This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PhotoScene: Photorealistic Material and Lighting Transfer for Indoor Scenes

Yu-Ying Yeh[1] Zhengqin Li[1] Yannick Hold-Geoffroy[2] Rui Zhu[1] Zexiang Xu[2]
Miloš Hašan[2] Kalyan Sunkavalli[2] Manmohan Chandraker[1]

[1]University of California, San Diego [2]Adobe Research

Abstract

Most indoor 3D scene reconstruction methods focus on recovering 3D geometry and scene layout. In this work, we go beyond this to propose PhotoScene[1], a framework that takes input image(s) of a scene along with approximately aligned CAD geometry (either reconstructed automatically or manually specified) and builds a photorealistic digital twin with high-quality materials and similar lighting. We model scene materials using procedural material graphs; such graphs represent photorealistic and resolution-independent materials. We optimize the parameters of these graphs and their texture scale and rotation, as well as the scene lighting to best match the input image via a differentiable rendering layer. We evaluate our technique on objects and layout reconstructions from ScanNet, SUN RGB-D and stock photographs, and demonstrate that our method reconstructs high-quality, fully relightable 3D scenes that can be re-rendered under arbitrary viewpoints, zooms and lighting.

1. Introduction

A core need in 3D content creation is to recreate indoor scenes from photographs with a high degree of photorealism. Such photorealistic "digital twins" can be used in a variety of applications including augmented reality, photographic editing and simulations for training in synthetic yet realistic environments. In recent years, commodity RGBD sensors have become common and remarkable progress has been made in reconstructing 3D scene geometry from both single [14, 47] and multiple photographs [51], as well as in aligning 3D models to images to build CAD-like scene reconstructions [5, 22, 23, 36]. But photorealistic applications require going beyond the above geometry acquisition to capture material and lighting too — to not only recreate appearances accurately but also visualize and edit them at arbitrary resolutions, under novel views and illumination.

Prior works assign material to geometry under the simplifying assumptions of homogeneous material [23] or single objects [38]. In contrast, we deal with the challenge of ascribing spatially-varying material to an indoor scene while reasoning about its complex and global interactions with arbitrary unknown illumination. One approach to our problem would be to rely on state-of-the-art inverse rendering methods [29, 32] to reconstruct per-pixel material properties and lighting. However, these methods are limited to the viewpoint and resolution of the input photograph, and do not assign materials to regions that are not visible (either outside the field of view or occluded by other objects). Instead, we posit that learned scene priors from inverse rendering are a good initialization, whereafter a judicious combination of expressive material priors and physically-based differentiable rendering can solve the extremely ill-posed optimization of spatially-varying material and lighting.

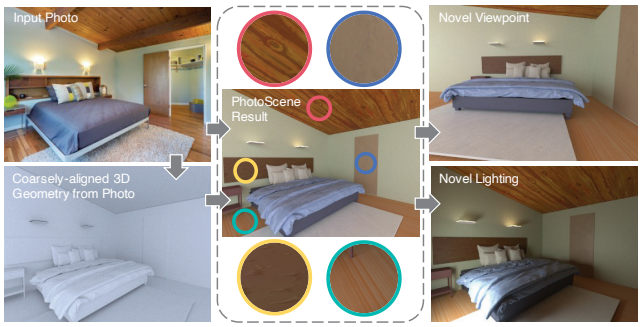

Figure 1. Given an input photo and a coarsely aligned 3D scene model, PhotoScene automatically infers high-quality spatially-varying procedural materials and scene illumination to closely match scene appearance. The reconstructed materials are resolution-independent (see zoom insets) and ascribed to the full 3D geometry, to create a high-quality photorealistic digital twin that can be rendered under novel views and lighting.

In this paper, we use procedural node graphs as compact yet expressive priors for scene material properties. Such graphs are heavily used in the content and design industry to represent *high-quality, resolution-independent* materials with a compact set of optimizable parameters [1, 3]. This offers a significant advantage: if the parameters of a procedural graph can be estimated from just the observed parts of the scene in an image, we can use the full graph to ascribe

[1]Code: https://github.com/ViLab-UCSD/photoscene

materials to the entire scene. Prior work of Shi et al. [43] estimates procedural materials, but is restricted to fully observed flat material samples imaged under known flash illumination. In contrast, we demonstrate that such procedural materials can be estimated from partial observations of indoor scenes under arbitrary, unknown illumination.

We assume as input a coarse 3D model of the scene with possibly imperfect alignment to the image, obtained through 3D reconstruction methods [5, 22, 36], or manually assembled by an artist. We segment the image into distinct material regions, identify an appropriate procedural graph (from a library) for each region, then use the 3D scene geometries and their corresponding texture UV parameterizations to "unwarp" these pixels into (usually incomplete) 2D textures. This establishes a fully differentiable pipeline from the parameters of the procedural material via a physically-based rendering layer to an image of the scene, allowing us to backpropagate the rendering error to optimize the material parameters. In addition, we also estimate rotation and scale of the UV parameterization and optimize the parameters of the globally-consistent scene illumination to best match the input photograph.

As shown in Fig. 1 our method can infer spatially-varying materials and lighting even from a single image. Transferring these materials to the input geometry produces a fully relightable 3D scene that can then be rendered under novel viewpoint or lighting. Since procedural materials are resolution-invariant and tileable, we can render closeup views that reveal fine material details, without having observed these in the input photograph. This goes significantly beyond the capabilities of current scene-level inverse rendering methods and allows for the creation of high-quality, photorealistic replicas of complex indoor scenes.

2. Related Works

Material acquisition and recognition High-quality materials have been estimated in many prior works, using both single [4, 12, 19, 30] or multiple [13, 15] input images. Most of the above methods estimate materials for planar samples, as opposed our inputs that are unconstrained images of complex indoor scenes. While material recognition methods have been proposed to classify image regions into material categories [8], they do not yield parametric materials that could be used for relighting and view synthesis. In recent years, several methods have been proposed to use procedural graphs as materials priors and estimate their parameters to match the appearance of captured images [17, 20, 43]. In particular, we use MATch [43], a differentiable procedural material model based on Substance node graphs, to constrain our materials to the SVBRDF manifold. However, the above methods only consider flat material samples captured under known flash illumination, while our goal is significantly more challenging – our inputs are photos of indoor scenes with complex geometry, lit by unknown spatially-varying illumination, with scene layout and occlusions leading to incomplete textures with arbitrary scales and rotation.

Inverse rendering of indoor scenes Our problem may be seen as an instance of inverse rendering [34, 39], but we must estimate materials that are amenable for rendering under novel views and lighting, as well as editability. Several approaches have been proposed for inverse rendering for objects [31, 33, 40, 42, 44, 48], but our focus is on indoor scenes, which have been considered in both early [7, 26, 27] and recent [29, 41] works. A convolutional neural network with a differentiable rendering layer estimates depths, per-pixel SVBRDF and spatially-varying lighting in [29] using a single input image. We use their material and lighting outputs as our initialization and to aid our differentiable rendering losses in image space, but go further to assign procedural materials that can be used for rendering novel views and estimate lighting that is consistent across views. Recent work has applied differentiable rendering [28] to recover spatially-varying reflectance and lighting given photos and 3D geometry [6, 37]. However, these methods estimate per-vertex BRDFs and require high-quality geometry as input, while our procedural models regularize scene materials, allowing us to infer them from only coarsely aligned 3D models and to re-render novel viewpoints with material detail that was not observed in the input photos.

Material transfer from photographs LIME [35] clones the homogeneous material from a single color image. Material Memex [24] and Unsupervised Texture Transfer [50] exploit correlations between part geometries and materials, or across patches for material transfer to objects. PhotoShape [38] assigns photorealistic materials to 3D objects through material classifiers trained on a synthetic dataset. In contrast, we seek material transfer in indoor scenes, which is significantly harder since image appearances entwine material properties with complex light transport. The material suggestion system of [9] textures synthetic indoor scenes with high-quality materials, but based on a set of pre-defined rules for local material and global aesthetics, whereas we must solve challenging inverse problems in a differentiable framework to match the appearances of real images.

Indoor scene 3D reconstruction Many works reconstruct indoor 3D scene geometry (objects and room layout) from single images [22, 23, 36] or RGBD scans [5], but either do not address material and lighting estimation, or use heuristics like median color to assign diffuse textures [23]. Our work focuses on reconstructing high-quality material and lighting from input photos and is complementary to geometric reconstruction methods; indeed, we can leverage these methods to build our input coarse 3D model. Like us, Plan2Scene [49] also aims to reconstruct textured 3D scenes, but is limited to diffuse textures and constrained by

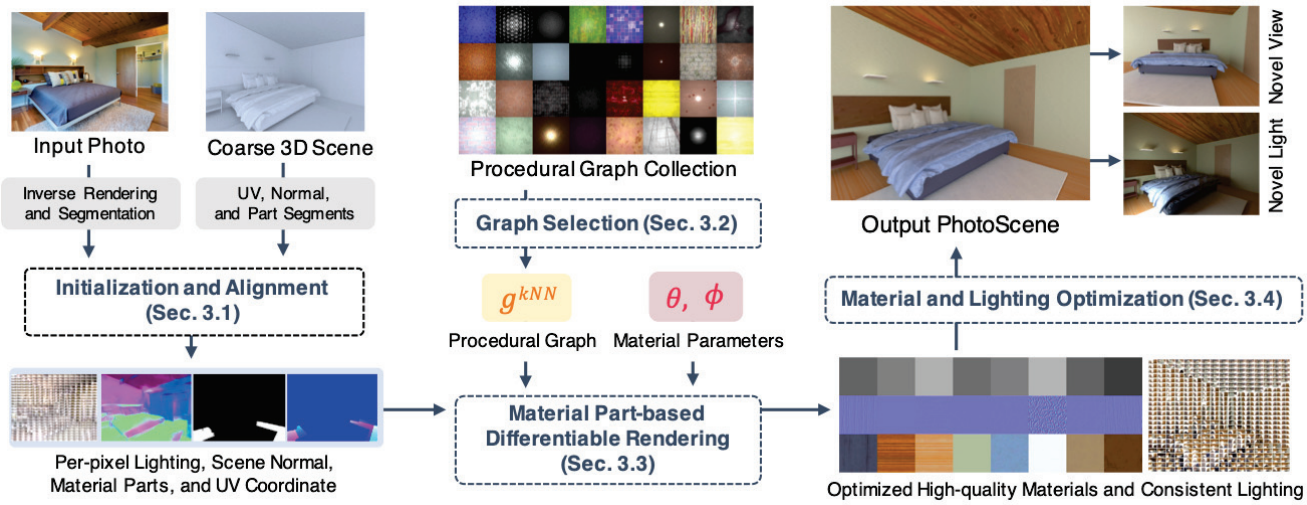

Figure 2. The PhotoScene framework. From the input photo(s) and 3D scene model, we estimate scene normals and lighting via an inverse rendering method, and compute material parts and align them to the model UVs and part segments. We model scene materials with procedural graphs. For each material part, we identify an appropriate graph from a collection and use a differentiable rendering module to optimize for the graph and UV transformation parameters. We also refine the initial lighting. Assigning the optimized materials and lighting to the input 3D model gives us our output PhotoScene—a renderable 3D scene that matches the appearance of the input photos.

the quality of the texture synthesis model. In contrast, by optimizing a procedural material model, we are able to recover high-quality non-Lambertian materials; we also optimize for illumination and hence better match the input image appearance.

3. Proposed Method

Our method starts from an input image (or multiple images) of an indoor scene and a roughly matching scene reconstruction (automatic or manual). Our goal is to obtain high-resolution tileable material textures for each object, as well as a globally consistent lighting for the scene.

The method consists of four high-level stages, as shown in Fig. 2. We compute an initial estimate of scene normals and lighting. We also find material parts, and align them between the input and rendered image, so that each material part can be optimized separately. Next, we choose a *material prior* for each material part, in the form of a procedural node graph that produces the material's textures (albedo, normal and roughness) given a small set of parameters. Finally, we optimize the parameters of all materials as well as the lighting in the scene. Below we describe this in detail.

3.1. Initialization and Alignment

In the initialization step, we obtain estimates of normals and lighting from the input image(s) that guide the subsequent optimization. Next, since the synthetic scene is composed of elements that are not perfectly aligned with the input photograph(s), we warp the pixels rendered from the geometry to best fit the scene structure in the input photograph *per material part*. If there are multiple input images per scene available, we perform consensus-aware view selection (see the supplementary materials for details).

Pixel-level normals and lighting initialization. We use the pretrained inverse rendering network (InvRenderNet) from Li et al. [29] to obtain spatially-varying incoming lighting estimates $\mathcal{L}^{\text{inv}}$ and per-pixel normals $\mathcal{N}^{\text{inv}}$ to guide the material optimization in pixel space (Sec. 3.3). We do not use the estimated albedo and roughness from InvRenderNet, except as baselines for comparison, as shown in Fig. 12.

Material part mask and mapping. For each material part, our method requires a mask M_{photo} to indicate the region of interest in the input image, and another mask M_{geo} to indicate the same in the synthetic image. The latter mask is trivially available by rendering the synthetic geometry. To obtain M_{photo} automatically, we make use of predictions from MaskFormer [10] as proposals, and find the mapping by computing maximum intersection over union (IoU) with respect to M_{geo} of all material parts. Semantic and instance labels (when available) can be used to reduce the proposals. To obtain more robust results, manually segmented masks can be taken as M_{photo} instead. More details can be found in the supplementary material.

Geometry/photo alignment and warping. To handle misalignment between material parts in the input image and the geometry, we compute an affine pixel warp. We obtain pixel locations $\mathbf{x}_s^*$ to sample values from M_{geo} of the geometry to warp to the material mask M_{photo} at pixel $\mathbf{x}_t$ as

$$\mathbf{x}_s^* = \frac{\mathbf{x}_t - \mathbf{c}_p}{\mathbf{l}_p} \cdot \mathbf{l}_g + \mathbf{c}_g \ , \tag{1}$$

where $\{\mathbf{c}_g, \mathbf{l}_g\}$ and $\{\mathbf{c}_p, \mathbf{l}_p\}$ are the centers and sizes of bounding boxes around the masks. As the affine warp is

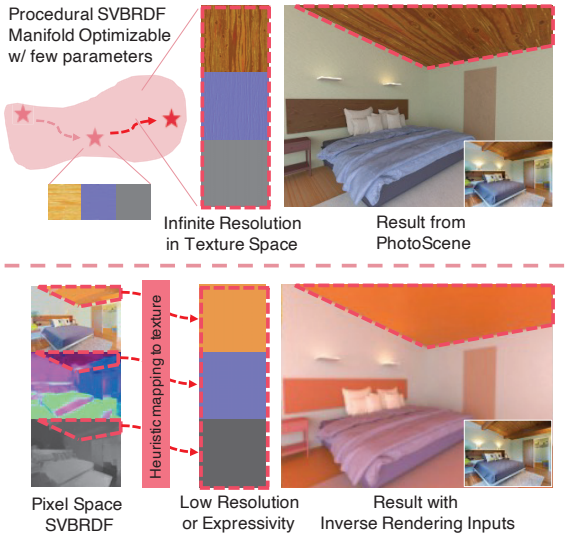

Figure 3. An expressive material prior optimizable with a few parameters allows recreating photorealistic appearances, in contrast to heuristics that may rely solely on pixel-space inverse rendering.

imperfect, some pixels will not have a correspondence and will be dropped. Please see supplementary for details.

At the end of this initialization and alignment step, we obtain per-material part UVs and corresponding masks in the input and synthetic images, as well as an initial estimate of scene lighting and normals.

3.2. Material Prior: Procedural Node Graphs

Modeling spatially-varying materials as 2D textures is difficult due to the ill-posed nature of the problem: the textures may not be fully observed in the input image, and they are lit by uncontrolled illumination. Therefore, a key step of our method is to constrain materials to lie on a valid SVBRDF manifold, by specifying a *material prior* that is expressive, yet determined by a small number of parameters. This material prior must be differentiable to allow parameter optimization via backpropagation. This is in contrast to a bottom-up approach that might rely on pixel space outputs from inverse rendering (see Fig. 3 and 12).

We use MATch [43], a material prior based on differentiable procedural node graphs. Their implementation provides 88 differentiable procedural graphs that model high quality spatially-varying materials, each with a unique set of parameters. For our purposes, these graphs are simply differentiable functions from a parameter vector θ to albedo, normal and roughness textures A_{uv}, N_{uv}, R_{uv}. We add an additional offset parameter for the albedo output from the graphs to more easily control the dominant albedo colors. We select 71 graphs that are representative of indoor scenes as our graph collection $\{g^1, ..., g^{71}\}$. We augment this set with a homogeneous material, used for untextured parts.

Material graph selection. For each material part in

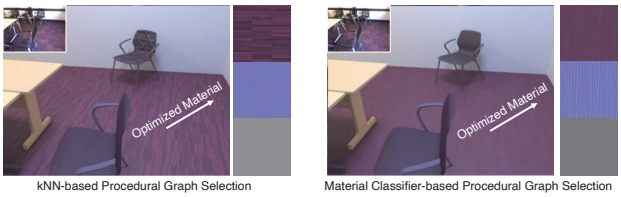

Figure 4. Graph selection with kNN versus material classifier.

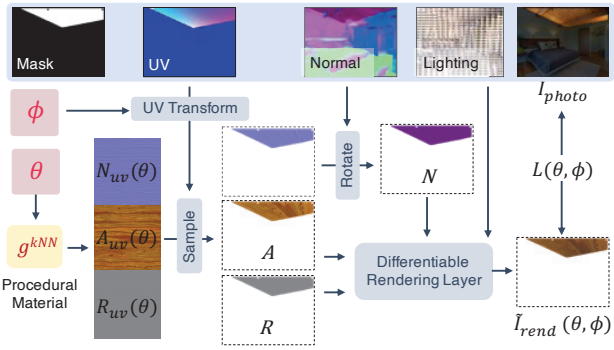

Figure 5. Given a material part mask, UV, scene normals and lighting (top), we construct a fully differentiable pipeline from material graph (θ) and UV transformation (ϕ) parameters via a texture-to-image mapping and differentiable rendering layer to a rendered image. We optimize for these parameters by comparing this rendering to the input photo.

the image, we need to choose an appropriate procedural node graph from the library. We address this by nearest neighbor search using a VGG feature distance. Specifically, we sample 10 materials from each of the 71 graphs with random parameters, resulting in 710 exemplar material maps $\{(A|N|R)^1_{uv}, \cdots, (A|N|R)^{710}_{uv}\}$. We render the part using each exemplar with our differentiable renderer (Sec. 3.3), resulting in 710 render-graph pairs $\{(\tilde{\mathcal{I}}^1_{rend}, g^1), (\tilde{\mathcal{I}}^2_{rend}, g^1), ...(\tilde{\mathcal{I}}^{710}_{rend}, g^{71})\}$, then select the k $(= 21)$ most similar renderings to the input using a masked VGG distance (Eq. 9), which vote for their corresponding graphs and we pick the graph g^{kNN} with the most votes.

We also experiment with predicting material super classes (e.g. wood, plastic, etc.) using a pretrained classifier and then selecting a graph from the class, but find the kNN search less susceptible to errors (Fig. 4). For small parts where it is difficult to observe spatial variations, we use homogeneous materials.

3.3. Material Part Differentiable Rendering

Our framework leverages a *material part differentiable rendering module* as a way to predict the appearance of a part given its texture-space material maps $(A|N|R)_{uv}$, pixel-space geometry $\mathcal{N}^{\text{inv}}$, and local lighting $\mathcal{L}^{\text{inv}}$. The differentiable rendering module can be used for optimization through back-propagation. We use it to optimize for material parameters in Sec. 3.4.

During rendering, we use a spatially-varying grid of in-

coming light environment maps [29] as our lighting representation, which allows the operation of the rendering module to remain local; no additional rays need to be traced by the rendering process, which is crucial for efficiency.

Our differentiable rendering module is shown in Fig. 5. Our material model is a physically-based microfacet BRDF [25]. The rendering module takes per-pixel texture (UV) coordinates, to sample material textures A_{uv}, N_{uv}, R_{uv} generated from the material prior using parameters θ as $g^{kNN}(\theta)$. The normals need to be rotated into the local shading frame of a given point on the material part. The rendering module then uses per-pixel material parameters A, N, R and spatially-varying local incoming lighting L to render the image pixels $\tilde{I}_{\text{rend}}$:

$$A, R = \mathbf{Sample}_{UV}(A_{uv}(\theta), R_{uv}(\theta)), \quad (2)$$

$$N = \mathbf{Rot}(\mathbf{Sample}_{UV}(N_{uv}(\theta))), \quad (3)$$

$$\tilde{\mathcal{I}}_{\text{rend}} = \mathbf{RenderLayer}(A, N, R, L). \quad (4)$$

As the originally assigned UV coordinates might not have the optimal scale and orientation to apply the corresponding material, we apply texture transformation parameters ϕ (rotation, scale, and translation) to map original coordinates UV^0 to more appropriate ones UV.

$$UV = \mathbf{UVTransform}(UV^0, \phi). \quad (5)$$

Fig. 9 provides visual examples of the importance of considering rotation and scale in our differentiable rendering.

3.4. Material and Lighting Optimization

Images conflate lighting, geometry, and materials into an intensity value. To better disambiguate lighting from material, we adopt a two-step process. First, we use our initial spatially-varying lighting prediction from InvRenderNet to optimize the materials for each object. Next, we perform a globally consistent lighting optimization to refine our illumination. Finally, we optimize the materials once more, this time using our refined lighting. This procedure reduces the signal leakage between material and lighting.

Material optimization. There is no exact correspondence between the rendered and reference pixels. Thus, we compute the absolute difference $\mathcal{L}_{stat}$ of the statistics (mean μ and variance σ^2) of the masked pixels of the part of interest to optimize both material prior parameters θ and UV transformation parameters ϕ:

$$\mathcal{L}_{\text{mean}} = |\mu(\mathcal{I}_{\text{photo}} \cdot M_{\text{aln}}) - \mu(\tilde{\mathcal{I}}_{\text{rend}} \cdot M_{\text{aln}})|, \quad (6)$$

$$\mathcal{L}_{\text{var}} = |\sigma^2(\mathcal{I}_{\text{photo}} \cdot M_{\text{aln}}) - \sigma^2(\tilde{\mathcal{I}}_{\text{rend}} \cdot M_{\text{aln}})|, \quad (7)$$

$$\mathcal{L}_{\text{stat}} = \mathcal{L}_{\text{mean}} + \mathcal{L}_{\text{var}}, \quad (8)$$

where M_{aln} is the resulting aligned mask from the alignment step.

Figure 6. Example of lighting optimization result using $N = 2$ ceiling lights and one environment map lighting. The optimized lighting is close to original ScanNet images.

Using this statistics loss encourages matching color distributions but not spatially-varying patterns. To further match the patterns, we add a masked version of VGG loss L_{vgg} [45]. Let $\tilde{C}_l$ and C_l be the normalized VGG feature maps of $\tilde{\mathcal{I}}_{\text{rend}}$ and I_{photo} extracted from layer l[2], we apply mask M_{aln} on the sum of upsampled L2 difference of normalized feature maps $\tilde{C}_l$ and C_l and compute the mask-weighted average among the pixels x:

$$\mathcal{L}_{\text{vgg}} = \frac{1}{\sum_x M_{\text{aln}}} \sum_x M_{\text{aln}} \left(\sum_l \text{Up}\left(\tilde{\mathcal{C}}_l - \mathcal{C}_l\right)^2 \right). \quad (9)$$

We also try a masked style loss based on the Gram matrices of VGG features [16], but find that it does not provide significant improvement over L_{stat} and L_{vgg}. Therefore, we use the following loss for material optimization:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{stat}} + \beta \mathcal{L}_{\text{vgg}}. \quad (10)$$

Rather than jointly optimizing for material and UV parameters, we find that convergence is more stable with alternately searching for UV parameters in a discretized space and optimizing for material graph parameters. Spatially-varying roughness parameters are difficult to optimize in a single view due to limited observations of highlights, so we replace the roughness output from the graph with a single mean value during optimization (the final result can still use the full roughness textures).

Globally consistent lighting optimization. To estimate globally consistent lighting, we represent indoor lighting as N area lights and one environment light which may be observed through the windows. We optimize for RGB intensities for each light source. For scenes without light source annotations, we uniformly place area lights on the ceiling every 3 meters of distance. With the materials we previously optimized, we render images with each single

[2]The layers used here are relu1_2, relu2_2, relu3_3, relu4_3, relu5_3.

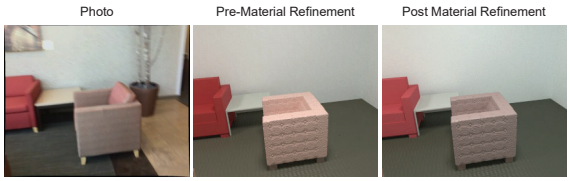

Figure 7. Second-round material refinement successfully corrects the inaccurate reflectance values estimated in the first round.

light source turned on. We compute the RGB intensities by comparing these renderings with the input view using least squares. Specifically, with N area lights (including ceiling lights and lamps) in a room with V input images, we solve for $3 \times (N+1)$ RGB coefficients $x_r, x_g, x_b \in \mathbb{R}^{N+1}$ (the $+1$ refers to an environment light visible through windows). We use these coefficients to re-weight the intensities of each light source. Fig. 6 demonstrates examples of rendering under selected views with each light source and the final combined optimized lighting. We additionally optimize for relative exposure values under different views, since they may vary over a video acquired using commodity cameras.

Material reoptimization. With the refined globally-consistent light sources, we re-optimize the materials to improve our results. We render a new spatially-varying incoming lighting grid L^{global} from the synthetic scene with the optimized light sources and use the same optimization loss as Sec. 3.3. We only optimize for a homogeneous rescaling of the albedo and roughness maps in this round, as the spatially-varying patterns are already correctly optimized by the first iteration. Fig. 7 demonstrates that this refinement step can rectify inaccurate material parameters caused by albedo-lighting ambiguities in the inverse rendering network.

4. Experiments

4.1. Datasets

We demonstrate our method on photos and corresponding scene data from several sources and demonstrate its robustness on images and scene geometry of varying quality.

ScanNet-to-OpenRooms. We use geometry and 3D part segmentations from OpenRooms [32], corresponding to *multi-view* input images from ScanNet [11] videos, with instance segmentation labels as mask proposals for M_{photo}.

Photos-to-Manual. For several high-quality real-world photos, we also manually construct matching scenes using Blender [2] from a *single view* and manually segment the material part masks for demonstration purposes.

SUN-RGBD-to-Total3D. Our method can also be used for fully automatic material and lighting transfer using a *single-image* mesh reconstruction from Total3D [36] applied to SUN-RGBD [46] inputs. The reconstruction in this case is coarser than CAD retrieval and with a single material per object. We use MaskFormer [10] to obtain segmentation labels as mask proposals and use object classes for mapping.

We assume all meshes have texture coordinates. If not, we use Blender's [2] Smart UV feature to generate them.

Figure 8. Example of material transfer results for different scenes with *ScanNet-to-OpenRooms*.

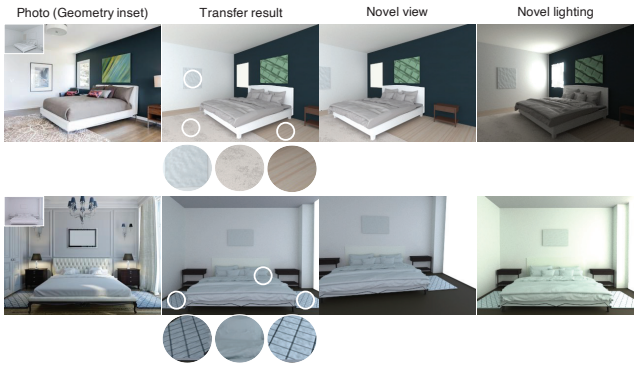

Figure 9. Our material and lighting transfer results for two scenes in *Photos-to-Manual* dataset. Note how our method is able to accurately reconstruct the appearance and orientation of the spatially-varying materials in these scenes.

4.2. Material and Lighting Transfer

We demonstrate our material and lighting transfer results in Fig. 8, 9 and 10, where our material prior allows interesting relighting effects such as specular highlights, shadows and global illumination under novel views and lighting.

Multiview inputs with CAD geometry. We demonstrate the multi-view setup in Fig. 8 with *ScanNet-to-OpenRooms* dataset. We select an optimal view for every single material to get the best observation as well as to estimate the global lighting for the entire room. Even though the CAD mod-

Figure 10. Examples of material transfer results with *SUN-RGBD-to-Total3D*. Note that these are fully automatic results by using Total3D to reconstruct 3D meshes and panoptic label predictions from MaskFormer. Our method is robust to imperfect geometry and panoptic prediction labels as shown in the inset.

Figure 11. Example of *editable* variations from originally optimized procedural graph materials for ceiling. We can perturb graph parameters or adjust UV parameters from optimized results to generate various appearances.

els are neither perfectly aligned nor perfect replicas of real objects, our method can closely match input appearances. More results are in supplementary.

Single-image inputs. Our method can even be applied to single image inputs, as shown in Fig. 1 and 9 with our own *Photos-to-Manual* dataset. Our framework allows material transfer for unseen portions of coarsely aligned CAD models from the photo, as shown in the novel view rendering. This makes the framework more practical than recent works [37] that require perfectly aligned geometry and multiview images to optimize for *observed* geometry. In the second row of Fig. 9, both *material* and *orientation* of the carpets with grid patterns are successfully estimated. Lastly, the estimated materials are high-resolution and photorealistic as shown in the zoom-in views and relighting results.

Automatic 3D reconstruction and masks. Our method can be fully automatic by using off-the-shelf single image 3D reconstruction and panoptic (or instance) prediction for initialization. We illustrate this in Fig. 10 with the *SUN-RGBD-to-Total3D* dataset. This shows that our method is robust even when both masks and meshes are imperfect and not aligned well, which also cannot be achieved in recent works [37] that need high-quality aligned geometry.

Variations from optimized material. Another advantage of procedural graph material representation is that it allows

	Classifier	InvRender Med.	Pixel Med.
Ours preferred over	68.19%	65.06%	69.70%

Table 1. User study asking which method produces results more similar to the reference with ScanNet-to-OpenRooms dataset.

	Classifier	InvRend. Med.	Pixel Med.	Ours
RMSE	0.452	0.349	0.337	**0.259**
SSIM	0.401	0.479	**0.497**	0.493
LPIPS	0.546	0.510	0.501	**0.489**

Table 2. Similarity evaluation between baselines rendering results and reference photo with ScanNet-to-OpenRooms dataset.

further edits from the current parameters. We can adjust material and UV parameters starting from the current estimation and generate edited results as shown in Fig. 11. Note that the image becomes brighter by perturbing graph parameters under the same lighting which explains materials can also change the brightness of an image. This demonstrates the benefit of globally consistent lighting optimization with material refinement stages, which ensures materials for each part are consistent under global lighting representation.

4.3. Baseline Comparisons

Material classifier. The most relevant work to ours is PhotoShape [38], which learns a material classifier from a dataset of shapes with material assignments. The input to the network is an image with an aligned material part mask. Although PhotoShape does not consider lighting or complex indoor scenes, we compare by mimicking their approach in our setting. We borrow the material classification model from [38] and re-train it in a whole scene setting, with classification of material parts over 886 materials and material category classification over 9 super-classes. For each input image associated with a material part mask, we predict one of 886 material labels. Implementation details can be found in supplementary materials.

Median of per-pixel material predictions. For this baseline, we can construct a homogeneous material from per-pixel predictions of the inverse rendering network [29] by computing median values of per-pixel albedo and roughness under selected view for each material in the masked region and setting the normal to flat.

Median of pixel values. We follow IM2CAD [23] to assign a homogeneous albedo as the median values of the 3 color channels independently within the masked region in the selected view for each material, set a fixed roughness value at 0.7 and use a flat normal.

Comparisons and user study. The comparisons of our results with baselines are shown in Fig. 12 with various datasets.[3] The material classifier can only predict material

[3]As none of the baselines can estimate global lighting well, we use our predicted lighting to render baseline images to ensure fair comparison.

from a predefined dataset, which is not guaranteed to match the actual appearance in the photo. The median of the inverse rendering method can generate appearances close to the photo, but the albedo color sometimes goes off due to the issue of albedo-lighting ambiguities. The median of photo pixels robustly computes the albedo color similar to the photo, but both the spatially-varying patterns and the roughness are not estimated. In contrast, our method can estimate accurate spatially-varying materials which is similar to the photo as well as the global lighting.

For quantitative evaluations, Table 2 reports similarity metrics (RMSE, SSIM, LPIPS) between photos and renderings of various methods with 70 randomly sampled scenes, consisting of 669 material parts, using ScanNet-to-OpenRooms dataset under uniformly sampled views in Table 2. We compute RMSE on the optimized region for each material, while SSIM and LPIPS are on the entire image. Note that these similarity metrics are not designed to evaluate similarity between misaligned images or to evaluate spatial variations, so tend to favor homogeneous outputs of the median-based methods. Nevertheless, PhotoScene outperforms all baselines on these metrics, except pixel median in SSIM. Note that the homogeneous albedo from pixel median may match a photo well on an average, but without spatial variations or accurate relighting in new views.

To evaluate methods with human perception, we provide a user study to evaluate the similarity in Table 1 using the same dataset. We choose 20 random scenes with uniformly sampled 4 to 12 views and render a set of images under selected views with our result. We ask users on Amazon Mechanical Turk to determine which set of images is more similar to the corresponding photo set. More details can be found in the supplementary material. About 65 to 70% users think PhotoScene generates results more similar to the inputs. Thus, our method outperforms the baselines both qualitatively and quantitatively.

More analysis. We further conduct an ablation study of component choice and robustness of using different ways to obtain material part masks in the supplementary.

Discussion and Limitations. Our algorithm assumes a part segmentation already exists in the reconstructed geometry, and our results depend on its quality and granularity. A finer part segmentation could be achieved by retrieving objects from a higher quality CAD model collection. Our graph collection is limited to the set provided by the existing implementation of MATch [43], though more general procedural graphs could be added with some effort, possibly using automatic techniques [21]. Our approach cannot handle specific patterns such as paintings, which could be addressed by training a generative model for such materials [18]. Lastly, we rely on a neural inverse rendering initialization, where the current state-of-the-art is restricted to small resolutions, so some high-frequency information

Figure 12. Qualitative comparison[3] with baselines on various datasets, where our method generates high-quality materials with spatially-varying patterns that better match the input photograph.

from photos might be lost. This will likely be improved by future architectures handling higher resolutions. We note a potential negative impact of spurious edits (Deepfakes) of indoor scenes, which we discuss further in supplementary.

5. Conclusion

We have presented a novel approach to transfer materials and lighting to indoor scene geometries, such that their rendered appearance matches one or more input images. Unlike previous work on material transfer for objects, we must handle the complex inter-dependence of material with spatially-varying lighting that encodes distant interactions. We achieve this through an optimization that constrains the material to lie on an SVBRDF manifold represented by procedural graphs, while solving for the material parameters and globally-consistent lighting with a differentiable renderer that best approximates the image appearances. We demonstrate high-quality material transfer on several real scenes from the ScanNet, SUN-RGBD dataset and unconstrained photographs of indoor scenes. Since we estimate tileable materials that can be procedurally generated, the scenes with transferred material can be viewed from novel vantage points, or under different illumination conditions, while maintaining a high degree of photorealism. We believe our work may have significant benefits for 3D content generation in artistic editing and mixed reality applications. Further, our approach can be used to create datasets for inverse rendering, where geometry is easier to acquire but ground truth material and lighting are hard to obtain.

Acknowledgments: We thank NSF awards CAREER 1751365, IIS 2110409 and CHASE-CI, generous support by Adobe, as well as gifts from Qualcomm and a Google Research Award.

References

[1] Adobe Stock. https://stock.adobe.com/3d-assets. 1

[2] Blender. http://www.blender.org. 6

[3] Substance Share. https://share.substance3d.com/. 1

[4] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Trans. Graph.*, 35(4), July 2016. 2

[5] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. *arXiv preprint arXiv:2003.12622*, 2020. 1, 2

[6] Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2

[7] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 2

[8] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[9] Kang Chen, Kun Xu, Yizhou Yu, Tian-Yi Wang, and Shi-Min Hu. Magic decorator: Automatic material suggestion for indoor digital scenes. *ACM Trans. Graph.*, 34(6), Oct. 2015. 2

[10] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint arXiv:2107.06278*, 2021. 3, 6

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 6

[12] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)*, 37(4):1–15, 2018. 2

[13] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. *Computer Graphics Forum*, 38(4):1–13, 2019. 2

[14] Clara Fernandez-Labrador, Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, and Jose J Guerrero. Layouts from panoramic images with geometry and deep learning. *IEEE Robotics and Automation Letters*, 3(4):3153–3160, 2018. 1

[15] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4), July 2019. 2

[16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 5

[17] Yu Guo, Milos Hasan, Ling-Qi Yan, and Shuang Zhao. A bayesian inference framework for procedural material parameter estimation. *CoRR*, abs/1912.01067, 2019. 2

[18] Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. Materialgan: reflectance capture using a generative svbrdf model. *arXiv preprint arXiv:2010.00114*, 2020. 8

[19] Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. Generative modelling of brdf textures from flash images. *arXiv preprint arXiv:2102.11861*, 2021. 2

[20] Yiwei Hu, Julie Dorsey, and Holly Rushmeier. A novel framework for inverse procedural texture modeling. *ACM Trans. Graph.*, 38(6), Nov. 2019. 2

[21] Yiwei Hu, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. An inverse procedural modeling pipeline for svbrdf maps. *arXiv preprint arXiv:2109.06395*, 2021. 8

[22] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 187–203, 2018. 1, 2

[23] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2017. 1, 2, 7

[24] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel. Material memex: Automatic material suggestions for 3d objects. *ACM Transactions on Graphics (TOG)*, 31(6):1–8, 2012. 2

[25] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4:3, 2013. 5

[26] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):1–12, 2011. 2

[27] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics (TOG)*, 33(3):1–15, 2014. 2

[28] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 2

[29] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan 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*, pages 2475–2484, 2020. 1, 2, 3, 5, 7

[30] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 72–87, 2018. 2

[31] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single

image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[32] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, and Manmohan Chandraker. OpenRooms: An end-to-end open framework for photorealistic indoor scene datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 6

[33] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2261–2269, 2017. 2

[34] Steve Marschner. Inverse rendering for computer graphics. 1998. 2

[35] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6315–6324, 2018. 2

[36] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun 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*, pages 55–64, 2020. 1, 2, 6

[37] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering. 2021. 2, 7

[38] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *ACM Trans. Graph.*, 37(6), Nov. 2018. 1, 2, 7

[39] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. *Computer Graphics Forum*, 22(4):663–687, 2003. 2

[40] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *ECCV*, 2020. 2

[41] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 2

[42] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6296–6305, 2018. 2

[43] Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekeur, Radomir Mech, and Wojciech Matusik. Match: Differentiable material graphs for procedural material capture. *ACM Trans. Graph.*, 39(6):1–15, Dec. 2020. 2, 4, 8

[44] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[46] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 6

[47] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019. 1

[48] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017. 2

[49] Madhawa Vidanapathirana, Qirui Wu, Yasutaka Furukawa, Angel X. Chang, and Manolis Savva. Plan2scene: Converting floorplans to 3d scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10733–10742, June 2021. 2

[50] Tuanfeng Y. Wang, Hao Su, Qixing Huang, Jingwei Huang, Leonidas Guibas, and Niloy J. Mitra. Unsupervised texture transfer from images to model collections. *ACM Trans. Graph.*, 35(6), Nov. 2016. 2

[51] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019. 1