

Material Swapping for 3D Scenes using a Learnt Material Similarity Measure

Maxine Perroni-Scharf¹, Kalyan Sunkavalli², Jonathan Eisenmann², Yannick Hold-Geoffroy²

¹Princeton University, ²Adobe Research

maxi@princeton.edu, sunkaval@adobe.com, eisenman@adobe.com, holdgeof@adobe.com

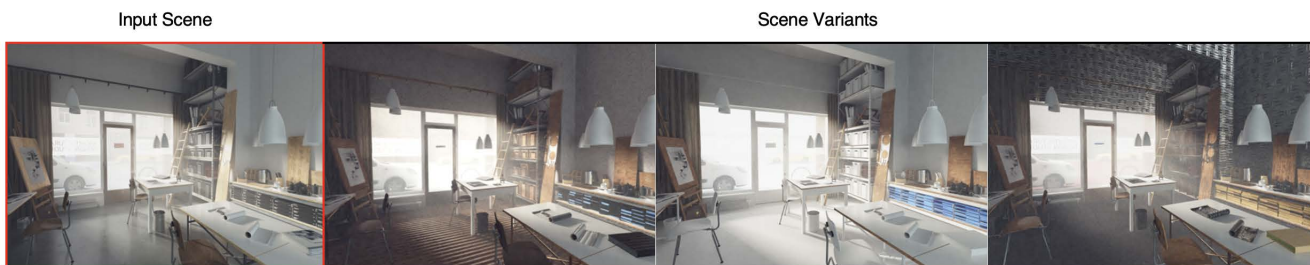


Figure 1. We propose a method that, given an input 3D scene (left), uses a learnt material similarity measure to automatically match scene materials to similar materials from a dataset and produce multiple high-quality scene variations (three variants shown on the right).

Abstract

We present a method for augmenting photo-realistic 3D scene assets by automatically recognizing, matching, and swapping their materials. Our method proposes a material matching pipeline for the efficient replacement of unknown materials with perceptually similar PBR materials from a database, enabling the quick creation of many variations of a given 3D synthetic scene. At the heart of this method is a novel material similarity feature that is learnt, in conjunction with optimal lighting conditions, by fine-tuning a deep neural network on a material classification task using our proposed dataset. Our evaluation demonstrates that lighting optimization improves CNN-based texture feature extraction methods and better estimates material properties. We conduct a series of experiments showing our method’s ability to augment photo-realistic indoor scenes using both standard and procedurally generated PBR materials.

1. Introduction

Synthetic data is key for training computer vision algorithms that require very large labeled datasets of photo-realistic image content. Unlike real-world data, synthetic data can easily be annotated with detailed labels of content such as semantics, geometry, appearance, pose and lighting, and can be used for supervised training tasks. Creating these synthetic datasets, however, can be time-consuming

and requires a significant amount of manual input from human artists, or the adoption of cost-intensive generative approaches that struggle with adapting to unfamiliar types of scenes or subject matter. Moreover, training deep networks on this data so that they generalize well to test data requires augmenting this synthetic data to generate as many plausible variations as possible [33].

This work aims to leverage, extend and improve existing synthetic 3D scene datasets by automatically replacing high-quality synthetic scene materials with similar (standard or procedural) materials from a database. Our method generates perceptually plausible scene variants, as shown in Fig. 1, which can significantly increase the size and variety of synthetic datasets. As an extension, procedural textures allow us to generate additional scene variants quickly and at little cost. Furthermore, this method is adaptable to any textured input scene and does not require additional information about the scene or the original scene materials.

Finding suitable matching materials requires establishing a material similarity metric. In order to identify good candidate material replacements, we propose a learning-based feature extraction method for PBR materials comprising of albedo, height, normal, metallic, roughness, opacity, and ambient occlusion maps. This method involves placing each texture onto a sphere, rendering images of the sphere, and extracting deep features (in our case, by using a fine-tuned VGG-16 network [29]) from them. An important design choice in this framework is the lighting that the spheres are rendered in. We simultaneously optimize for

the optimal lighting along with the feature extractor for the material classification task. We do so by using one-light-at-a-time (OLAT) renderings and estimating their optimal weights when constructing the final image.

For each material in a given 3D scene, we extract features and use them to find similar materials in a database. We also account for material scale in this matching process. Finally, we replace the input scene materials with the found scene materials to generate variations like the ones shown in Fig. 1. As can be seen here, our method creates a wide diversity of scene appearance that is visually plausible because of our material similarity metric.

We summarize our contributions as:

- A material matching pipeline, allowing the perceptually plausible swapping of textures;
- A differentiable OLAT-based image merging process, enabling the optimization of lighting conditions to improve ease of discrimination of materials;
- A dataset of materials annotated with classes.

2. Related Work

Texture Features and Similarity Estimation Texture analysis is key for many computer vision-related tasks, such as scene understanding, image annotation, and object recognition [37]. Earlier work has studied methods for retrieving meaningful low-dimensional embeddings from textures [4, 20]. Neural networks have shown impressive capabilities to estimate human perception of appearance and texture [39]. Our method relies on such embeddings to identify candidate material replacements for input scenes. Extensive research has explored the use of convolutional neural networks for texture feature extraction and classification [28, 31]. CNN features have also been used for material estimation [10, 26]. These methods typically extract features from input images using a pre-trained neural network, and compute a distance on the resulting features. We adopt this approach in our work, using a VGG-16 architecture [29] to extract features from renders of textured spheres.

Texture classification is typically performed on photographs captured in the wild [36], rather than PBR materials represented by multiple parameter maps. When working with photographs of textures, one has to make the learned features robust to variations in illumination, environment, scale, viewpoint, contrast, and color temperature. We avoid many of these complications by rendering our materials onto a sphere under a consistent viewpoint and lighting environment. In fact, we explicitly optimize the lighting environment for our renderings to the most discriminative lighting conditions for material classification.

Material Similarity Measures Previous work has attempted to model human perception of materials via crowd-

sourced material attribute labeling [18, 19, 24, 35]. A number of studies have also focused on the perception of individual aspects of a material such as gloss [22] or translucency [8]. Most of these previous methods focus on homogeneous BRDFs, i.e., they do not model spatially-varying material appearance. In our work, we aim to extract features that combine the effects of all the attributes of a given spatially-varying material by acquiring images of the material under the most discriminative possible lighting conditions.

To compare materials, prior work typically render materials on specific shapes under specific lighting conditions. The choice of lighting is extremely important; for example, human viewers might rely on the presence of specular highlights to distinguish between shiny and matte materials, and the proper lighting conditions will make this an easier task. Fleming et al. [7] study the importance of illumination and environment for human material matching tasks. They found that, when answering texture understanding questions, humans relied on their stored assumptions about the world; that is, subjects performed better at estimating surface reflectance properties when the objects they were observing were under realistic illumination conditions.

Havran et al. [11] propose to optimize both lighting and 3D shape to ensure the best possible sampling of the BRDF. In contrast, we optimize the lighting to directly aid material classification. Lagunas et al. [18] propose a perceptual similarity metric for point-wise BRDFs. They do so by rendering objects under a specific environment illumination, collecting crowd-sourced perceptual similarity measurements and training a deep model to predict features that correlate with human perception. Serrano et al. [25] extend this work to analyze material perception under a set of nine lighting conditions and also focus on specific perceptual attributes of materials. While our goal is similar, we differ in a number of ways: we seek a material similarity metric for *spatially-varying* BRDFs, train our deep features on the material classification task (without any perceptual labels) and explicitly optimize for a lighting condition that improves the discriminative power of our extracted features.

Inspired by Xu et al. [38], we develop a procedure to find the optimal lighting conditions for learning a performant material similarity metric for material swapping. Our approach relies on OLAT rendering, where images of a subject are captured with a single point light. Prior work also uses OLAT rendering to estimate the reflectance field of a subject [12] and to re-light portraits [30].

Data Augmentation Data augmentation is a concept widely used to improve model robustness and generalizability to real-world applications [27]. For example, the simple process of horizontally flipping training images can lead to differences in a model’s performance [17]. In the past, data-

augmentation has been used for a variety of applications, including as a safeguard against overfitting during image classification [23] and to regularize generative models [15].

Prior work has used various techniques for augmenting synthetic 3D scenes before rendering them into images. One approach is mixed sample augmentation, which consists of taking an existing 3D dataset and generating mixtures of individual samples from this dataset [9, 13]. For instance, this can be achieved by directly taking the union of the actual 3D mesh of different scenes in the dataset [21]. The analysis in this prior work demonstrates the merits of increasing the size and variety of 3D scene datasets for scene understanding tasks. However, this approach is computationally expensive, requires careful tailoring of mixing proportions based on scene context, and does not guarantee that the generated scenes are always plausible.

Simply augmenting scene renders via the randomization of color, lighting, and texture has also been shown to help deep learning models generalize from performing well on synthetic data to performing well on real-world data [32]. For example, neural style transfer for data augmentation [14] transforms the entire style of original image data to produce style variants of the same scene. While this approach maintains the consistency of the style of objects within each scene, it affects the global style of resulting scene images. In our work, we instead augment scene materials individually, allowing us to maintain consistency between all other aspects of the scene renders in our dataset, such as illumination, style, and post-processing.

3. Material Similarity Measure

We are interested in measuring material similarity between materials that can be applied to 3D assets. A natural choice for this task is neural networks, notably VGG [29], which was demonstrated to be particularly effective at extracting information from texture. Our insight is to fine-tune a pre-trained VGG-16 on a material similarity task, for which we can create annotations. We want to evaluate the similarity of the materials’ appearance when rendered in a 3D scene that requires specific lighting. To maximize how distinguishable the materials are under various lighting conditions, we devise an end-to-end lighting optimization.

In the following, we detail the network we use for feature extraction, our lighting optimization process, and an evaluation of the material matching we obtain using our method.

3.1. Network

Feature Extraction To extract feature vectors—or embeddings—from the material, we start from a pre-trained VGG-16 f_θ to process the material renders R_w , which is forwarded through the neural network as

$$\hat{y}_{fc} = f_\theta(R_w), \quad (1)$$

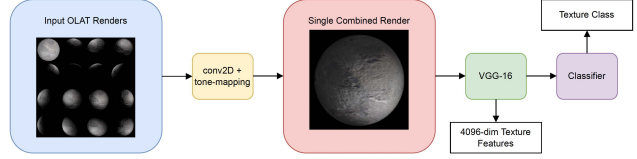


Figure 2. Our lighting optimization and material feature extraction scheme. Input EXR files are rendered out, and then a Conv2D layer combines them into one linear sum. This is tone-mapped into a PNG style input for a VGG, and 4096-dim features are extracted from the penultimate layer.

where $\hat{y}_{fc} \in \mathbb{R}^{4096}$ is the feature vector output by the penultimate layer of VGG-16.

Fine-tuning To improve the fitness of the extracted feature vectors \hat{y}_{fc} to our task, we fine-tune the last three layers of the model on a texture classification task. We use a linear layer from 4096-dim to 8-dim to obtain the material class output $\hat{y}_c \in \mathbb{R}^8$. We fine-tune the whole network using the loss

$$\mathcal{L}_c = \ell_{\text{nll}}(\mathbf{y}_c, \hat{y}_c), \quad (2)$$

where \mathbf{y}_c is the ground truth class of the texture and ℓ_{nll} is the negative log-likelihood function. We train the neural network f starting from the pre-trained weights θ to obtain our fine-tuned classifier f_{θ^*} ,

$$\theta^* = \arg \min_{\theta} \mathcal{L}_c. \quad (3)$$

3.2. Lighting Optimization

We are interested in matching the appearance of textures under any lighting condition. To this end, we adopt the following pipeline, as summarized in Fig. 2. First, we render 43 images $\mathbf{R} = \{R_1, \dots, R_{43}\}$ from a material using Blender with the Cycles renderer [6]. The setup of our scene consists of a sphere, a background image, and 42 directional lights that point at the sphere, corresponding to 42 of our renders. The lights are evenly spaced and equidistant from the material sphere, located at the vertices of an 42-vertex subdivided icosahedron. In addition to these directional light renders, we also render a single image with uniform diffuse lighting; this constitutes our 43rd render R_{43} .

We apply the input texture onto the sphere and then turn the directional lights on one at a time to render a 224x224 linear unsaturated HDR image using the OpenEXR format. This process produces a series of renders of the textured sphere under single isolated directional lighting conditions, as can be seen in Fig. 3.

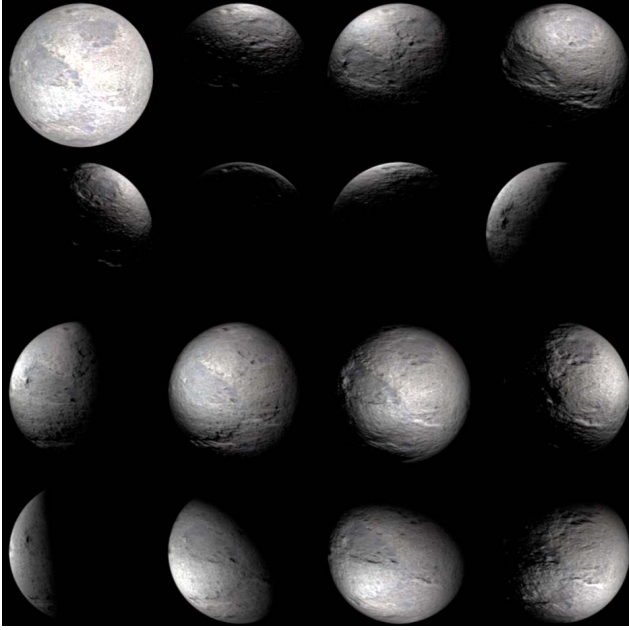


Figure 3. A sample of OLAT renders of a concrete material with ambient light (top left) and directional lights.

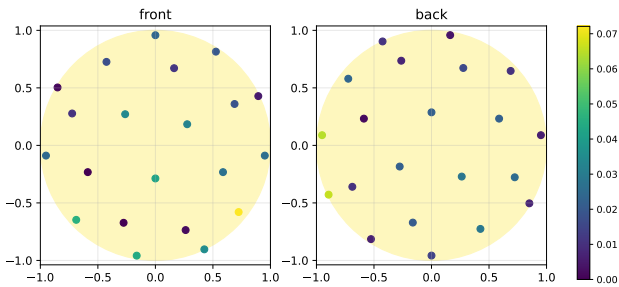


Figure 4. Results of the lighting optimization. We place a light on each vertex of a 42-vertex subdivided icosahedron and optimize the light intensity for material classification. The ambient term (background circle) is the strongest, and the weights sums to 1.

These 43 renders cover the spectrum of appearance of the material well. However, we realize that not all lighting directions convey the same information, and many are redundant. Therefore, we linearly combine all the renders and optimize their weights $w = \{w_1, \dots, w_{43}\}$ to increase how distinguishable the material properties are. Inspired by [38], we perform this optimization jointly with the fine-tuning of eq. (3). This end-to-end optimization provides our render weights w as the parameters of the first layer of the network.

Our optimized weights w are shown in Fig. 4, where the ambient lighting R_{43} dominates with few strong directional lights from the sides. We apply our lighting scheme to a set

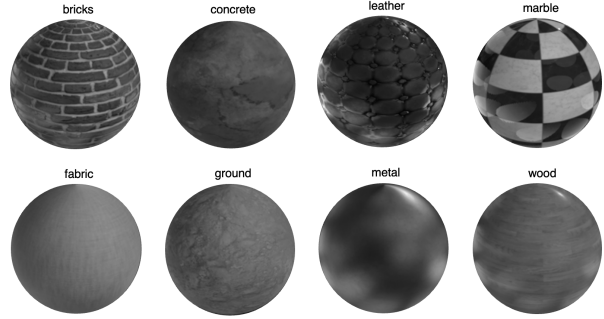


Figure 5. Visualization of the optimal lighting chosen by the network on grayscale renders of materials from each category. The network consistently created strong lights around the edge of the object, and favored a stronger asymmetrical light on one side of the object.

of renders R using

$$R_w = T \left(\sum_{i=1}^{43} w_i R_i \right), \quad (4)$$

where T is the tone-mapping operator to convert from linear to sRGB space [5].

3.3. Dataset

To train and evaluate our method, we downloaded 750 publicly available PBR materials from three different online sources [1–3]. We chose the following eight classes as they are particularly relevant to indoor scenes: bricks, concrete, fabric, ground, leather, marble, metal, wood (categories assigned by the authors).

We apply the above feature extraction process to each material in this dataset. Note that, although extracting texture features for the whole database is time-consuming and can take around one hour for 1000 textures on our GPU, this only needs to be done once; after this, we can use our database of candidate replacement texture features for any new input scene. We can compute the feature vector embedding y_{fc} of a new unknown material and perform a nearest neighbor search in our database to retrieve the closest matches, a process which we will discuss in the next section.

3.4. Implementation Details

We implement our network using PyTorch. For the lighting optimization, the first depth-wise 2D convolution in the network has 43 in channels (corresponding to the 43 OLAT renders for each texture, see Fig. 4) and one out channel, 1x1 kernel, stride 1 and padding 0 for OLAT image combination. Using the ADAM optimizer [16] with an initial learning rate of 1×10^{-4} , we train the model for 30 epochs,

Input	Lighting	Accuracy (%)
Color, no finetuning	uniform	66.47 \pm 0.49
Grayscale	uniform	80.32 \pm 0.46
Grayscale (ours)	optimized	84.91 \pm 0.62
Color	uniform	88.01 \pm 0.47
Color (ours)	optimized	90.31 \pm 0.74

Table 1. Ablation study showing the performance of our model in material classification with and without OLAT lighting optimization. Our method significantly outperforms vanilla VGG (top) and consistently improves with optimized lighting.

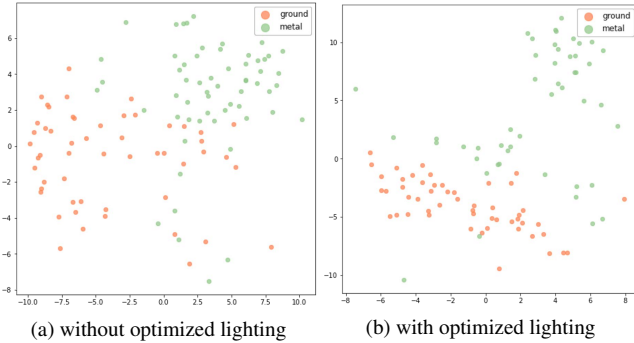


Figure 6. t-SNE plots for metal and ground without and with OLAT optimization. The features produced for metal and ground are more clearly separated when we optimize for lighting.

which took around 8 hours on a NVIDIA Tesla K80. We automate the process of Blender scene texture extraction, matching and replacement by utilizing Blender’s scripting functionalities.

3.5. Evaluation

Texture Classification Accuracy We now perform a series of experiments to assess the capacity of our network to produce distinguishable features representative of the material and its properties.

First, we compare the performance of our classifier with and without fine-tuning the lighting conditions of our renders in Table 1. We use a train/train split of 60/40 on our standard dataset of 750 textures (Section 3.3). The baseline without fine-tuning directly uses the feature vectors from a vanilla VGG-16. For uniform lighting, we train and test on rendered material spheres with uniform diffuse lighting. For the optimized lighting, we allow the network to adjust the weights of the OLAT renders to arrive at the optimal non-uniform lighting distribution, as described in Section 3.2. We also experiment using grayscale and color renders as inputs to the network. We repeat each training five times and report in the table the averages and max-min ranges of our resulting accuracy.

We also save the final weights of the conv2D layer on

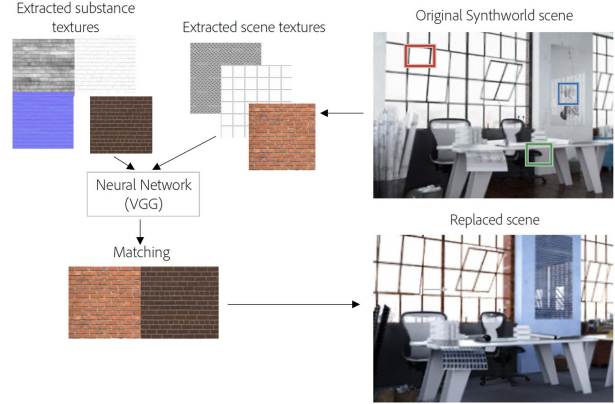


Figure 7. Our novel material replacement pipeline. Maps are extracted from an input scene materials and source candidate materials, and candidate matches are identified and inserted into the original scene.

the OLAT renders from these experiments. These are the lighting weights that enabled the VGG to classify textures from our dataset with the highest accuracy. Example renders lit by this weighted lighting is visualized in Fig. 5. The lighting appears to reveal strong specular highlights on the edges of shiny materials, and is stronger on one side of the sphere than on the other—rather than being incident from the same direction as the camera—which also emphasizes the impact of normal maps, height maps and bump maps on rougher materials.

t-SNE plots Using our network, we extracted the 4096-dim features \hat{y}_{fc} from our test set and studied these features. We used t-SNE [34] to reduce the dimensionality down to 2 dimensions and visualize the separation between features of different materials. In Fig. 6, we compare the t-SNE plots of the network with and without lighting optimization for the categories metal and ground. We compare these categories as textures in the metal category are typically shiny, and those in ground are typically matte, so we hypothesize that specular highlights as emphasized by our optimized lighting should be important for differentiating between these material categories. We find that lighting optimization more clearly differentiate and disentangle metal and ground materials from each other, as fewer points are overlapping and result in better defined clusters.

4. Material Swapping for 3D Scenes

We now employ the material similarity measure we devised in Section 3 to replace materials in existing 3D scenes. A summary of our replacement pipeline is shown in Fig. 7.

We seek to find top candidate replacements in our



Figure 8. Our pipeline chooses to leave original materials in the scene if it cannot find a suitable replacement. The three images above are the albedo maps of materials our pipeline ignored.

datasets for each original scene material in an input scene. We first compute the feature vectors as described in Section 3.1 for all the materials in our datasets and save them in a feature vector database. For a given input 3D scene, we perform the same feature extraction for the material of one of its assets. We use the cosine similarity between this material and all the materials from the database to retrieve the closest feature vectors from the database. We then swap this material in the 3D scene with the retrieved material. We can repeat this process for all the assets present in the scene.

Filtering Poor Matches Some 3D assets have their geometry and texture tightly coupled. This includes highly structured surfaces containing written text or where particular texture coordinates map to a specific location in 3D. Replacing these materials would be detrimental to the scene semantics. As such, we automatically detect and filter such materials for which we cannot find a suitable match in our database.

To ensure that we only replace materials with a good plausible candidate, we use a threshold on the cosine similarity when performing material replacement. We empirically found that a threshold of 0.28 provides a good balance to achieve plausible yet diverse material replacement. Therefore, we leave all materials as-is in the 3D scene if their closest match is larger than this value. Examples of materials filtered by our method are shown in Fig. 8.

Scaling We do not assume any prior information or consistency in the scale and UV mapping of materials in the source data. Inconsistent scales and resolutions for the original material maps can cause replacement textures to have implausible scales. For example, the wood grain on a table may end up becoming significantly enlarged during replacement. To identify the best scale to use for each material match, we render ten different tiled variants of each texture in our dataset. First, we find the closest texture match for an input texture. Then we run our model on the scaled variants of this texture match, and then choose the scaled variant with the closest embedding to the input texture’s embedding

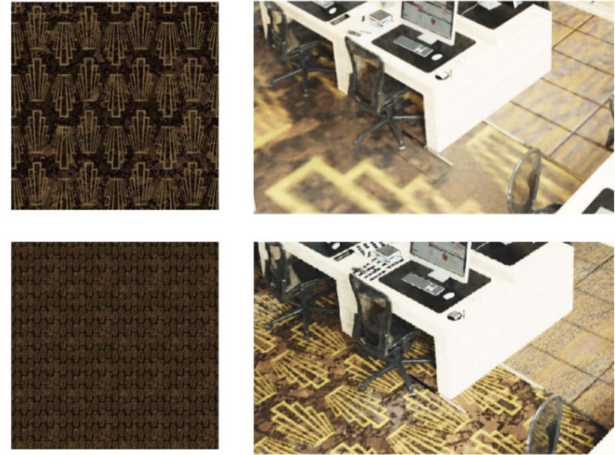


Figure 9. Before (top) and after (top) effects of learning material scale. The wood grain on the ground and the pattern on the carpet material in the top image are too large. Scaling fixes most of these issues.

rather than just using the default texture scale, as shown in Fig. 9.

4.1. Evaluation

Procedural Materials Dataset We evaluate our scene material replacement method using the publicly available Adobe Substance Source and Share PBR datasets [1].¹ These procedural materials have the advantage of being parametric: each material expose several parameters to the user which alter its appearance. This allows the generation of a large corpus of appearances with little effort. From the 400 procedural materials, we take advantage of those parameters to obtain 4,000 texture maps. We employ this commercial dataset only to evaluate and showcase the flexibility of our method; it is not needed to reproduce our feature extraction method.

Similarity Measure We use the cosine distance between extracted feature vectors to assess the similarity between materials. We retrieve the top five closest substances for each original material based on our similarity measure, and use these as the candidate replacement set for each material. For example, Fig. 10 shows the results of using cosine differences to obtain candidate replacements from the Adobe Substance Source dataset on three different input textures.

Qualitative Scene Assessment We report the results of our pipeline tested on several 3D scenes from Evermo-

¹<https://substance3d.adobe.com/assets/>








Input Texture	Five Closest Matches (as identified by our network)					
						
						
						

Figure 10. Albedo maps of input materials (left column) and their closest matches obtained by our VGG-based material matching model.

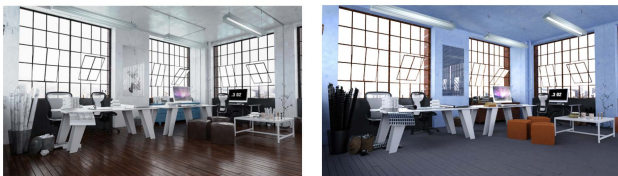


Figure 11. Input scene from Evermotion (left) and results from our pipeline (right).



Figure 12. Comparison between random material assignment and similarity-based material assignments on an input scene of an office space. Our method produces more plausible and aesthetically pleasing variations.

tion,² which comprises of photo-realistic synthetic scenes

²<https://evermotion.org/>, specifically several scenes from ArchInteriors Vol. 33.

of rooms and buildings.

We achieve promising qualitative results by running sample scenes through our material swapping pipeline, and rendering these results. Fig. 10 presents sample material matches directly to illustrate the results of our material matching model. For each input scene, we used the top material matches and default generative parameters to inspect the scenes side-by-side. For example, in Fig. 11, our pipeline successfully replaced the transparent window material with another transparent glass material, and all of the material replacements appear to be qualitatively plausible.

In Fig. 12, we can see that our method is qualitatively more successful than randomly replacing the materials in a given scene. In the randomly replaced version, there is an out-of-scale floor, a shutter wall, a pink concrete desk and a gravel texture on the double bass. In our version, we have successfully scaled the varnished floor texture. The candidate replacements all appropriately match the objects in the scene, such as a wooden texture replacement that was chosen for the double-bass and desk.

We also generated a number of scene variants for our test scenes. We used the top three matches for each material, and for each of these matches we created three variants by automatically modifying generative substance material parameters as can be seen in Fig. 13.

Impact of Lighting Optimization We tested our scene augmentation pipeline with and without our lighting optimization. That is, in one case we use texture features obtained with the optimal lighting described in Section 3.2, and in the other case we use texture features obtained with uniform lighting. An example of this can be seen in Fig. 14. In this case, an orange wood on the wall of an office space was replaced by an orange metal when we did not use optimized lighting for feature extraction, and was replaced by cherry wood when we did use optimized lighting for feature extraction. While the appearance of the metal wall and cherry wood walls differ greatly under the lighting conditions of the scene, the colors of these two materials alone appear similar as can be seen in their albedo maps at the bottom of Fig. 14.

5. Conclusion

We present a novel data-augmentation pipeline that can create a large number of variants of a synthetic scene. Our central idea is to automatically replace the materials in the scene with alternate similar materials to quickly and easily create many plausible scene variations. We identify a qualitatively successful means of finding close matches between the features of source and candidate materials for replacement. Furthermore, we used OLAT renderings to identify the most discriminative lighting conditions to use to render a material, allowing us to encode as much information as

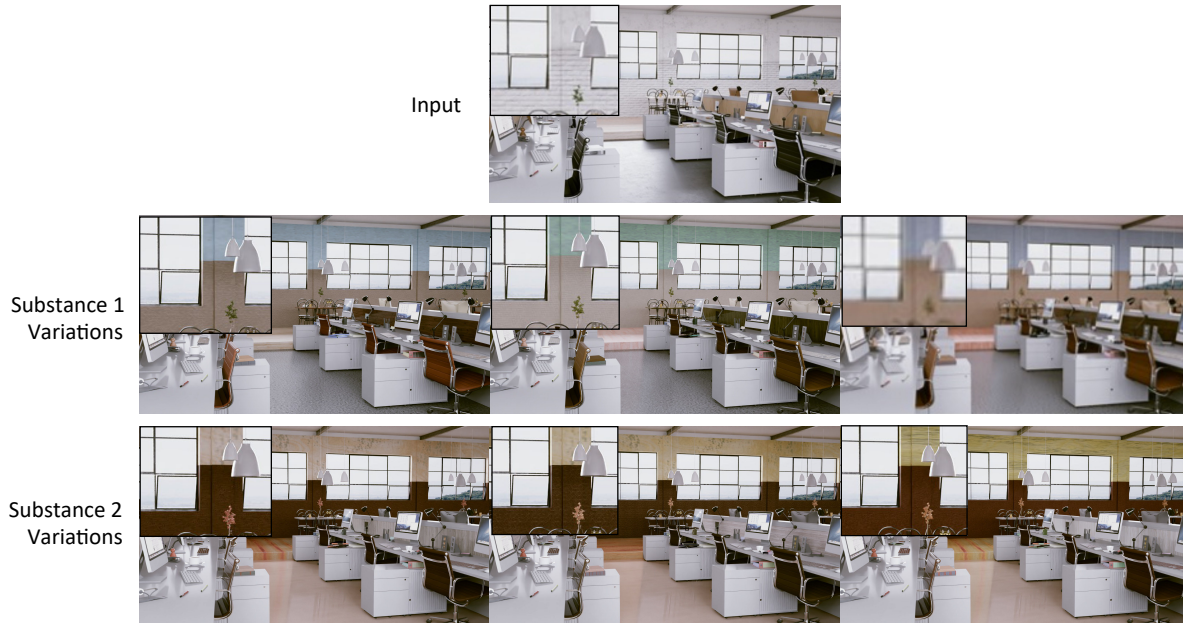


Figure 13. Pipeline-generated variants on a single input scene. Each set of variants (1, 2, 3) and (4, 5, 6) each represent three procedurally generated variants of one set of procedural material replacements.



Figure 14. Original scene (top), replacements achieved without lighting optimization (left) and replacements achieved with lighting optimization (right). Wood was replaced by a metal texture when lighting optimization was not used (bottom left) and a cherry wood grain when lighting optimization was used (bottom right).

possible about a material in a single merged render before extracting its features.

Despite the success of our method, it can only be applied to textures with low or no structure. In particular, textures strongly coupled with geometry through UV-mapping

or very structured textures such as text fonts cannot be used with our technique. An interesting extension to our work would be to focus on replacing parts of textures, alleviating this limitation. As future work, we would like to explore the use of optimized lighting in tandem with a contrastive learning framework to extend our model beyond our eight material classes. Furthermore, we hope to test the effectiveness of synthetic datasets created by our method for training on downstream tasks such as depth estimation or intrinsic decomposition.

We hope that our method can pave the way for large-scale synthetic dataset creation and help bridge the domain gap for methods that train on synthetic data.

6. Acknowledgements

The authors are grateful to the organizers and reviewers of the Women in Computer Vision workshop, and to Szymon Rusinkiewicz for his helpful feedback. Part of this work was done while Maxine Perroni-Scharf was an intern at Adobe Research.

References

- [1] Adobe Substance Dataset. <https://substance3d.adobe.com/assets>. Accessed: 2021-12-14. 4, 6
- [2] Ambient CG Website. <https://ambientcg.com/list?type=PhotoTexturePBR,AtlasPBR,DecalPBR>. [Online; accessed 20-Feb-2022]. 4

- [3] Free PBR Website. <https://freepbr.com>. [Online; accessed 20-Feb-2022]. 4
- [4] S. Basu, S. Mukhopadhyay, M. Karki, R. DiBiano, S. Ganguly, R. Nemani, and S. Gayaka. Deep neural networks for texture classification: A theoretical analysis. *Neural Networks*, 97:173–182, 2018. 2
- [5] I. Commission et al. IEC 61966-2-1:1999 Multimedia Systems and Equipment – Colour Measurement and Management – Part 2-1: Colour management – Default RGB colour space – sRGB. Standard, 1999. 4
- [6] B. O. Community. *Blender: A 3D Modelling and Rendering Package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 3
- [7] R. W. Fleming, R. O. Dror, and E. H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of vision*, 3(5):3–3, 2003. 2
- [8] I. Gkioulekas, B. Walter, E. H. Adelson, K. Bala, and T. Zickler. On the appearance of translucent edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5528–5536, 2015. 2
- [9] D. Guo, Y. Kim, and A. M. Rush. Sequence-level mixed sample data augmentation. *arXiv preprint arXiv:2011.09039*, 2020. 3
- [10] Y. Guo, C. Smith, M. Hašan, K. Sunkavalli, and S. Zhao. Materialgan: Reflectance capture using a generative svbrdf model. 39(6), Nov. 2020. 2
- [11] V. Havran, J. Filip, and K. Myszkowski. Perceptually Motivated BRDF Comparison Using Single Image. In *Computer Graphics Forum*. Wiley Online Library, 2016. 2
- [12] L. Huynh, B. Kishore, and P. Debevec. A new dimension in testimony: Relighting video with reflectance field exemplars. *arXiv preprint arXiv:2104.02773*, 2021. 2
- [13] H. Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 3
- [14] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara. Style augmentation: Data augmentation via style randomization. In *CVPR Workshops*, pages 83–92, 2019. 3
- [15] H. Jun, R. Child, M. Chen, J. Schulman, A. Ramesh, A. Radford, and I. Sutskever. Distribution augmentation for generative modeling. In *International Conference on Machine Learning*, pages 5006–5019. PMLR, 2020. 3
- [16] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012. 2
- [18] M. Lagunas, S. Malpica, A. Serrano, E. Garces, D. Gutierrez, and B. Masia. A similarity measure for material appearance. *arXiv preprint arXiv:1905.01562*, 2019. 2
- [19] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional tex-tons. *International Journal of Computer Vision*, 43(1):29–44, 2001. 2
- [20] W. Matusik. *A Data-driven Reflectance Model*. PhD thesis, Massachusetts Institute of Technology, 2003. 2
- [21] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *arXiv preprint arXiv:2110.02210*, 2021. 3
- [22] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 55–64, 2000. 2
- [23] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 3
- [24] G. Schwartz and K. Nishino. Recognizing material properties from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1981–1995, 2019. 2
- [25] A. Serrano, B. Chen, C. Wang, M. Piovarči, H.-P. Seidel, P. Didyk, and K. Myszkowski. The effect of shape and illumination on material perception: Model and applications. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021. 2
- [26] L. Shi, B. Li, M. Hašan, K. Sunkavalli, T. Boubekeur, R. Mech, and W. Matusik. Match: Differentiable material graphs for procedural material capture. *ACM Trans. Graph.*, 39(6), Nov. 2020. 2
- [27] C. Shorten and T. M. Khoshgoufar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. 2
- [28] P. Simon and V. Uma. Deep learning based feature extraction for texture classification. *Procedia Computer Science*, 171:1680–1687, 2020. 2
- [29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2, 3
- [30] T. Sun, J. T. Barron, Y.-T. Tsai, Z. Xu, X. Yu, G. Fyffe, C. Rhemann, J. Busch, P. E. Debevec, and R. Ramamoorthi. Single image portrait relighting. *ACM Trans. Graph.*, 38(4):79–1, 2019. 2
- [31] F. H. C. Tivive and A. Bouzerdoum. Texture classification using convolutional neural networks. In *TENCON 2006 – 2006 IEEE Region 10 Conference*, pages 1–4, 2006. 2
- [32] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017. 3
- [33] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Bochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018. 1
- [34] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning research*, 9(11), 2008. 5
- [35] P. Vangorp, J. Laurijssen, and P. Dutré. The influence of shape on the perception of material reflectance. In *ACM SIG-GRAPH 2007 Papers*, pages 77–es. 2007. 2
- [36] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1):61–81, 2005. 2

- [37] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. [2](#)
- [38] Z. Xu, K. Sunkavalli, S. Hadap, and R. Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. [2](#), [4](#)
- [39] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. [2](#)