MatEval: Evaluating Indoor 3D Scene Material Recovery from a Single Image

Dongchen Yang
Simon Fraser University
dya78@sfu.ca

Manolis Savva
Simon Fraser University
msavva@sfu.ca

***Abstract*—Converting a single image to a 3D scene with geometry, materials, and lighting is a challenging problem. While geometry reconstruction from a single view has been extensively studied, material recovery for the single-photo-to-scene task remains underexplored. Recent advances in differentiable procedural materials, inverse rendering, and texture generation can be potentially applied to this task. However, they have not been systematically evaluated in a benchmark. In this project, we establish a comprehensive benchmark for material recovery in the single-image-to-scene task. We assume ground-truth 3D geometry as input to isolate material estimation from geometric error. We evaluate three families of methods inspired by recent state-of-the-art approaches in inverse rendering, texture generation, and single-image-to-scene. Our results show that single-view inverse rendering baselines outperform procedural material baselines (19.40 vs 13.60 in PSNR for albedo on original views), highlighting the strong potential of methods based on single-view inverse rendering for material recovery in the single-image-to-scene task. We will release the full dataset, evaluation code, and baseline implementations to support future work.**

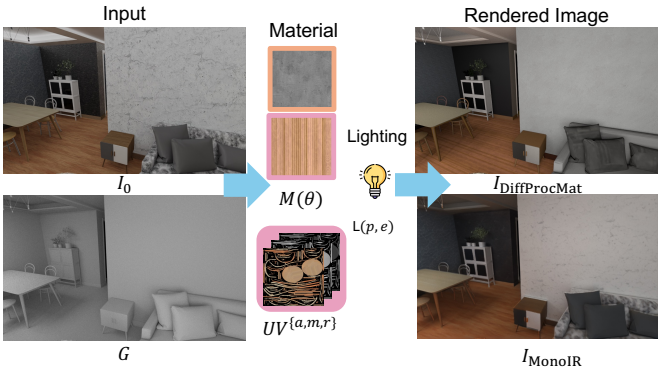

Fig. 1. **Overview of our problem statement**. The input is a single-view RGB image I_0 and the geometry of the scene G. To focus on material recovery, we use ground-truth geometry, factoring out geometric errors. The output is the reconstructed lighting $L(p, e)$ and the recovered procedural materials $M(\theta)$ or image based material $\text{UV}^{\{a,m,r\}}$ of the scene, which we render to images $I_{\text{DiffProcMat}}$ and I_{MonoIR} respectively.

I. Introduction

Converting a single-view image of an indoor scene into a 3D digital twin is a key problem in digital content creation. Such digital twins contain physical properties for the scenes, including geometry, materials, and illumination. While recovering these properties from multi-view images has been widely studied [1–6], single-view images of indoor scenes are far more ubiquitous in practice. Understanding how well physical properties can be recovered from a single image is therefore an important problem. Recovering materials and other physical properties from a single image has broad applications in virtual reality, robotics, and 3D content creation.

Automatically generating such digital content from a single-view image is a challenging and fundamental problem in computer vision. Significant progress has been made on the geometry side, including single-view geometry reconstruction [7], and layout estimation with retrieved objects [8, 9]. For single-view material recovery, several works from Li et al. [10], Zhu et al. [11], Kocsis et al. [12] estimate visible material properties in image space. However, material recovery for indoor scenes in 3D from a single image has been far less explored, despite being critical for achieving reconstructions with high visual fidelity. See Figure 1 for an overview of the material recovery task.

With the progress in differentiable rendering [13], Yeh et al. [14] and Yan et al. [15] addressed the material recovery problem for 3D indoor scenes under the single-view setting. However, their experiments assume imperfect or misaligned geometry, which makes it difficult to disentangle errors in geometry from errors in material recovery. Our benchmark factors out the effect of geometry to focus on material recovery. Moreover, prior works [14, 15] conducted evaluations on a limited set of data. They mostly focused on evaluating the rendered image on the input camera view and lacked evaluations on novel views and materials. In contrast, our benchmark provides larger-scale qualitative and quantitative evaluations on both observed and novel views. In addition, we explored the potential of the single-view inverse rendering approaches. See Table I for a comparison to prior work.

In summary, our contributions are: 1) We build a benchmark to evaluate single-image material recovery from observed and novel views at a larger scale than prior work. 2) We systematically evaluate recent procedural material, single-view material estimation, and texture generation approaches on the material recovery in the single-image-to-scene task. To ensure fair comparison, we focus exclusively on material recovery and factor out geometric errors by using ground-truth geometry. 3) Our experiments show that simple combinations of single-view inverse rendering with texture generation are surprisingly effective, outperforming procedural material approaches.

II. Related Work

Our benchmark focuses on material recovery for the single-image to 3D scene task. We review related work on material modeling for this task, single-view inverse rendering, texture generation, and datasets and benchmarks for inverse rendering.

Material modeling from a single image. Modeling materials from an image is a long-standing problem, especially for indoor scenes with significant occlusions [16]. IM2CAD [17] naïvely assigned the median pixel color as the material color for each object. DiffMat [13, 18] enabled gradient-based optimization for procedural material graphs. Inspired by DiffMat, PhotoScene [14] and PSDR-Room [15] retrieved and optimized differentiable procedural materials [13, 19]. These methods are limited by the size of their material database and the sensitivity of procedural material optimization to local minima. In our benchmark, we compare these methods to learning-based single-view inverse rendering approaches.

Single-view inverse rendering. Inverse rendering aims to recover scene properties such as materials, geometry, and lighting from images. This is a challenging task due to its ill-posed nature [20]. Recent progress in deep learning enabled learning methods to estimate these properties from large-scale synthetic datasets [21, 22]. The priors in diffusion generative models have been used to estimate materials for real images [12, 23]. Generally, these single-view methods recover materials only in image space. Their potential for modeling material properties in 3D scenes has not been studied in a focused manner. Chen et al. [1] lifted 2D estimates to 3D, but did so in a multi-view setting instead of a single-view setting, which is our focus. In our work, we explore the potential of single-view inverse rendering approaches by benchmarking them for indoor 3D scene material recovery from a single image.

Texture generation for 3D objects. With the emergence of 2D diffusion models, DreamFusion [24] and its follow-ups [25, 26] used score distillation sampling (SDS) losses to generate neural radiance fields or textured meshes. DreamMat [27] and TextureDreamer [28] generated 3D assets with BRDF materials. However, these SDS methods are time-consuming because they require test-time optimization. Instead of distilling pretrained 2D diffusion priors to 3D, TEXGen [29] directly generated UV atlas textures and achieved significantly faster inference speed. Because TEXGen [29] is trained to inpaint random missing regions in UV space, it has the potential to be applied to our indoor setting, where occlusions and partial observations are common. Recent works [27, 28, 30] have also explored generating PBR materials. Among these, TextureDreamer [28] requires multi-view input, whereas DreamMat [27] and Material Anything [30] are text-driven. SceneTEX [31] and RoomTEX [32] generate textures for indoor scenes directly, but they are also text-driven. Thus, we study combinations of single-view inverse rendering methods with TEXGen [29] to inpaint materials for unobserved regions.

Inverse rendering datasets and benchmarks. OpenIllumination [33] introduced a real dataset for objects under controlled illumination. Stanford-Orb [34] extended the setting to in-the-wild illumination. However, these datasets focused on object-centric inverse rendering instead of scenes. Inverse rendering for indoor scenes is more challenging with occlusions between objects and more complex lighting. Li et al. [21] proposed OpenRooms, a large-scale indoor scene dataset with ground-truth materials, geometry, and lighting. However, the OpenRooms scenes are sparsely populated and the objects lack diversity (9.4 objects and 4.1 categories per scene) since they originate from Scan2CAD [35] alignments of ShapeNet [36] objects to scans. Roberts et al. [37] introduced Hypersim, and Zhu et al. [22] introduced InteriorVerse, both providing higher photorealism. However, the 3D assets in Hypersim must be purchased separately and the 3D assets in InteriorVerse are not publicly available. 3D-FRONT [38] is an open-sourced indoor 3D scene dataset, containing higher modeling quality objects from 3D-FUTURE [39] compared to OpenRoom objects. Therefore, we leverage 3D-FRONT to create our material recovery benchmark. In addition, we leverage 4 high-quality artist-designed open-sourced indoor scene models from Bitterli's rendering resources [40]. PhotoScene [14] and PSDR-Room [15] only evaluated on small datasets and reported primarily qualitative results only on original camera viewpoints. In contrast, we benchmark these baselines on a much larger dataset, both qualitatively and quantitatively, and on both original and novel views.

TABLE I

In previous work, PSDR-Room [15] and PhotoScene[14], only a limited number of quantitative evaluations are included for the original camera view. In addition, there is no quantitative evaluation of novel views for previous work.

	PSDR-Room	**PhotoScene**	**MatEval (ours)**
Number of scenes	7	70	184
Avg. Objects/Scene	/	9.4	13.2
Avg. Categories/Scene	/	4.1	4. 3
Original view eval.	✓	✓	✓
Novel view eval.	✗	✗	✓
Albedo eval.	✗	✗	✓

III. Problem Statement

Figure 1 provides an overview of material recovery for the single-image-to-scene task. Given a single-view RGB image I_0 as input, we focus on the material estimation part of the problem. To isolate the problem of material estimation from other factors such as errors in geometry and camera alignment, we use ground-truth 3D geometry G and camera pose. The output consists of the lighting $L(p, e)$ and the material for the scene, where p and e parameterize the position and the emission for the emitters. The material parameterization depends on whether materials are defined by **diff**erentiable **proc**edural **mat**erial (DiffProcMat) or estimated by **mono**cular **i**nverse **r**endering methods (MonoIR). The former is parameterized as $M(\theta)$, where θ is the optimizable parameters for the procedural materials M, such as rotation and scale, and the latter as $\{a, m, r\}$ including albedo a, roughness r, and metallic m in UV space. We assume the ground-truth part

Fig. 2. **Comparison between MatEval (ours) and OpenRooms [21].** MatEval includes scenes curated from the 3D-FRONT [38] (left) and Bitterli [40] (middle) datasets. Compared with OpenRooms [21] (right), our dataset features furniture with more diverse and photorealistic material textures, leading to richer visual variety across indoor scenes.

segmentation is provided for architectures such as walls and floors. For furniture objects (e.g., beds and sofas), we assume ground-truth instance-level segmentation.

IV. Dataset

A dataset for evaluating material recovery should contain ground-truth geometry, images with corresponding camera poses, and ground-truth material properties. Ideally, each scene should be photorealistic with a diverse set of objects. Part-level segmentation for both objects and architectures in the scene is also useful, as some baselines retrieve and optimize materials based on parts. A potential choice to source 3D scenes from is OpenRooms [21]. However, OpenRooms is based on Scan2CAD [35], which aligns ShapeNet [36] objects to scans. Thus, the object geometry and materials are not carefully designed by professionals, resulting in renderings that may lack photorealism (see Figure 2). Hypersim [37] provides a collection of high-quality indoor scenes renderings. While providing photorealictic images, the 3D scene models underlying the renderings must be purchased separately. This makes it less suitable for a open-sourced benchmark with geometry and material annotation. 3D-FRONT [38] is a large-scale synthetic indoor scene dataset, which is an alternative choice. It covers a variety of indoor scenes and is designed by professionals. Furniture objects in 3D-FRONT are from 3D-FUTURE [39], which is also authored by professional artists with ground-truth material information.

Our dataset is based on 3D-FRONT and consists of 180 scenes. Each scene contains one original camera view and four novel views generated by rotating the original camera about its optical center without translation by five degrees up-right, up-left, down-right, and down-left. Although the unobserved regions appear as thin strips at the image borders, completing these regions is still a challenging task based on our results in section VI. For each view, we render the RGB image and diffuse albedo, and the associated part-level segmentation for architectural elements, such as walls and floors. We export the 3D models at the instance-segmented level for furniture objects and the part-segmented level for architectures. For baselines that require part segmentations, we additionally export the 3D models for the furniture objects at the part-segmented level. Compared with OpenRooms [21], our dataset is more photo-realistic due to the carefully designed 3D-FRONT scenes and 3D-FUTURE furniture objects. See Figure 2 for a comparison of renderings. On average, each room scene in our dataset has 13.2 objects and 4.3 object categories (compare to 9.4 objects and 4.1 for OpenRooms). Overall, our scenes are more densely populated. Note that the categorization scheme we adopt is coarser (8 categories, including "other") relative to OpenRooms which uses the ShapeNet 50 categories set.

Other than 3D-FRONT, we included four scenes from the Bitterli [40] open-sourced online 3D scene dataset with significantly higher quality and object density. See Figure 2 for examples. Unlike 3D-FRONT, the Bitterli dataset contains mirror regions, complementing the material diversity of our benchmark. As with 3D-FRONT, we render four novel views for each scene using the same five-degree rotation strategy.

V. Evaluated Methods

Given a single-view image and the corresponding indoor scene geometry, we aim to recover the material properties along with lighting under the assumption from Section III. The closest related works are PSDR-Room [15] and PhotoScene [14], which optimized procedural materials to recover material properties. However, these works used retrieved geometry and did not factor out the geometric errors. To establish baselines, we modify PSDR-Room to fit our setting by replacing the geometry pipeline with ground-truth geometry with part segmentation. In addition, we explore the potential of single-view inverse rendering and texture generation. In summary, we evaluate three families of baselines: 1) **DiffProcMat:** differentiable procedural materials (modified PSDR-Room); 2) **MonoIR + NN:** single-view inverse rendering methods with unobserved regions filled via nearest neighbor lookup; and 3) **MonoIR + TEXGen:** single-view inverse rendering methods with unobserved regions filled via TEXGen [29]. The MonoIR baselines use ColorfulShading [41], RGBX [23], and Marigold [42]. Since the full implementation of PSDR-Room [15] is not available, we reimplemented it based on the description in the paper and the provided partial implementation. Because the geometry assumption in PSDR-Room is retrieved geometry, using ground-truth part-level segmentation for furniture is fair. However, part-level segmentation for furniture objects is not needed for MonoIR-based approaches. To ensure fairness, we include another version of PSDR-Room, where the part-segmentation for furniture objects is computed via PartField [43].

Overall, the baselines are implemented under a three-stage framework: 1) material stage, 2) light stage, and 3) optimization stage. We describe the baseline implementation below.

A. Material Stage

DiffProcMat methods retrieve per-part materials from a database. MonoIR methods estimate materials in camera space

and project them onto 3D geometry. Since monocular inverse rendering methods only estimate the materials for visible regions, we fill the unobserved regions either by nearest neighbor lookup or texture generation

DiffProcMat. We generally follow PSDR-Room [15]. See original paper for more details. For the material database, we use the 118 procedural materials released with PSDR-Room. We first render the material part masks of the original camera view based on the segmented meshes. We then extract square crops from these masks. Each cropped part is compared with thumbnail renderings of all procedural materials in the database. The closest match is selected via k-nearest neighbor lookup in the CLIP embedding space. If the CLIP similarity score is lower than 0.25 or if the size of the cropped square is smaller than a threshold (100×100 pixels), we consider the match unreliable and assign a homogeneous material whose base color is the median pixel color from the input image.

MonoIR + NN. To apply the 2D estimated materials to 3D scenes, we use the nearest neighbor lookup as a naïve baseline (see Figure 3). Based on the depth map D of the input image, we first project the estimated material properties $\{a, m, r\}$ from camera space to a partial point cloud $p_{\text{partial}}^{\{a,m,r\}}$ of the scene. For each object in the scene, we sample an uncolored point cloud from the ground-truth 3D models in Section IV. We then assign material attributes of the partial point cloud to the uncolored point cloud based on nearest neighbor lookup, which fills the unobserved regions with the nearest material attributes. Finally, we convert the resulting point cloud with material properties $p_{\text{nn}}^{\{a,m,r\}}$ to a UV atlas $\text{UV}_{\text{nn}}^{\{a,m,r\}}$ for each object using the rasterization-based approach of Yu et al. [44]. We apply the estimated materials from recent monocular inverse rendering methods, including ColorfulShading [41], RGBX [23], and Marigold [42]. To provide an upper bound, we also apply the ground-truth albedo. For material properties not estimated or not provided in the ground truth, we use default values from Mitsuba3 [45].

MonoIR + TEXGen. To explore texture generation, we lift the 2D-estimated material to 3D via TEXGen [29] to handle the unobserved regions (see Figure 3). Because TEXGen only works on albedo image, we only lift the albedo property to 3D via TEXGen. For other material properties, we use the default values from Mitsuba3 [45] or lift via nearest neighbor lookup. Because TEXGen [29] is object-centric, we process objects separately. Given the 3D model G_i, based on the rasterization approach from TEXGen [29], we can get a position map containing the 3D position for each face of the mesh in UV space, and a mask map $\text{UV}_{\text{full}}^{\text{M}_i}$ denoting the valid texture regions. Using the 3D model G_i, input camera pose, the estimated albedo $\{a\}$, and object mask M_i, based on the rasterization approach from TEXGen [29], we obtain the partial UV-atlas $\text{UV}_{\text{partial}}^{\{a\}}$ (visible textures), and the corresponding mask map $\text{UV}_{\text{partial}}^{M_i}$ (visible regions in UV space). With the position map, $\text{UV}_{\text{full}}^{\text{M}_i}$, $\text{UV}_{\text{partial}}^{\{a\}}$, $\text{UV}_{\text{partial}}^{M_i}$, the masked albedo image $a{*}M_i$, and a text prompt for the object, we use TEXGen [29] to inpaint the unobserved regions to get $\text{UV}_{\text{texgen}}^{\{a\}}$.

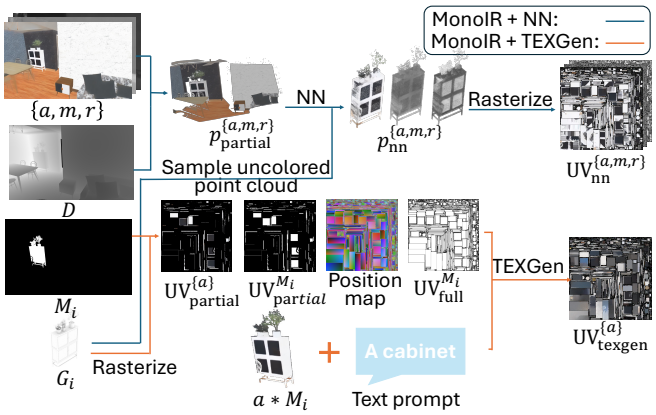

Fig. 3. **Overview of pipeline to lift 2D materials to 3D.** We lift 2D estimations $\{a, m, r\}$ to 3D in UV space for each object i. For the nearest neighbor approach: (1) map estimated 2D material properties $\{a, m, r\}$ to a point cloud $p_{\text{partial}}^{\{a,m,r\}}$ and fill unobserved regions with nearest neighbor lookup to get $p_{\text{nn}}^{\{a,m,r\}}$, and (2) project $p_{\text{nn}}^{\{a,m,r\}}$ to UV space $\text{UV}_{\text{nn}}^{\{a,m,r\}}$. For the texture generation approach: (1) map the albedo and object mask from camera space to UV space $\text{UV}_{\text{partial}}^{\{a\}}$ and $\text{UV}_{\text{partial}}^{M_i}$, (2)obtain the position map and mask map $\text{UV}_{\text{full}}^{M_i}$ from the 3D model of the object G_i (2) apply TEXGen [29] to fill unobserved regions and obtain $\text{UV}_{\text{texgen}}^{\{a\}}$.

B. Lighting Stage

For 3D-FRONT, we partially follow PSDR-Room's [15] light placement strategy. PSDR-Room (i) divides the ceiling into a grid of area lights and removes those that intersect the camera frustum or fall behind the camera, (ii) adds one large area light behind the camera and one per unobserved wall, and (iii) turns visible windows and lamps into emissive objects. We adopt (i) and (ii) but skip (iii). We deviate here because 3D-FRONT images rendered by BlenderProc [46] treat the entire ceiling and lamp fixture as emissive and filter the direct-lighting out of the final image to avoid overbright surfaces. Modelling these visible surfaces as explicit area lights in our pipeline would introduce that direct term and over-brighten ceilings/lamps. For the Bitterli dataset [40], the light source placement is well-calibrated, so we use it directly.

After initializing the light placement p, we assign default material properties for the entire scene: diffuse albedo [0.5,0.5,0.5], roughness 0.5, and render from the input camera pose by setting all light sources to a uniform radiance value e from [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0]. Then we compare the luminance difference between the input image and the rendered image under each radiance value. The radiance with the lowest mean squared error is used to initialize the lights.

C. Optimization Stage

DiffProcMat. The material stage for DiffProcMat selects a procedural material for each part but does not initialize its rotation and scale parameters. Following PSDR-Room [15], we initialize these parameters by rendering the images under different rotation angles $r \in [-45, 0, 45, 90]$ and scaling factors $s \in (0.5, 8.0)$ for each procedural material part under the initialized lighting. We compute a Gram matrix loss $\mathcal{L}_1(T_G(R_i(r, s)), T_G(I_i))$ for each material part i, where T_G

denotes the Gram matrix of VGG features [47]. For image I, the Gram Matrix is $T_G(I) = \text{VGG}(I)\text{VGG}(I)^T$.

Consistent with PSDR-Room, we also render another image assuming homogeneous material: 1) the base color for each part is set to its median pixel color from the input image; 2) default values are applied to other material properties. If the Gram Matrix loss under the homogeneous assumption is lower than the best previous loss for procedural material, we will replace the procedural material with a homogeneous material.

With scene geometry, initial lighting, and initial material, we jointly optimize the light radiance, the homogeneous material color, and the procedural material parameters, including rotation, scale, and color. The overall loss is combined with several terms: 1) a downsampled L1 loss between the rendered and input images $\mathcal{L}_1(R_{1/8(\theta)}, I_{1/8})$, 2) a Gram matrix Loss for each cropped region $\sum_i (T_G(R_i(\theta)), T_G(I_i))$, 3) and a mean RGB loss over each object mask $\sum_i \mathcal{L}_1 p(\mu(R_i(\theta)), \mu(I_i))$. The complete loss is $\mathcal{L} = \lambda_1 \mathcal{L}_1(R_{1/8}, I_{1/8}) + \lambda_2 \sum_i (T_G(R_i(\theta)), T_G(I_i)) + \lambda_3 \sum_i \mathcal{L}_1(\mu(R_i(\theta)), \mu(I_i))$, where θ represents material and light radiance parameters, $R_{1/8}$ and $I_{1/8}$ are images downsampled by $1/8$. We use the same λ_i values in PSDR-Room. We optimize using the Adam [48] optimizer.

MonoIR. The optimization stage for MonoIR-based pipelines optimizes the radiance for the emitters to have the final renderings match with the input image We optimize the radiance of each emitter to align the rendered image with the input image. The objective is an L1 loss: $\mathcal{L}_1(\text{R}, \text{I})$, optimized with the Adam [48] optimizer.

VI. Experiments

A. Evaluation Protocol

We render the RGB images based on the recovered material and lighting properties. We also render albedo images for direct evaluation of the estimated albedo. For PSDR-Room, we use PSDR-Jit [49] for the final renderings. For MonoIR baselines, we obtain the final renderings with Mitsuba 3 [45]. Some inverse rendering methods do not estimate roughness or metallic values [41]. We set default values of 0.5 for roughness and 0 for metallic. We compare the rendered images with the input images qualitatively and quantitatively. Additionally, we render and compare novel view images to evaluate material recovery on unseen regions. Regions containing information outside of the current optimized indoor scene, such as windows, are masked out during quantitative evaluation. For albedo evaluation, we follow prior works [2, 3, 10] and compute a scale-invariant error, where we aligned the estimated albedo to ground truth by a channel-wise scaling factor.

B. Evaluation Metrics

We use a suite of metrics: 1) Mean Squared Error (MSE), 2) Peak Signal-to-Noise Ratio (PSNR), 3) Structural Similarity Index Measure (SSIM), and 4) Learned Perceptual Image Patch Similarity (LPIPS). MSE and PSNR measure per-pixel differences between the original input image and the rendered image. SSIM evaluates patch-level structural similarity. LPIPS measures the similarity in a learned feature space.

C. Analysis

We compare PSDR-Room [15] with our MonoIR-based approaches. PSDR-Room needs part-level segmentation while MonoIR methods do not. To ensure fairness, we include an additional baseline PSDR-Room* that uses PartField [43] to compute part segmentation for furniture objects. We summarize our main observations below.

MonoIR methods outperform DiffProcMat. As shown in Tables II and III, MonoIR-based methods outperform DiffProcMat across most quantitative metrics. Qualitative results in fig. 4 show that MonoIR-based methods (last 3 columns) can recover textures more accurately compared with DiffProcMat methods (2nd and 3rd column), such as patterns on the tables and bed and pictures on the wall. This improvement is because MonoIR-based approaches are not constrained by a fixed material database, whereas DiffProcMat methods are. In addition, MonoIR-based methods avoid the process of optimizing various parameters of procedural materials. This optimization process can get stuck at local minima if the parameters for procedural materials are not initialized well. For example, the direction for the stripe patterns on the bed for PSDR-Room (2nd and 3rd column) is orthogonal to the ground truth. The recent generative models for the single-view inverse rendering task achieve strong material estimation [12, 23, 42], but can introduce unintended content changes in the image. For example, in the third scene in Figure 4, RGBX incorrectly estimates the albedo of the wall picture as black, since RGBX is a generative model that may alter content from the input image. While using estimated albedo improves performance, we observe that estimating roughness and metallic channels remains challenging, because using the additional estimated roughness and metallic channels does not bring any improvement. The 3D-FRONT scenes we rendered do not contain many highly specular regions, so using the default value 0.5 for roughness and 0 for metallic can already generate reasonable results. Overall, using the ground-truth albedo with default parameters for other material channels still performs best across most of the metrics, which suggests room for improvement for single-view material estimation.

Is texture generation helpful for completing invisible regions? The quantitative results for novel view RGB in Table II and albedo renderings Table IV do not clearly show texture generation provides measurable benefits. We think this could be related to the novel view images still containing a large portion of regions that were previously observed. Therefore, we include an additional evaluation restricted to unobserved regions. The evaluations on albedo for unobserved regions in Table V show that the methods incorporating TEXGen outperform other methods across most metrics. Figure 5 further shows that TEXGen actually helps with generating textures for those unobserved regions.

Texture generation failures. While TEXGen [29] generally produces reasonable textures in unobserved regions, one

TABLE II

QUANTITATIVE COMPARISON ON ORIGINAL AND NOVEL CAMERA VIEWS FOR RGB RENDERINGS ON OUR 3D-FRONT SCENES. "PSDR-ROOM*" MEANS THE PSDR-ROOM BASELINE IS USING MESH SEGMENTATIONS BY PARTFIELD [43]. SUPERSCRIPTS WITH LETTERS INDICATE WHICH ESTIMATED MATERIAL CHANNEL IS USED FROM MONOIR METHODS. **BOLD** IS FOR THE BEST PERFORMANCE AND UNDERSCORE IS FOR THE SECOND BEST PERFORMANCE. AS SHOWN IN THIS TABLE, USING ESTIMATED ALBEDO GENERALLY OUTPERFORMS PROCEDURAL MATERIAL OPTIMIZATION, WHILE ADDING THE ESTIMATED METALLIC CHANNEL DOES NOT CONSISTENTLY IMPROVE RENDERING QUALITY.

Method	Original View RGB				Novel View RGB			
	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DiffProcMat								
PSDR-Room	7.235	22.43	0.6381	0.4068	8.653	21.56	0.6365	0.4097
PSDR-Room*	7.285	22.15	0.6368	0.4138	8.888	21.27	0.6350	0.4168
MonoIR + NN								
GT[a]	6.946	**23.33**	**0.7977**	0.2568	8.711	21.91	**0.7787**	0.2781
ColorfulShading[a]	6.856	22.80	0.7738	0.2956	8.746	21.55	0.7562	0.3158
RGBX[a]	7.425	22.81	0.7541	0.2626	9.347	21.49	0.7393	0.2862
Marigold[a]	6.446	22.78	0.7185	0.3054	8.283	21.59	0.7121	0.3235
RGBX[a,r]	7.403	22.81	0.7532	0.2642	9.333	21.49	0.7383	0.2876
Marigold[a,r]	**6.356**	22.83	0.7190	0.3044	**8.187**	21.63	0.7125	0.3226
RGBX[a,m,r]	8.468	22.06	0.7232	0.3072	10.391	20.87	0.7106	0.3272
Marigold[a,m,r]	6.448	22.79	0.7211	0.3042	8.309	21.56	0.7146	0.3222
MonoIR + TEXGen								
GT[a]	6.624	23.04	0.7771	**0.2489**	8.203	**21.92**	0.7524	**0.2692**
ColorfulShading[a]	8.098	22.08	0.7476	0.2737	9.714	21.13	0.7267	0.2929
RGBX[a]	7.803	22.36	0.7356	0.2509	9.507	21.32	0.7142	0.2738
Marigold[a]	7.642	22.11	0.6995	0.2806	9.330	21.14	0.6866	0.3009
RGBX[a,r]	7.788	22.37	0.7346	0.2523	9.499	21.32	0.7139	0.2741
Marigold[a,r]	7.656	22.10	0.6997	0.2806	9.342	21.13	0.6862	0.3018
RGBX[a,m,r]	8.557	21.73	0.7078	0.2917	10.294	20.79	0.6892	0.3092
Marigold[a,m,r]	7.525	22.11	0.6990	0.2877	9.212	21.14	0.6876	0.3056

TABLE III

QUANTITATIVE COMPARISON ON ORIGINAL CAMERA VIEWS FOR ALBEDO RENDERINGS ON 3D-FRONT SCENES. "PSDR-ROOM*" MEANS THE PSDR-ROOM BASELINE IS USING MESH SEGMENTATIONS BY PARTFIELD [43]. SUPERSCRIPTS WITH LETTERS INDICATE WHICH ESTIMATED MATERIAL CHANNEL IS USED FROM MONOIR METHODS. **BOLD** IS FOR THE BEST PERFORMANCE AND UNDERSCORE IS FOR THE SECOND BEST PERFORMANCE. THESE DIRECT EVALUATIONS ON ALBEDO SHOW THAT APPLYING ESTIMATED MATERIALS INSTEAD OF OPTIMIZING PROCEDURAL MATERIALS PRODUCES BETTER RESULTS IN MATERIAL RECOVERY.

Method	Albedo				Aligned Albedo			
	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DiffProcMat								
PSDR-Room	46.911	13.60	0.6277	0.4303	20.290	17.59	0.6602	0.3968
PSDR-Room*	49.023	13.41	0.6213	0.4386	21.533	17.28	0.6543	0.4045
MonoIR + NN								
GT[a]	**0.848**	**32.98**	**0.9225**	**0.1051**	**0.809**	**33.10**	**0.9227**	**0.1051**
ColorfulShading[a]	19.555	17.54	0.7873	0.3130	8.474	21.30	0.7965	0.2925
RGBX[a]	16.809	19.40	0.7621	0.2487	9.926	21.64	0.7693	0.2428
Marigold[a]	28.279	15.84	0.6993	0.3220	7.855	21.63	0.7212	0.2966
MonoIR + TEXGen								
GT[a]	1.399	29.07	0.8693	0.1348	1.253	29.73	0.8683	0.1349
ColorfulShading[a]	17.573	18.01	0.7219	0.2936	8.762	21.06	0.7398	0.2765
RGBX[a]	15.721	19.71	0.6998	0.2700	10.480	21.20	0.7107	0.2637
Marigold[a]	24.599	16.45	0.6377	0.3091	8.651	21.08	0.6677	0.2863

Fig. 4. **Qualitative comparisons on the original camera view on 3D-FRONT scenes.** "PSDR-Room*" indicates the PSDR-Room baseline using meshes segmented by PartField [43]. Superscripts denote which material channels are estimated by MonoIR methods. MonoIR methods (last 3 columns) recover diverse material textures and avoid the local minima issue seen in DiffProcMat (2nd and 3rd columns).

TABLE IV

QUANTITATIVE COMPARISON ON NOVEL CAMERA VIEWS FOR ALBEDO RENDERINGS ON OUR 3D-FRONT SCENES. "PSDR-ROOM*" MEANS THE PSDR-ROOM BASELINE IS USING MESH SEGMENTATIONS BY PARTFIELD [43]. THE SUPERSCRIPTS WITH LETTERS INDICATE WHICH ESTIMATED MATERIAL CHANNEL IS USED FROM MONOIR METHODS. **BOLD** IS FOR THE BEST PERFORMANCE AND UNDERSCORE IS FOR THE SECOND BEST PERFORMANCE. AS SHOWN IN THE TABLE, THE NOVEL VIEW RENDERINGS WITH TEXGEN LEAD TO OVERALL BETTER RESULTS ACROSS MOST METRICS.

Method	Albedo				Aligned Albedo			
	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DiffProcMat								
PSDR-Room	48.003	13.50	0.6272	0.4287	22.346	17.12	0.6593	0.3966
PSDR-Room*	50.233	13.30	0.6208	0.4374	23.895	16.78	0.6538	0.4049
MonoIR + NN								
GT[a]	<u>3.620</u>	**26.14**	**0.8860**	**0.1468**	<u>3.556</u>	**26.19**	**0.8857**	**0.1468**
ColorfulShading[a]	21.679	17.07	0.7702	0.3363	10.745	20.26	0.7790	0.3168
RGBX[a]	19.009	18.63	0.7483	0.2769	12.173	20.44	0.7545	0.2710
Marigold[a]	30.491	15.50	0.6946	0.3447	10.079	20.57	0.7153	0.3199
MonoIR + TEXGen								
GT[a]	**3.183**	<u>25.65</u>	<u>0.8265</u>	<u>0.1687</u>	**2.998**	<u>26.02</u>	<u>0.8247</u>	<u>0.1689</u>
ColorfulShading[a]	18.367	17.79	0.6995	0.3117	10.230	20.34	0.7170	0.2962
RGBX[a]	16.789	19.28	0.6941	0.2758	11.494	20.65	0.7044	0.2697
Marigold[a]	25.765	16.24	0.6402	0.3148	10.010	20.47	0.6693	0.2930

TABLE V

QUANTITATIVE COMPARISON ON ALIGNED ALBEDO FOR UNOBSERVED REGIONS ON 3D-FRONT SCENES. "PSDR-ROOM*" MEANS THE PSDR-ROOM BASELINE IS USING MESH SEGMENTATIONS BY PARTFIELD [43]. SUPERSCRIPTS WITH LETTERS INDICATE WHICH ESTIMATED MATERIAL CHANNEL IS USED FROM MONOIR METHODS. **BOLD** IS BEST AND UNDERSCORE IS 2ND BEST PERFORMANCE. THE DIRECT EVALUATION ON ALBEDO SHOWS APPLYING TEXGEN PRODUCES BETTER RESULTS IN MATERIAL RECOVERY FOR UNOBSERVED REGIONS.

Method	MSE (10^{-3}) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DiffProcMat				
PSDR-Room	33.335	16.15	0.9396	0.0679
PSDR-Room*	35.815	15.76	0.9390	0.0696
MonoIR + NN				
GT[a]	17.417	19.78	**0.9529**	0.0433
ColorfulShading[a]	22.999	17.62	0.9482	0.0592
RGBX[a]	24.673	17.58	0.9458	0.0560
Marigold[a]	22.238	17.92	0.9473	0.0614
MonoIR + TEXGen				
GT[a]	**11.909**	**20.49**	0.9359	**0.0418**
ColorfulShading[a]	18.660	18.18	0.9326	0.0551
RGBX[a]	20.142	18.22	0.9292	0.0530
Marigold[a]	20.065	17.96	0.9313	0.0578

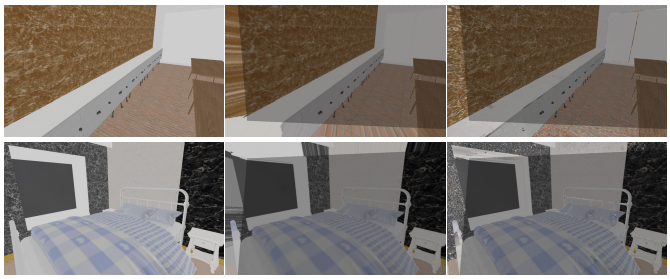

Fig. 5. **Qualitative comparison of aligned albedo for novel view renderings on 3D-FRONT scenes.** Darkened regions are previously observed regions. TEXGen generates reasonable textures for unobserved regions.

observation is that TEXGen can fail on some simple surfaces like walls, and result in some artifacts, such as the ceiling parts (see fig. 6). These artifacts are likely related to the lack of observations. The ceilings are usually located at the corner part of the original image, and only a small portion of the entire ceiling is observed. This suggests room for improvement for texture generation with very limited observations. Another possible reason is that TEXGen is an object-centric model for texture generation, and the architectures of walls and ceilings are out-of-distribution cases.

Failures on mirror regions. Mirrors are frequently present within indoor scenes, but all baselines struggle to recover reflective materials in the Bitterli dataset (see fig. 7). DiffProcMat tries to retrieve the material from a database based on the CLIP features of the cropped image patches. If the mirror reflects other scene contents, the retrieved material will be incorrect. In addition, the optimization process will be confused by the contents reflected by the mirror. For MonoIR baselines, some do not model the mirror effects and only recover a diffuse albedo and a shading [41]. They may also have the content reflected by the mirror baked in the

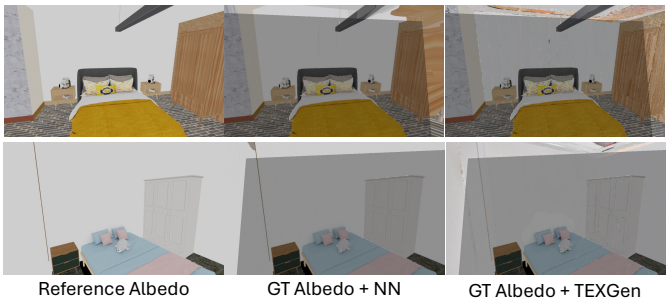

Fig. 6. **TEXGen noise issues.** The darkened regions are previously observed regions. TEXGen generates noise in featureless regions (e.g., ceilings).

Fig. 7. **Mirror region failures.** All baselines struggle with mirrors. In this scene from the Bitterli dataset, for the mirror on the closet, DiffProcMat assigns a median color and MonoIR produces a "baked in" appearance.

recovered albedo, which will result in unrealistic renderings. This suggests a promising direction for future research in material recovery for highly specular surfaces.

VII. LIMITATIONS

We used TEXGen [29] for texture generation, but it is limited to albedo inpainting. Future work can extend TEXGen or develop alternative methods to generate other intrinsic channels. Our novel views are five-degree camera-center rotations, which we already find challenging to complete in the unobserved image-border regions. Larger viewpoint changes that reveal more occluded surfaces would be a stronger test, and we leave this to future work. Our benchmark assumes ground-truth geometry by design to isolate material recovery from geometric error. Measuring how each baseline performs under imperfect geometry such as retrieved or reconstructed geometry is a complementary study for future work. Our benchmark uses renderings from 3D-FRONT [38] and Bitterli dataset [40]. These synthetic datasets exhibit a domain gap to real scenes. Extending our benchmark with real scene reconstructions and annotated ground-truth geometry and materials is another promising direction for future work.

VIII. CONCLUSION

We present a benchmark for material recovery in the single-image-to-scene task. We systematically evaluated three families of methods on this task, exploring the potential of differentiable procedural materials, and recent advances in single-view inverse rendering and texture generation. We found that simple combinations of single-view inverse rendering with texture generation outperform recent procedural material methods. Despite these advances, substantial space for improvement remains, highlighting the potential for continued work on material recovery in the single-image-to-scene task.

References

[1] X. Chen, S. Peng, D. Yang, Y. Liu, B. Pan, C. Lv, and X. Zhou, "Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination," in *European Conference on Computer Vision*. Springer, 2024, pp. 450–467.

[2] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 5453–5462.

[3] X. Zhang, P. P. Srinivasan, B. Deng, P. E. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Transactions on Graphics (TOG)*, vol. 40, pp. 1 – 18, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235313848

[4] L. Wu, R. Zhu, M. B. Yaldiz, Y. Zhu, H. Cai, J. Matai, F. Porikli, T.-M. Li, M. Chandraker, and R. Ramamoorthi, "Factorized inverse path tracing for efficient and accurate material-lighting estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3848–3858.

[5] J. Zhu, Y. Huo, Q. Ye, F. Luan, J. Li, D. Xi, L. Wang, R. Tang, W. Hua, H. Bao, and R. Wang, "I^2-sdf: Intrinsic indoor scene reconstruction and editing via raytracing in neural sdfs," in *CVPR*, 2023.

[6] Y. Litman, O. Patashnik, K. Deng, A. Agrawal, R. Zawar, F. de la Torre, and S. Tulsiani, "Materialfusion: Enhancing inverse rendering with material diffusion priors," *2025 International Conference on 3D Vision (3DV)*, pp. 802–812, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272826887

[7] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang, "Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 55–64.

[8] D. Gao, D. Rozenberszki, S. Leutenegger, and A. Dai, "Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–15, 2024.

[9] Q. Wu, D. Iliash, D. Ritchie, M. Savva, and A. X. Chang, "Diorama: Unleashing zero-shot single-view 3d scene modeling," *arXiv e-prints*, pp. arXiv–2411, 2024.

[10] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "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*, 2020, pp. 2475–2484.

[11] R. Zhu, Z. Li, J. Matai, F. Porikli, and M. Chandraker, "Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2822–2831.

[12] P. Kocsis, V. Sitzmann, and M. Nießner, "Intrinsic image diffusion for indoor single-view material estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5198–5208.

[13] L. Shi, B. Li, M. Hašan, K. Sunkavalli, T. Boubekeur, R. Mech, and W. Matusik, "Match: Differentiable material graphs for procedural material capture," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.

[14] Y.-Y. Yeh, Z. Li, Y. Hold-Geoffroy, R. Zhu, Z. Xu, M. Hašan, K. Sunkavalli, and M. Chandraker, "Photoscene: Photorealistic material and lighting transfer for indoor scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 562–18 571.

[15] K. Yan, F. Luan, M. Hašan, T. Groueix, V. Deschaintre, and S. Zhao, "Psdr-room: Single photo to scene using differentiable rendering," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–11.

[16] D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross, "Brdf representation and acquisition," in *Computer graphics forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 625–650.

[17] H. Izadinia, Q. Shan, and S. M. Seitz, "Im2cad," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5134–5143.

[18] B. Li, L. Shi, and W. Matusik, "End-to-end procedural material capture with proxy-free mixed-integer optimization," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–15, 2023.

[19] Adobe, "Substance designer," 2025. [Online]. Available: https://www.adobe.com/

[20] G. Patow and X. Pueyo, "A survey of inverse rendering problems," in *Computer graphics forum*, vol. 22, no. 4. Wiley Online Library, 2003, pp. 663–687.

[21] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi *et al.*, "Openrooms: An end-to-end open framework for photorealistic indoor scene datasets," *arXiv preprint arXiv:2007.12868*, 2020.

[22] J. Zhu, F. Luan, Y. Huo, Z. Lin, Z. Zhong, D. Xi, R. Wang, H. Bao, J. Zheng, and R. Tang, "Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing," in *SIGGRAPH Asia 2022 Conference Papers*. ACM, 2022. [Online]. Available: https://doi.org/10.1145/3550469.3555407

[23] Z. Zeng, V. Deschaintre, I. Georgiev, Y. Hold-Geoffroy, Y. Hu, F. Luan, L.-Q. Yan, and M. Hašan, "Rgb $\leftrightarrow$ x: Image decomposition and synthesis using material- and lighting-aware diffusion models," in *ACM SIGGRAPH 2024 Conference Papers*, ser. SIGGRAPH '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available:

https://doi.org/10.1145/3641519.3657445

[24] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=FjNys5c7VyY

[25] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 300–309.

[26] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *Advances in neural information processing systems*, vol. 36, pp. 8406–8441, 2023.

[27] Y. Zhang, Y. Liu, Z. Xie, L. Yang, Z. Liu, M. Yang, R. Zhang, Q. Kou, C. Lin, W. Wang *et al.*, "Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–18, 2024.

[28] Y.-Y. Yeh, J.-B. Huang, C. Kim, L. Xiao, T. Nguyen-Phuoc, N. Khan, C. Zhang, M. Chandraker, C. S. Marshall, Z. Dong *et al.*, "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.

[29] X. Yu, Z. Yuan, Y.-C. Guo, Y.-T. Liu, J. Liu, Y. Li, Y.-P. Cao, D. Liang, and X. Qi, "Texgen: a generative diffusion model for mesh textures," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–14, 2024.

[30] X. Huang, T. Wang, Z. Liu, and Q. Wang, "Material anything: Generating materials for any 3d object via diffusion," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 556–26 565.

[31] D. Z. Chen, H. Li, H.-Y. Lee, S. Tulyakov, and M. Nießner, "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, pp. 21 081–21 091.

[32] Q. Wang, R. Lu, X. Xu, J. Wang, M. Y. Wang, B. Dai, G. Zeng, and D. Xu, "Roomtex: Texturing compositional indoor scenes via iterative inpainting," in *European Conference on Computer Vision*. Springer, 2024, pp. 465–482.

[33] I. Liu, L. Chen, Z. Fu, L. Wu, H. Jin, Z. Li, C. M. R. Wong, Y. Xu, R. Ramamoorthi, Z. Xu *et al.*, "Openillumination: A multi-illumination dataset for inverse rendering evaluation on real objects," *Advances in Neural Information Processing Systems*, vol. 36, pp. 36 951–36 962, 2023.

[34] Z. Kuang, Y. Zhang, H.-X. Yu, S. Agarwala, E. Wu, J. Wu *et al.*, "Stanford-orb: a real-world 3d object inverse rendering benchmark," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 938–46 957, 2023.

[35] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Niessner, "Scan2cad: Learning cad model alignment in rgb-d scans," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[37] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *International Conference on Computer Vision (ICCV) 2021*, 2021.

[38] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, "3d-front: 3d furnished rooms with layouts and semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 933–10 942.

[39] H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. Maybank, and D. Tao, "3d-future: 3d furniture shape with texture," *International Journal of Computer Vision*, vol. 129, no. 12, pp. 3313–3337, 2021.

[40] B. Bitterli, "Rendering resources," 2016, https://benedikt-bitterli.me/resources/.

[41] C. Careaga and Y. Aksoy, "Colorful diffuse intrinsic image decomposition in the wild," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–12, 2024.

[42] B. Ke, K. Qu, T. Wang, N. Metzger, S. Huang, B. Li, and A. Obukhov, "Marigold: Affordable adaptation of diffusion-based image generators for image analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 07 2025.

[43] M. Liu, M. A. Uy, D. Xiang, H. Su, S. Fidler, N. Sharp, and J. Gao, "Partfield: Learning 3d feature fields for part segmentation and beyond," *arXiv preprint arXiv:2504.11451*, 2025.

[44] X. Yu, P. Dai, W. Li, L. Ma, Z. Liu, and X. Qi, "Texture generation on 3d meshes with point-uv diffusion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4206–4216.

[45] W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang, "Mitsuba 3 renderer," 2022, https://mitsuba-renderer.org.

[46] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "Blenderproc," *arXiv preprint arXiv:1911.01911*, 2019.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14124313

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] K. Yan and G. Cai, "andyyankai/psdr-jit: v0.1.8," Sep. 2024. [Online]. Available:

https://doi.org/10.5281/zenodo.13777325