MaPa: Text-driven Photorealistic Material Painting for 3D Shapes

Shangzhan Zhang[1,2] Sida Peng[1] Tao Xu[1] Yuanbo Yang[1] Tianrun Chen[1]
Nan Xue[2] Yujun Shen[2] Hujun Bao[1] Ruizhen Hu[3] Xiaowei Zhou[1]
[1]Zhejiang University [2]Ant Group [3]Shenzhen University

Figure 1: Examples from MaPa Gallery, which facilitates photo-realistic 3D rendering by generating materials for daily objects.

ABSTRACT

This paper aims to generate materials for 3D meshes from text descriptions. Unlike existing methods that synthesize texture maps, we propose to generate segment-wise procedural material graphs as the appearance representation, which supports high-quality rendering and provides substantial flexibility in editing. Instead of relying on extensive paired data, *i.e.*, 3D meshes with material graphs and corresponding text descriptions, to train a material graph generative model, we propose to leverage the pre-trained 2D diffusion model as a bridge to connect the text and material graphs. Specifically, our approach decomposes a shape into a set of segments and designs a segment-controlled diffusion model to synthesize 2D images that are aligned with mesh parts. Based on generated images, we initialize parameters of material graphs and fine-tune them through the differentiable rendering module to produce materials in accordance with the textual description. Extensive experiments demonstrate the superior performance of our framework in photorealism, resolution, and editability over existing methods. Project page: https://zhanghe3z.github.io/MaPa/.

CCS CONCEPTS

• **Computing methodologies** → *Appearance and texture representations.*

KEYWORDS

material painting, 3D asset creation, generative modeling

1 INTRODUCTION

3D content generation has attracted increasing attention due to its wide applications in video games, movies and VR/AR. With the integration of diffusion models [Ho et al. 2020; Rombach et al. 2022; Song et al. 2020] into 3D generation [Cheng et al. 2023; Liu et al. 2023; Poole et al. 2022], this field has witnessed remarkable advancements and impressive results. Many works focus on generating 3D objects [Lin et al. 2023; Liu et al. 2023; Long et al. 2023; Poole et al. 2022] from text prompts or a single image. In addition to 3D shape generation, generating appearance for existing meshes is also an important problem. The conventional process of designing appearances for meshes involves extensive and laborious manual work, creating a pressing need within the community for a more efficient approach to producing mesh appearance. Many 3D texture synthesis methods [Cao et al. 2023; Chen et al. 2023b, 2022; Metzer et al. 2022; Richardson et al. 2023; Yeh et al. 2024; Yu et al. 2023] have been proposed to solve this problem, which can create diverse 3D textures for meshes according to the text input by users.

However, creating textures alone doesn't fully meet the needs of downstream applications, as we often render meshes under various lighting conditions. For example, in video games, choosing the right materials for rendering meshes in a scene is critical, as this impacts how the mesh interacts with the light around it, which in turn affects its appearance and realism. Therefore, designing an algorithm for painting materials on meshes is important. Despite the remarkable success in texture generation, there is a lack of research focusing on generating high-quality materials for 3D objects.

Recently, Fantasia3D [Chen et al. 2023a] tries to generate 3D objects with materials by distilling the appearance prior from 2D generative models. However, this approach faces substantial obstacles due to optimization instability and often fails to yield high-quality materials. Moreover, it models the object material as a per-point representation, which could be inconvenient for downstream user modifications.

In this paper, our goal is to generate photorealistic and high-resolution materials for meshes from textual descriptions, which can be conveniently edited by users. We observe that the materials of newly manufactured objects in real life often exhibit consistency within specific areas due to the nature of the manufacturing processes. Motivated by this, we hope that our generation process can mimic this characteristic, which has the advantage of leading to more organized results and allowing users to effortlessly swap the material of a specific area in modeling software, without the need to recreate the entire material map. However, it is still an unsolved problem to separately generate materials for each part of the mesh and ensure the consistency of materials within the local region.

To this end, we propose a novel framework, named MaPa, to generate materials for a given mesh based on text prompts. Our key idea is to introduce a segment-wise procedural material graphs representation for text-driven material painting and leverage 2D images as a bridge to connect the text and materials. Procedural material graphs [Hu et al. 2022a; Li et al. 2023; Shi et al. 2020], a staple in the computer graphics industry, consist of a range of simple image processing operations with a set of parameters. They are renowned for their high-quality output and resolution independence, and provide substantial flexibility in editing. The segment-wise representation mimics the consistency of materials in the real world, making the results look more realistic and clean. However, to train a generative model that directly creates material graphs for an input mesh, extensive data for pairs of texts and meshes with material graphs need to be collected. Instead, we leverage the pre-trained 2D diffusion model as an intermediate bridge to guide the optimization of procedural material graphs and produce diverse material for given meshes according to textual descriptions.

Specifically, our method contains two main components: segment-controlled image generation and material graph optimization. Given a mesh, it is firstly segmented into several segments and projected onto a particular viewpoint to produce a 2D segmentation image. Then, we design a segment-controlled diffusion model to generate an RGB image from the 2D segmentation. In contrast to holistically synthesizing images [Richardson et al. 2023], conditioning the diffusion model on the projections of 3D segments can create 2D images that more accurately align with parts of the mesh, thereby enhancing the stability of the subsequent optimization process. Subsequently, we produce the object material by estimating segment-wise material graphs from the generated image. The material graphs are initialized by retrieving the most similar ones from our collected material graph library, and then their parameters are optimized to fit the image through a differentiable rendering module. The generated materials can then be imported into existing commercial software for users to edit and generate various material variants conveniently.

We extensively validate the effectiveness of our method on different categories of objects. Our results outperform three strong baselines in the task of text-driven appearance modeling, in terms of FID and KID metrics, as well as in user study evaluation. We also provide qualitative comparisons to the baselines, showcasing our superior photorealistic visual quality. Users can easily use our method to generate high-quality materials for input meshes through text and edit them conveniently. Meanwhile, since we can generate arbitrary-resolution and tileable material maps, we can render fine material details at high resolution.

2 RELATED WORKS

2D diffusion models. Recently, diffusion models [Ho et al. 2020; Nichol et al. 2021; Rombach et al. 2022; Song et al. 2020] have been dominating the field of image generation and editing. These works can generate images from text prompts, achieving impressive results. Among them, Stable Diffusion [Rombach et al. 2022] is a large-scale diffusion model that is widely used. ControlNet [Zhang et al. 2023] introduces spatial conditioning controls (e.g., inpainting masks, Canny edges [Canny 1986], depth maps) to pretrained text-to-image diffusion models. These signals provide more precise control over the diffusion process. Another line of work focuses on controlling diffusion models with exemplar images. Textual Inversion [Gal et al. 2022] represents the exemplar images as learned tokens in textual space. DreamBooth [Ruiz et al. 2023] finetunes the diffusion model and uses several tricks to prevent overfitting. IP-Adapter [Ye et al. 2023] trains a small adapter network to map the exemplar image to the latent space of the diffusion model.

Appearance modeling. With the emergence of diffusion models, some works begin to use diffusion models to generate textures for 3D objects. Latent-nerf [Metzer et al. 2022] extends DreamFusion to optimize the texture map for a given 3D object. However, Latent-nerf requires a long training time and produces low-quality texture maps. TEXTure [Richardson et al. 2023] and Text2tex [Chen et al. 2023b] utilize a pretrained depth-to-image diffusion model to iteratively paint a 3D model from different viewpoints. TexFusion [Cao et al. 2023] introduces an iterative aggregation scheme from different viewpoints during the denoising process. These works greatly improve the quality of the generated texture map and speed up the texture generation process. However, these works can only generate texture maps, not realistic materials. Material modeling [Guarnera et al. 2016] has long been a problem of interest in computer graphics and vision. PhotoShape [Park et al. 2018] uses shape collections, material collections, and image collections to learn material assignment to 3D shapes based on given images, where the shape and images need to be precisely aligned. TMT [Hu et al. 2022b] addresses the limitations of data constraints in PhotoShape by allowing different shape structures. However, its successful implementation necessitates a substantial quantity of 3D data with part segmentation [Mo et al. 2019] for training, which makes it can only work on a few categories of objects. Moreover, both of those works treat material modeling as a classification problem and can only use materials from the pre-collected dataset, which often leads to incorrect material prediction under challenging lighting and dissimilar material assignment due to the restriction of dataset. MATch [Shi et al. 2020] and its extension [Li et al. 2023] introduce a differentiable procedural material graphs. Procedural material graphs are popular in the graphics industry, which can generate resolution-invariant,

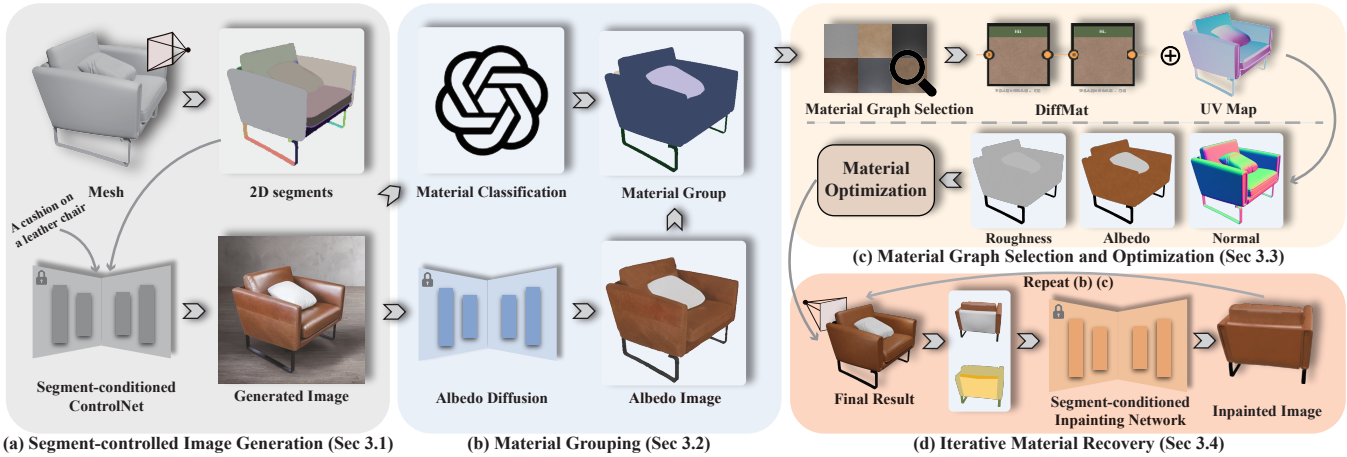

Figure 2: Illustration of our pipeline. Our pipeline primarily consists of four steps: a) Segment-controlled image generation. First, we decompose the input mesh into various segments, project these segments onto 2D images, and then generate the corresponding images using the segment-controlled ControlNet. b) Material grouping. We group segments that share the same material and have similar appearance into a material group. c) Material graph selection and optimization. For each material group, we select an appropriate material graph based on generated images and then optimize this material graph. d) Iterative material recovery. We render additional views of the input mesh with the optimized material graphs, inpaint the missing regions in these rendered images, and repeat steps b) and c) until all segments are assigned with material graphs.

tileable, and photorealistic materials. PhotoScene [Yeh et al. 2022] and PSDR-Room [Yan et al. 2023] produce the materials of indoor scenes from single images using differentiable procedural material graphs. Different from these works, we focus on assigning realistic materials to a given object. The closest work is Fantasia3D [Chen et al. 2023a], which follows DreamFusion and incorporates a spatially varying bidirectional reflectance distribution function (BRDF) to learn the surface materials for generated 3D objects. However, the process of distilling 3D materials from a 2D pretrained diffusion model is not stable, and often fails to generate reasonable materials. Moreover, the materials generated by Fantasia3D are often cartoonish and unrealistic.

Material and texture perception. Material perception and texture recognition are long-standing problems. A considerable amount of earlier research on material recognition has concentrated on addressing the classification problem [Bell et al. 2015; Cimpoi et al. 2014; Hu et al. 2011; Liu et al. 2010]. Other works [Sharma et al. 2023; Upchurch and Niu 2022; Zheng et al. 2023] attempt to solve material segmentation problem. As for texture perception, a number of traditional approaches treat texture as a set of handcrafted features [Caputo et al. 2005; Guo et al. 2010; Lazebnik et al. 2005]. Recently, deep learning methods [Bruna and Mallat 2013; Chen et al. 2021; Cimpoi et al. 2015; Fujieda et al. 2017] have been introduced to address the problem of texture perception. Our work benefits from these works, and we empirically find that GPT-4v [OpenAI 2023] can be applied to material classification and texture description tasks in a zero-shot manner.

3 METHOD

Given a mesh and a textual prompt, our method aims to produce suitable materials to create photorealistic and relightable 3D shapes. The overview of the proposed model is illustrated in Figure 2. Section 3.1 first describes our proposed segment-controlled image generation. The image-based material optimization is divided into three components: material grouping, material graph selection and optimization, and iterative material recovery. Section 3.2 provides details on using the generated image for material grouping. Section 3.3 introduces how to select initial material graphs from the material graph library and optimize the parameters of procedural node graphs according to the generated image. Then, we design an iterative approach to generate materials for segments of the mesh that have not been assigned materials in Section 3.4. Finally, we introduce a downstream editing module in Section 3.5 to allow users to edit the generated materials conveniently.

3.1 Segment-controlled image generation

Given a mesh, we first oversegment it into a series of segments $\{\mathbf{s_i}\}_{i=1}^{N}$. During the modeling process, designers often create individual parts, which are later assembled into a complete model. Hence, our method can separate these components by grouping the connected components. If the mesh is scanned or watertight, we can also use existing techniques [Katz and Tal 2003] to decompose it into a series of segments. Motivated by the progress of 2D generative models [Ho et al. 2020; Rombach et al. 2022; Zhang et al. 2023], we opt to project these segments to a viewpoint to generate 2D segmentation masks and perform subsequent conditional generation in 2D space. For the good performance of conditional 2D generation, we carefully select a viewpoint to ensure a good initial result. Specifically, a set of viewpoints are uniformly sampled with a 360-degree azimuth range, a constant 25-degree elevation angle, and a fixed radius of 2 units from the mesh. We project the 3D segments onto these viewpoints and choose the one with the highest number of 2D segments as our starting viewpoint. If multiple viewpoints have the same number of 2D segments, we empirically

choose the viewpoint with the largest projected area of the mesh, as it is more likely to contain more details. The 2D segments are sorted by area size and numbered from large to small to form the 2D segmentation mask.

From this viewpoint, the 2D segmentation mask is used to guide the generation of the corresponding RGB image, which then facilitates the process of material prediction that follows. Note that previous works [Cao et al. 2023; Chen et al. 2023b; Richardson et al. 2023] usually use depth-to-image diffusion model to generate image conditioned on meshes, which may result in multiple color blocks within a segment and lead to unstable material optimization, thus we choose ControlNet with SAM mask [Kirillov et al. 2023] as condition [Gao et al. 2023] for our segment-controlled image generation. This process is defined as:

$$\begin{aligned} \{\hat{\mathbf{s}}_\mathbf{i}\}_{i=1}^{M} &= \mathcal{P}(\{\mathbf{s}_\mathbf{i}\}_{i=1}^{N}), \\ z_{\mathrm{t}-1} &= \boldsymbol{\epsilon}_\theta(z_\mathrm{t}, \boldsymbol{t}, \{\hat{\mathbf{s}}_\mathbf{i}\}_{i=1}^{M}), \end{aligned} \tag{1}$$

where $\mathcal{P}$ is the projection function, and $\boldsymbol{\epsilon}_\theta$ is the pre-trained ControlNet. z_t is the noised latent at time step t, $\boldsymbol{t}$ is the time step, and $\{\hat{\mathbf{s}}_\mathbf{i}\}_{i=1}^{M}$ is the 2D segments. We finetune the SAM-conditioned ControlNet [Gao et al. 2023] on our own collected dataset to obtain a segment-conditioned ControlNet. The implementation details of segment-conditioned ControlNet and the comparison to depth-to-image diffusion model are presented in the supplementary material.

3.2 Material grouping

To save the optimization time and obtain visually more coherent results, we first merge segments with similar appearance into groups and generate one material graph per group, instead of directly optimizing the material for each segment. Specifically, we build a graph with each segment as a node, and connect two segments if they have the same material class and similar colors. Then, we extract all connected components from the graph as the grouping results.

In more details, for material classification, we use GPT-4v [OpenAI 2023] due to its strong visual perception capabilities and convenient in-context learning abilities. We empirically find that GPT-4v works well with images created by a diffusion model. The details of the prompt are presented in the supplementary material. For the color similarity, we train an albedo estimation network to remove effects caused by the shadows and strong lighting in the generated images. We fine-tune Stable Diffusion conditioned on CLIP image embeddings on the ABO material dataset [Collins et al. 2022], with a RGB image as the input and the corresponding albedo image as the output. The implementation details of albedo estimation network are presented in the supplementary material. If the distance between the median color of two segments in the albedo image in the CIE color space [Sharma et al. 2005] is less than a threshold λ, these two segments are considered to be similar. We empirically set λ to 2, which means the difference is perceptible through close observation [Minaker et al. 2021].

3.3 Material graph selection and optimization

In this section, we describe how to recover the parameters of procedural node graphs from the generated images. For each material group, we retrieve the most similar material graph from the material graph library. Then, we optimize the material graph to obtain the final material graph.

Material graph selection. We build a material graph library in advance, which contains a variety of material categories. Thumbnails of material graphs are rendered using eight different environment maps to enhance the robustness of the retrieval process. Following PSDR-Room [Yan et al. 2023], we adopt CLIP for zero-shot retrieval of material graphs. We crop the material group region from the generated image and compute its cosine similarity with same-class material graphs in our library through CLIP model. The material graph with the highest similarity is selected. Similar to PhotoScene [Yeh et al. 2022], we apply homogeneous materials to material groups with an area smaller than 1000 pixels, as it is challenging to discern spatial variations in such small area.

Material graph optimization. To make the retrieved material graph close to the generated image, we optimize the parameters of the material graph using our differentiable rendering module. The differentiable rendering module contains a differentiable renderer and DiffMat v2 [Li et al. 2023]. DiffMat v2 [Li et al. 2023] is a framework that differentiably converts material graphs into texture-space maps, such as albedo map $\mathbf{A_{uv}}$, normal map $\mathbf{N_{uv}}$, roughness map $\mathbf{R_{uv}}$, etc. We utilize UV maps generated by Blender to sample per-pixel material maps $\mathbf{A}$, $\mathbf{N}'$, $\mathbf{R}$. The normals $\mathbf{N}'$ should be rotated to align with the local shading frame to obtain the final normal map $\mathbf{N}$. We use physically-based microfacet BRDF [Karis 2013] as our material model. Our lighting model is 2D spatially-varying incoming lighting [Li et al. 2020]. The incoming lighting is stored in a two-dimensional grid that varies spatially. Using this lighting model significantly enhances the efficiency of our rendering process, which eliminates the need to trace additional rays. InvRenderNet [Li et al. 2020] is a network that can predict the 2D spatially-varying lighting from a single image. We use the prediction of InvRenderNet as our initial lighting $\mathbf{L_{init}}$. During training, we optimize the lighting $\mathbf{L}$ by adding residual lighting $\widetilde{\mathbf{L}}$ to $\mathbf{L_{init}}$. The information of $\widetilde{\mathbf{L}}$ is stored in a 2D hash grid [Müller et al. 2022]. For each pixel, we query the corresponding feature from the hash grid and use a shallow MLP to convert it into a residual lighting. We set the initial value of $\widetilde{\mathbf{L}}$ to 0, and use the ReLU activation function to prevent the value of $\mathbf{L}$ from being negative. The differentiable rendering module is described as:

$$\begin{aligned} \mathbf{A_{uv}}, \mathbf{N_{uv}}, \mathbf{R_{uv}} &= \mathrm{DiffMat}(\mathbf{G}), \\ \mathbf{A}, \mathbf{R} &= \mathrm{S}(\mathbf{A_{uv}}, \mathbf{R_{uv}}), \\ \mathbf{N} &= \mathrm{Rot}(\mathrm{S}(\mathbf{N_{uv}})), \\ \mathbf{I}_{rend} &= \mathcal{R}(\mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{L}), \end{aligned} \tag{2}$$

where $\mathbf{G}$ is the procedural material graph, S is the UV sampling function, Rot is the rotation function that rotates the normals $\mathbf{N}'$, and $\mathcal{R}$ is the differentiable renderer adopted from InvRenderNet [Li et al. 2020]. We can render a 2D image using the differentiable rendering module, and then optimize the parameters of the material graph by making the rendered image as similar as possible to the generated image. However, images generated by the diffusion model typically have strong lighting and shadows. To avoid overfitting, we introduce a regularization term that constrains the

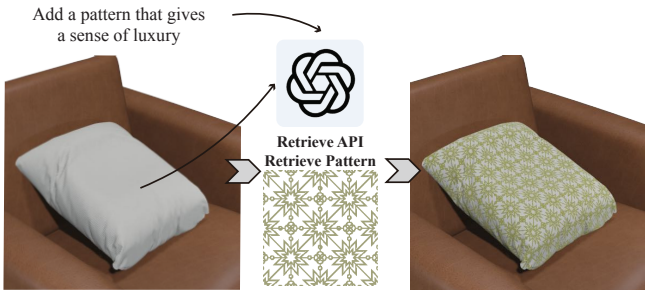

Figure 3: Downstream editing. We perform material editing on generated material. The user can edit the material using textual prompts through the GPT-4 and a set of predifined APIs.

rendering result of the albedo map A of the material graph to be as similar as possible to the albedo image predicted by the network mentioned in Section 3.2. Additionally, differentiable procedural material graphs highly constrain the optimization space of the material model, which can also alleviate the instability during the training process due to the presence of strong lighting and shadows. The loss functions and more technical details are presented in the supplementary material.

3.4 Iterative material recovery

The aforementioned process can only generate materials for one viewpoint of the mesh. For areas without material, we adopted an iterative approach to inpaint them. Among all the viewpoints generated in Section 3.1, we select the viewpoint adjacent to the initial one for the next iteration. During this iteration, it is necessary to determine which segments require material assignment. A segment is considered in need of material if it has not yet been assigned one, or if its projected area in the previous viewpoint is less than a quarter of its projected area in the current viewpoint.

To ensure the consistency of the inpainting results, we adopt a common design [Dihlmann et al. 2024; Zeng 2023] in the community. We render the mesh with partially assigned materials both from the previous iteration viewpoints and the current iteration viewpoint using Blender Cycles Renderer. Then, the previous iteration viewpoint images and the current iteration viewpoint image are concatenated as the input of the inpainting network. Only the segments that need material assignment are inpainted, and the inpainting mask of other regions is set to 0. The SAM-conditioned inpainting network [Gao et al. 2023] is adopted as our inpainting network. We employ the process described in Section 3.3 again to generate materials for these segments. If segments requiring material assignment and those already assigned material are grouped into the same material group, we directly copy the existing material graph to the segments requiring material. Above steps are repeated until all regions are assigned materials.

3.5 Downstream editing

Different from previous work [Cao et al. 2023; Chen et al. 2023b,a; Richardson et al. 2023], our method can generate editable material graphs, which is paramount for users. Users can edit the material graph directly in Substance Designer [Adobe 2021]. For example, they can change the noise that affects the base color to change the texture, or add some nodes that affect the roughness to add details such as scratches and other surface imperfections.

However, it is difficult for most users to operate the underlying noise nodes which are complex and intertwined. Inspired by the work of VISPROG [Gupta and Kembhavi 2023], we provide users with some high-level operations, allowing them to easily modify the material through textual instructions. For example, when users input the requirement "add a pattern that gives a sense of luxury", our method will find a geometric seamless pattern mask that matches the description and generate the corresponding python command to add such pattern so that the user can run this command locally to obtain the desired material modification. One example result of adding texture is shown in Figure 3. More details about this function can be found in the supplementary material.

4 EXPERIMENTS

4.1 Implementation details

Our material graph optimization in three stages uses Adam [Kingma and Ba 2014] as the optimizer. We perform a total of 300 iterations for optimization. In general, the entire process of our method can be completed in less than 10 minutes on an A6000, which is faster than Text2tex [Chen et al. 2023b] (15 minutes) and Fantasia3D [Chen et al. 2023a] (25 minutes). We collect 8 material categories, including wood, metal, plastic, leather, fabric, stone, ceramic, and rubber. The number of each category is given in the supplementary material. Since the eight materials mentioned above already cover the vast majority of man-made objects, we did not collect additional types of material graphs.

4.2 Comparison to baselines

Baselines. We compare our method with three strong baselines on text-driven appearance modeling. TEXTure [Richardson et al. 2023] introduces an iterative texture painting scheme to generate texture maps for 3D objects. Text2tex [Chen et al. 2023b] uses a similar scheme to TEXTure. Additionally, it creates an automated strategy for selecting viewpoints during the iterative process. Fantasia3D [Chen et al. 2023a] separates the modeling of geometry and appearance, adopting an MLP to model the surface material properties of a 3D object. For the following experiments, we employ an environmental map to generate outcomes for our method and Fantasia3D. The results for TEXTure and Text2tex are derived from their diffuse color pass in Blender.

Quantitative comparison. Our goal is to obtain high rendering quality that is as close as possible to the real image, so we measure the distribution distance between the rendered image and the real image using FID [Heusel et al. 2017] and KID [Binkowski et al. 2018]. We select 300 chairs from TMT [Hu et al. 2022b] and 30 random objects from ABO dataset [Collins et al. 2022] as test data. 10 views are uniformly sampled for each chair with a 360-degree azimuth range, a constant 45-degree elevation angle, and a fixed radius of 1.5 units. The prompt in the experiment is extracted from the rendering of original objects using BLIP [Li et al. 2022]. Each

Table 1: Comparison results. We compare our method with three strong baselines. Our approach yields superior quantitative results and attains the highest ratings in user studies.

Dataset	Methods	FID↓	KID↓	Overall quality↑	Fidelity↑
Chair	TEXTure	94.6	0.044	2.43	2.41
	Text2tex	102.1	0.048	2.35	2.51
	Fantasia3D	113.7	0.055	1.98	2.08
	Ours	**88.3**	**0.037**	**4.33**	**3.35**
ABO	TEXTure	118.0	0.027	2.65	2.41
	Text2tex	109.8	0.020	3.10	2.94
	Fantasia3D	138.1	0.034	1.51	1.80
	Ours	**87.3**	**0.014**	**4.10**	**3.22**

baseline uses the same prompts. The segmentation of the each input shape is obtained by finding connected components.

Moreover, we also conducted a user study to assess the rendering results. Each respondent was asked to evaluate the results based on two aspects: overall quality and fidelity to the text prompt, using a scale of 1 to 5. Overall quality refers to the visual quality of the rendering results, while fidelity to the text prompt indicates how closely these results align with the given text prompt. There were 40 users participated in the study, including 16 professional artists and 24 members of the general public. We gathered 60 responses from each participant, and the final score represents the average of all these responses. As shown in Table 1, our method achieves the best quantitative results and the highest user study scores.

Qualitative comparison. We collect several high-quality 3D models from Sketchfab [SketchFab 2012] and use these data for qualitative experiments. As shown in Figure 4, our method can achieve more realistic rendering results than the baselines. TEXTure and Text2tex are prone to produce inconsistent textures, which makes their results look dirty. Fantasia3D's results frequently exhibit oversaturation or fail directly.

4.3 More qualitative results

Diversity of results. 2D diffusion model can generate different images with different seeds, which naturally brings diversity to our framework when optimizing material according to the generated images. We show an example shape with diverse results in Figure 5. Note how the painted materials change when different images of the same object are generated.

Appearance transfer from image prompt. We can use the image encoder introduced in IP-Adapter [Ye et al. 2023] to be compatible with image prompt. As shown in Figure 6, our method can transfer the appearance from the reference image to the input objects by first generating the corresponding image of the given object with similar appearance and then optimizing the materials using our framework.

Results on watertight meshes. Although we mainly focus on meshes that can be segmented by grouping connected components, our method can also handle watertight meshes. In Figure 9, we also show the results of painted watertight meshes. We use the graph cut algorithm [Katz and Tal 2003] to segment the watertight mesh, and then we generate materials for each segment.

Figure 4: Qualitative comparisons. The results generated by our method and all the baselines are rendered in the same CG environment for comparison. The prompts for the three objects are: "a photo of a wooden bedside table," "a photo of a toy rocket," and "a photo of a brand-new sword."

Figure 5: Diversity of our generated material. We show the diversity of results synthesized by our framework with the same prompt: "A photo of a toy airplane". Images in the first row are generated by diffusion models, and models in the second row are our painted meshes.

Downstream editing results. Our method can edit materials based on user-provided prompts, and we show several editing results on a bag in Figure 7. We can see that we are able to add different patterns to the bag according to the prompts.

4.4 Ablation studies

The significance of albedo estimation network. To justify the usage of our albedo estimation network, we compare with the setting

Figure 6: Appearance transfer. Our method can also take image prompt as input, transferring the appearance of reference images to input objects.

"w/o Albedo", where we remove this module from our method and use the generated image directly. As illustrated in Figure 10, the baseline "w/o Albedo" fails to merge the segments with the similar color, hindered by the impacts of shadow and lighting. Additionally, a noticeable color disparity exists between the rendering result of "w/o Albedo" and the generated image.

Robustness analysis of grouping results. Our method can also work without material grouping. We show the results of our method without material grouping in Figure 10, referred as "w/o Grouping". The results of "w/o Grouping" are not significantly different from our results. However, the optimization time will be much longer without grouping. We find our method takes 7 minutes on average to finish the all optimization process per shape. If we remove the material grouping, the optimization time on average will increase to 33 minutes.

The significance of material graph optimization. We also show some representative visual comparisons with the setting "w/o Opt", where the material graph is selected without further optimization in Figure 11. We can clearly see that the optimization highly increases the appearance similarity to the generated images and is not limited by the small material graph data set. Generated images (a), (b), and (c) are generated under the same pose. The retrieval results of the wood material graph in the three images are the same shown as "w/o Opt". This proves that even if our material graph data set is small, our method can still produce diverse texture results. We also conduct an ablation study on effectiveness of residual lighting, refereed as "w/o Res". Compared to our result (a), "w/o Res" produces textures that are inconsistent with the generated images. This is because the initial 2D spatially-varying lighting is not very accurate, which is caused by the domain gap between the training data of InvRenderNet and the images generated by the diffusion model.

5 CONCLUSIONS

In this paper, we propose a novel method for generating high-quality material map for input 3D meshes. We introduce segment-wise procedural material graphs representation for this challenging task and adopt a pre-trained 2D diffusion model as an intermediary

Figure 7: Results of downstream editing. We perform text-driven editing on generated material through the GPT-4 and a set of predifined APIs.

to guide the prediction of this representation. Our results yield photorealistic renderings that are also conveniently flexible for editing. It is worth exploring ways to train an amortized inference network that can directly predict the parameters of the material graph from the input image to highly improve the efficiency. Our limitations are as follows:

Domain gap of generated images. Our method sometimes fails to generate rendering results that match the generated image, as shown in Figure 8, mainly due to the different domain of the diffusion model and our albedo estimation network. This is because the diffusion model is trained on natural images, while our albedo estimation network is trained on synthetic images. We suggest that fine-tuning the diffusion model on synthetic images could mitigate this issue. Alternatively, employing a more advanced albedo estimation network could potentially serve as a replacement for our current albedo estimation network. Moreover, we currently use GPT-4v for material classification as we found that its performance is better than other prior works, and it would be easy to be replaced if more powerful methods for material classification come out.

Complex objects with unobvious segments. For complex objects with unobvious segments, we tend to over-segment the shape and then group the segments with similar materials according to the generated image. Final results will depend on over-segmentations and the segment-conditioned ControlNet may fail to produce reasonable guidance for grouping due to the lack of training data of such complex objects.

Expressiveness of the material graph. The material can be optimized to produce spatially varying appearance only if the material

Figure 8: Failure case. Because of the unusual light effects generated by diffusion (silver metallic with yellow light), the albedo estimation network fails to accurately estimate the albedo, leading to dissimilar material prediction.

graph contains some corresponding nodes for optimization, as the wood material shown in Figure 11. Currently, most of the other materials we used lack such nodes, and the complex look in Figure 7 indeed comes from the downstream editing by retrieving predefined patterns. We echo that the method could handle more complex targets if we have a larger collection of graphs with more expressiveness.

ACKNOWLEDGMENTS

We would like to thank Xiangyu Su for his valuable contributions to our discussions. This work was partially supported by NSFC (No. 62172364 and No. 62322207), Ant Group and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

REFERENCES

Adobe. 2021. 3D design software for authoring - Adobe Substance 3D — adobe.com. https://www.adobe.com/products/substance3d-designer.html.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE Computer Society, Boston, MA, USA, 3479–3487.

Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*. OpenReview.net, Vancouver, BC, Canada, pp.

Joan Bruna and Stéphane Mallat. 2013. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1872–1886.

John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 8, 6 (1986), 679–698.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. 2023. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Paris, France, 4169–4181.

Barbara Caputo, Eric Hayman, and P Mallikarjuna. 2005. Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. IEEE, 1597–1604.

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396* (2023).

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. 2022. TANGO: Text-driven Photorealistic and Robust 3D Stylization via Lighting Decomposition. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 30923–30936.

Zhile Chen, Feng Li, Yuhui Quan, Yong Xu, and Hui Ji. 2021. Deep texture recognition via exploiting cross-layer statistical self-similarity. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 5231–5240.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. 2023. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4456–4465.

Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. 2014. Deep convolutional filter banks for texture recognition and segmentation. *arXiv preprint arXiv:1411.6836* (2014).

Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. 2015. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3828–3836.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. 2022. ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. *CVPR* (2022).

Jan-Niklas Dihlmann, Andreas Engelhardt, and Hendrik Lensch. 2024. SIGNeRF: Scene Integrated Generation for Neural Radiance Fields. *arXiv preprint arXiv:2401.01647* (2024).

Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. 2017. Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv:1707.07394* (2017).

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. https://doi.org/10.48550/ARXIV.2208.01618

Shanghua Gao, Zhijie Lin, Xingyu Xie, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. 2023. EditAnything: Empowering Unparalleled Flexibility in Image Editing and Generation. In *Proceedings of the 31st ACM International Conference on Multimedia, Demo track*. ACM, Ottawa, ON, Canada.

Darya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. 2016. BRDF representation and acquisition. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 625–650.

Zhenhua Guo, Lei Zhang, and David Zhang. 2010. A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing* 19, 6 (2010), 1657–1663.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14953–14962.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

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

Diane Hu, Liefeng Bo, and Xiaofeng Ren. 2011. Toward Robust Material Recognition for Everyday Objects.. In *BMVC*, Vol. 2. Citeseer, 6.

Ruizhen Hu, Xiangyu Su, Xiangkai Chen, Oliver Van Kaick, and Hui Huang. 2022b. Photo-to-shape material transfer for diverse structures. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

Yiwei Hu, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. 2022a. An inverse procedural modeling pipeline for svbrdf maps. *ACM Transactions on Graphics (TOG)* 41, 2 (2022), 1–17.

Brian Karis. 2013. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice* 4, 3 (2013), 1.

Sagi Katz and Ayellet Tal. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM transactions on graphics (TOG)* 22, 3 (2003), 954–961.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2005. A sparse texture representation using local affine regions. *IEEE transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1265–1278.

Beichen Li, Liang Shi, and Wojciech Matusik. 2023. End-to-End Procedural Material Capture with Proxy-Free Mixed-Integer Optimization. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–15.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2475–2484.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. 2010. Exploring features in a bayesian framework for material recognition. In *2010 ieee computer society conference on computer vision and pattern recognition*. IEEE, 239–246.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023. Zero-1-to-3: Zero-shot One Image to 3D Object. arXiv:2303.11328 [cs.CV]

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2023. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008* (2023).

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2022. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. *arXiv preprint arXiv:2211.07600* (2022).

Samuel A Minaker, Ryan H Mason, and David R Chow. 2021. Optimizing Color Performance of the Ngenuity 3-Dimensional Visualization System. *Ophthalmology Science* 1, 3 (2021), 100054.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 909–918.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

R OpenAI. 2023. GPT-4 technical report. *arXiv* (2023), 2303–08774.

Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M Seitz. 2018. Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761* (2018).

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721* (2023).

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

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. 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*.

Gaurav Sharma, Wencheng Wu, and Edul N Dalal. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur* 30, 1 (2005), 21–30.

Prafull Sharma, Julien Philip, Michaël Gharbi, Bill Freeman, Fredo Durand, and Valentin Deschaintre. 2023. Materialistic: Selecting similar materials in images. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.

Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekeur, Radomir Mech, and Wojciech Matusik. 2020. Match: Differentiable material graphs for procedural material capture. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

SketchFab 2012. Sketchfab - The best 3D viewer on the web — sketchfab.com. https://sketchfab.com/.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

Paul Upchurch and Ransen Niu. 2022. A dense material segmentation dataset for indoor and outdoor scene parsing. In *European Conference on Computer Vision*. Springer, 450–466.

Kai Yan, Fujun Luan, Miloš Hašan, Thibault Groueix, Valentin Deschaintre, and Shuang Zhao. 2023. PSDR-Room: Single Photo to Scene using Differentiable Rendering. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. (2023).

Yu-Ying Yeh, Jia-Bin Huang, Changil Kim, Lei Xiao, Thu Nguyen-Phuoc, Numair Khan, Cheng Zhang, Manmohan Chandraker, Carl Marshall, Zhao Dong, and Zhengqin Li. 2024. TextureDreamer: Image-guided Texture Synthesis through Geometry-aware Diffusion. *ArXiv* abs/2401.09416 (2024).

Yu-Ying Yeh, Zhengqin Li, Yannick Hold-Geoffroy, Rui Zhu, Zexiang Xu, Miloš Hašan, Kalyan Sunkavalli, and Manmohan Chandraker. 2022. Photoscene: Photorealistic material and lighting transfer for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18562–18571.

Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. 2023. Texture Generation on 3D Meshes with Point-UV Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4206–4216.

Xianfang Zeng. 2023. Paint3D: Paint Anything 3D with Lighting-Less Texture Diffusion Models. *arXiv preprint arXiv:2312.13913* (2023).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models.

Junwei Zheng, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. 2023. MATERobot: Material Recognition in Wearable Robotics for People with Visual Impairments. *arXiv preprint arXiv:2302.14595* (2023).

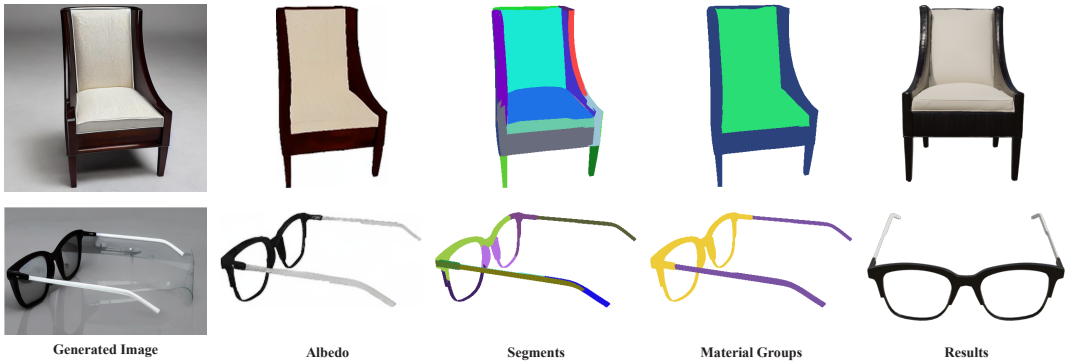

Figure 9: Results on watertight meshes. We decompose the mesh into several segments using the graph cut algorithm and generate materials for each segment. The text prompts for those two examples are "A photo of a modern chair with brown legs" and "A black and white eyeglass", respectively.

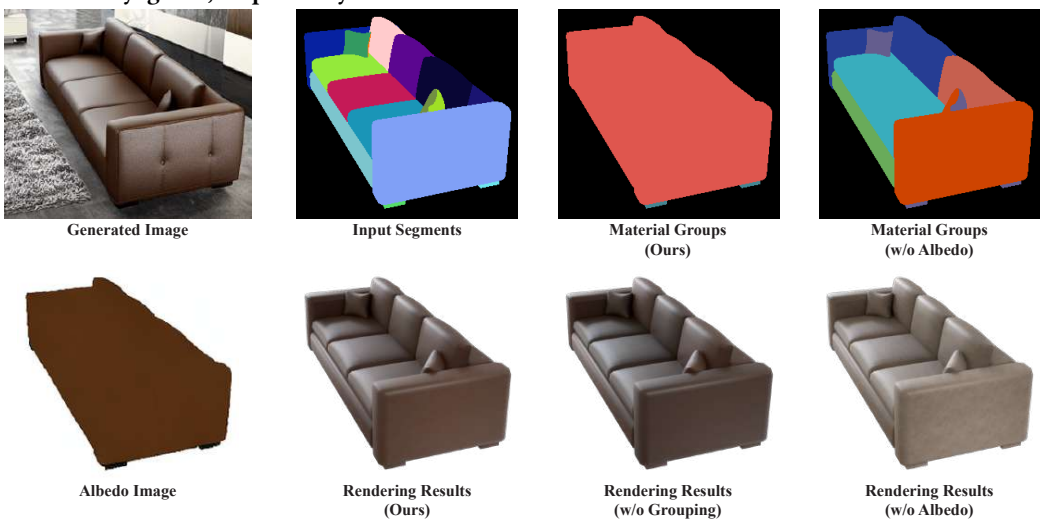

Figure 10: Ablation study of "w/o Albedo" and "w/o Grouping". Example with text prompt "A dark brown leather sofa". "w/o Albedo" exhibits color disparity between the rendering result and generated image, and fails to merge segments with similar color. "w/o Grouping" has similar rendering results to our method.

Figure 11: Ablation study of our material graph optimization. We show the ablation study of our material graph optimization and the effectiveness of residual light. The baseline "w/o Opt" and "w/o Res" are optimized to minimize the loss of the generated image (a) and the rendered image. The generated image (b) and (c) are used to prove that our method can yield diverse results even with a small material graph dataset.