.html. 2, 11, 12, 15

[WLX*24] WANG Q., LU R., XU X., WANG J., WANG M. Y., DAI B., ZENG G., XU D.: Roomtex: Texturing compositional indoor scenes via iterative inpainting. In *European Conference on Computer Vision* (2024), Springer, pp. 465–482. doi:10.1007/978-3-031-73113-6_27. 1, 13, 18, 22

[WMG14] WÄECHTER M., MOEHRLE N., GOESELE M.: Let There Be Color! Large-Scale Texturing of 3D Reconstructions. In *European Conference on Computer Vision* (2014), vol. 8693 of *LNCS*, Springer, pp. 836–850. doi:10.1007/978-3-319-10605-2_54. 10

[wol16] WOLF: PBR Materials Addon (BlenderArtists forum post #42). BlenderArtists Forum, July 2016. Posted July 2016. Accessed 2026-02-01. URL: https://blenderartists.org/t/pbr-materials-addon/671208/42. 5

[Wol17] WOLFE J.: Scanline VFX Licenses Ziva Technology for Virtual Character Simulation. Animation World Network (AWN) News, June 2017. Published June 20, 2017. Accessed January 25, 2026. URL: https://www.awn.com/news/scanline-vfx-licenses-ziva-technology-virtual-character-simulation. 22

[WSB03] WANG Z., SIMONCELLI E. P., BOVIK A. C.: Multiscale structural similarity for image quality assessment. In *The thrity-seventh Asilomar Conference on Signals, Systems & Computers* (2003), vol. 2, IEEE, pp. 1398–1402. 21

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2015), pp. 1912–1920. doi:10.1109/CVPR.2015.7298801. 19

[WWS*25] WANG Z., WEI X., SHI R., ZHANG X., SU H., LIU M.: PartUV: Part-Based UV Unwrapping of 3D Meshes. In *SIGGRAPH Asia 2025 Conference Papers* (New York, NY, USA, 2025), SA Conference Papers '25, Association for Computing Machinery. doi:10.1145/3757377.3763843. 3, 4

[WZL*19] WANG J., ZHONG Y., LI Y., ZHANG C., WEI Y.: Re-Identification Supervised Texture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 6040–6049. doi:10.1109/CVPR.2019.01212. 15, 16, 17

[XGF*25] XIANG X., GORELIK L. S., FAN Y., ARMSTRONG O., IANDOLA F., LI Y., LIFSHITZ I., RANJAN R.: Make-A-Texture: Fast Shape-Aware Texture Generation in 3 Seconds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2025), IEEE, pp. 4872–4881. doi:10.1109/WACV61041.2025.00477. 13, 18

[XLH*25] XIONG B., LIU J., HU J., WU C., WU J., LIU X., ZHAO C., DING E., LIAN Z.: TexGaussian: Generating High-quality PBR Material via Octree-based 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025). doi:10.1109/CVPR52734.2025.00060. 1, 10

[XMS14] XIANG Y., MOTTAGHI R., SAVARESE S.: Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2014), pp. 75–82. doi:10.1109/WACV.2014.6836101. 19

[XTS*22] XIE Y., TAKIKAWA T., SAITO S., LITANY O., YAN S., KHAN N., TOMBARI F., TOMPKIN J., SITZMANN V., SRIDHAR S.: Neural Fields in Visual Computing and Beyond. *Computer Graphics Forum 41*, 2 (2022), 641–676. doi:10.1111/cgf.14505. 3, 5

[XZP*25] XIN L., ZHANG Z., PAN Z., WEI J., GAO D., GAO W.: DreamPBR: Text-driven High-Resolution SVBRDF Generation with Multimodal Guidance. In *2025 IEEE International Conference on Multimedia and Expo (ICME)* (2025), pp. 1–6. doi:10.1109/ICME59968.2025.11210202. 12, 18

[XZT*24] XIE Z., ZHANG Y., TANG X., WU Y., CHEN D., LI G., JIN X.: StyleTex: Style Image-Guided Texture Generation for 3D Models. *ACM Transactions on Graphics (TOG) 43*, 6 (2024). doi:10.1145/3687931. 12, 18, 21, 22

[YDL*23] YU X., DAI P., LI W., MA L., LIU Z., QI X.: Texture Generation on 3D Meshes with Point-UV Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4183–4193. doi:10.1109/ICCV51070.2023.00388. 14, 18

[YDPT21] YU R., DONG Y., PEERS P., TONG X.: Learning Texture Generators for 3D Shape Collections from Internet Photo Sets. In *British Machine Vision Conference* (2021). 2, 8, 10, 11, 18, 20

[YGC*24] YANG M., GUO J., CHEN Y., CHEN L., LI P., CHENG Z., ZHANG X., HUANG H.: InstanceTex: Instance-level Controllable Texture Synthesis for 3D Scenes via Diffusion Priors. In *SIGGRAPH Asia Conference Papers* (2024). doi:10.1145/3680528.3687633. 13, 18, 22

[YGS*21] YIN K., GAO J., SHUGRINA M., KHAMIS S., FIDLER S.: 3DStyleNet: Creating 3D Shapes with Geometric and Texture Style Variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12462–12471. doi:10.1109/ICCV48922.2021.01226. 10, 18

[YHK*24] YEH Y.-Y., HUANG J.-B., KIM C., XIAO L., NGUYEN-PHUOC T., KHAN N., ZHANG C., CHANDRAKER M., MARSHALL C. S., DONG Z., LI Z.: TextureDreamer: Image-Guided Texture Synthesis through Geometry-Aware Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4304–4314. doi:10.1109/CVPR52733.2024.00412. 2, 8, 12, 13, 18, 22

[YJPMO24] YOUWANG K., JO Y., PONS-MOLL G., OH T.-H.: Paint-it: Text-to-Texture Synthesis via Deep Convolutional Texture Map Optimization and Physically-Based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.00416. 12, 18

[YKSH19] YÜKSEL C., KALDOR J. M., SANDER P. V., HOPPE H.: Rethinking Texture Mapping. *Computer Graphics Forum 38*, 2 (2019), 535–551. doi:10.1111/cgf.13656. 3, 4

[YLLT15] YANG L., LUO P., LOY C. C., TANG X.: A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2015), pp. 3973–3981. doi:10.1109/CVPR.2015.7299023. 20

[YYG*24] YU X., YUAN Z., GUO Y.-C., LIU Y.-T., LIU J., LI Y., CAO Y.-P., LIANG D., QI X.: Texgen: a generative diffusion model for mesh textures. *ACM Transactions on Graphics (TOG) 43*, 6 (2024), 1–14. doi:10.1145/3687909. 14, 15, 18

[YYS*25] YUAN Z., YU X., SUN Y., GUO Y.-C., CAO Y.-P., LIANG D., QI X.: SeqTex: Generate Mesh Textures in Video Sequence. In *SIGGRAPH Asia 2025 Conference Papers* (New York, NY, USA, 2025), SA Conference Papers '25, Association for Computing Machinery. URL: https://doi.org/10.1145/3757377.3763863, doi:10.1145/3757377.3763863. 14, 18

[YZL*23] YE H., ZHANG J., LIU S., HAN X., YANG W.: IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2308.06721* (2023). doi:10.48550/arXiv.2308.06721. 7, 9, 13

[ZCQ*24] ZENG X., CHEN X., QI Z., LIU W., ZHAO Z., WANG Z., FU B., LIU Y., YU G.: Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4252–4262. doi:10.1109/CVPR52733.2024.00407. 13, 18

[ZHWH24] ZHANG Q., HOU J., WANG W., HE Y.: Flatten Anything: Unsupervised Neural Surface Parameterization. *Advances in Neural Information Processing Systems 37* (2024), 2830–2850. URL: https://proceedings.neurips.cc/paper_files/paper/2024/hash/054f771d614df12fe8def8ecdbe4e8e1-Abstract-Conference.html. 4

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 586–595. doi:10.1109/CVPR.2018.00068. 21

[ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). URL: https://arxiv.org/abs/1605.04797, doi:10.48550/arXiv.1605.04797. 19

[ZK14] ZHOU Q.-Y., KOLTUN V.: Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras. *ACM Transactions on Graphics 33*, 4 (2014), 155:1–155:10. doi:10.1145/2601097.2601132. 10

[ZLL*25] ZHAO Z., LAI Z., LIN Q., ZHAO Y., LIU H., YANG S., FENG Y., YANG M., ZHANG S., YANG X., ET AL.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025). URL: https://github.com/Tencent-Hunyuan/Hunyuan3D-2, doi:10.48550/arXiv.2501.12202. 17

[ZLX*24] ZHANG Y., LIU Y., XIE Z., YANG L., LIU Z., YANG M., ZHANG R., KOU Q., LIN C., WANG W., JIN X.: DreamMat: High-quality PBR Material Generation with Geometry- and Light-aware Diffusion Models. *ACM Transactions on Graphics (TOG)* (2024). doi:10.1145/3658170. 12, 18, 22

[ZLZ*24] ZHANG J., LI X., ZHANG Q., CAO Y., SHAN Y., LIAO J.: HumanRef: Single Image to 3D Human Generation via Reference-Guided Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.00181. 17

[ZLZS20] ZHAO F., LIAO S., ZHANG K., SHAO L.: Human Parsing Based Texture Transfer From Single Image to 3D Human via Cross-View Consistency. In *Advances in Neural Information Processing Systems* (2020), vol. 33, pp. 14326–14337. URL: https://proceedings.neurips.cc/paper/2020/hash/a516a87cfcaef229b342c437fe2b95f7-Abstract.html. 16, 17

[ZMY*20] ZHANG C., MILLER B., YAN K., GKIOULEKAS I., ZHAO S.: Path-Space Differentiable Rendering. *ACM Transactions on Graphics 39*, 4 (2020), 143:1–143:19. doi:10.1145/3386569.3392383. 4

[ZPX*24] ZHANG S., PENG S., XU T., YANG Y., CHEN T., XUE N., SHEN Y., BAO H., HU R., ZHOU X.: MaPa: Text-driven Photorealistic Material Painting for 3D Shapes. In *ACM SIGGRAPH 2024 Conference Papers* (2024). doi:10.1145/3641519.3657504. 12, 13, 18

[ZPZ*24] ZHANG H., PAN Z., ZHANG C., ZHU L., GAO X.: TexPainter: Generative Mesh Texturing with Multi-view Consistency. In *ACM SIGGRAPH Conference Papers* (2024). doi:10.1145/3641519.3657494. 14, 18

[ZRA23] ZHANG L., RAO A., AGRAWALA M.: Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 3813–3824. doi:10.1109/ICCV51070.2023.00355. 7

[ZSD*21] ZHANG X., SRINIVASAN P. P., DENG B., DEBEVEC P., FREEMAN W. T., BARRON J. T.: NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *ACM Transactions on Graphics 40*, 6 (2021), 237:1–237:18. doi:10.1145/3478513.3480496. 23

[ZST*15] ZHENG L., SHEN L., TIAN L., WANG S., WANG J., TIAN Q.: Scalable Person Re-Identification: A Benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (December 2015), pp. 1116–1124. doi:10.1109/ICCV.2015.133. 20

[ZWC*24] ZHANG C., WANG Y., CARRASCO F. V., WU C., YANG J., BEELER T., DE LA TORRE F.: FabricDiffusion: High-Fidelity Texture Transfer for 3D Garments Generation From In-the-Wild Clothing Images. In *SIGGRAPH Asia 2024 Conference Papers* (2024). doi:10.1145/3680528.3687637. 1, 15, 17, 18, 20, 21, 22

[ZWZ*24] ZHANG L., WANG Z., ZHANG Q., QIU Q., PANG A., JIANG H., YANG W., XU L., YU J.: CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Trans. Graph. 43*, 4 (July 2024). doi:10.1145/3658146. 17, 18

[ZXW*25] ZHANG Y., XU H., WU Y., CHEN S., LIN S., LI X., GAO X., JIN X.: AlignTex: Pixel-Precise Texture Generation from Multi-view Artwork. *ACM Transactions on Graphics (TOG) 44*, 4 (2025), 1–12. doi:10.1145/3731158. 15, 18

[ZYY24] ZHANG Z., YANG Z., YANG Y.: SIFU: Side-View Conditioned Implicit Function for Real-World Usable Clothed Human Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). doi:10.1109/CVPR52733.2024.00948. 17

[ZZMC25] ZHANG Y., ZHANG L., MA R., CAO N.: TexVerse: A Universe of 3D Objects with High-Resolution Textures, 2025. URL: https://arxiv.org/abs/2508.10868, arXiv:2508.10868, doi:10.48550/arXiv.2508.10868. 19, 20