Interactive Color Theme Editing System for Interior Design

Hai Yan1,*, Bin Yang1, Xueming Li1, Hongyu Wu2 and Qiang Fu1

1School of Digital Media and Design Arts, Beijing University of Posts and Telecommunications
2School of Computer Science and Engineering, Beihang University
E-mail: *yanhai@bupt.edu.cn

Abstract. In this paper, we develop a simple yet effective system for interactive indoor scene color theme editing. Our system permits the user to synthesize material/texture with user-specified colors and attaching the synthesized results to the given 3D indoor scenes. Two main improvements are made in this paper. Firstly, we give a new strategy to decompose texture to color, grey texture image and material parameters thus enable the user editing on certain of these items. Secondly, our system is a plus-in unit that integrated into Unity3D. Thus a quick preview of the color theme edited by the user can be rendered in real time. Experiments show that our system can assist the user to create an interior color theme efficiently. Moreover, our system is simple and easy to use, and has good runtime performance.

1. Introduction
Color and style harmonies are important to interior design. Once the 3D model of an indoor scene created, the designer still has to spend lots of time refining the color, texture and material of objects and even their components to ensure the color and style harmonies of the whole scene. To promote the efficiency of this task, several works have paid their attention to example-based approaches that leverage the co-exist relations of colors/materials between different categories of indoor objects, to facilitate the automatic interior color theme suggestion [1, 2].

On the other hand, interior designers always prefer interactive color theme editing rather than the full-automatic solution. This is mainly because the designers either have their preferred color theme or should consider their customer needs. These currently proposed automatic interior color theme suggestion approaches, however, focus more on the variations of their design results. Moreover, as the datasets of colors/textures adopted by these methods have no capability of self-growth, color themes generated by these methods might still need further manual manipulations to added some user-preferred textures which do not exist in their datasets to complete the design. Therefore, developing interactive systems that allow users to edit textures and attach them to the indoor object in real time is meaningful.

In this paper, we develop an interactive system to assist human designers to quickly edit the interior color theme of a given room. As shown in Figure 1, our system consists of two functionalities: a texture synthesis user interface for users to select color, grey texture image and material parameters to synthesize a colored texture map; a user interface based on Unity3D to edit the color theme of the given room, via replacing the existing textures of indoor objects and their components to these created by the users. For the first function, we observe that
Figure 1. The user can specify certain colors, material parameters, and grey texture maps to synthesize new data, and attach them to the components of the given indoor scene for real-time visualization, through the interactive framework of our system.

directly modifying the color of an existing texture map might lead to color distortion which is always caused by the gaps between the original color of the texture and the user-specified color. To address this issue, we decompose a texture map to a grey texture image and the main color, and we also consider the associated material parameters thus to make the generated texture maps can be directly attached to the objects to support the real-time 3D indoor scene design. The second function relies on Unity3D in which an input 3D indoor scene is presented and allows the users to modify the existing textures as well as the associated materials in real time. More specifically, the given indoor scene has been separated into components with textures, a clustering algorithm is employed to gather the components with similar textures/materials, thus to ensure the user editing can be quickly applied to all related textures. Thanks to the real-time rendering of the Unity3D platform, the edited results can be immediately presented to the designers. In this way, we can improve the efficiency of the interior color theme design.

To sum up, we propose an easy-to-use system with the following contributions:

1) An end-to-end system that supports the real-time interior color theme design, via editing the textures/materials of the given 3D scene;

2) A synthesis method that assist the color theme design with selectable colors, textures and material parameters.

We provide a series of experiments to demonstrate the effectiveness and user-friendliness of our system in interior color theme design, including several editing results, and user studies to compare the productiveness between our system and Adobe 3Dmax which is a common commercial software used in interior design.

2. Related Work

As an assistance tool for indoor scene design, our system is able to cooperate with most of the indoor scene synthesis methods. Many approaches have been proposed to create plausible layouts for indoor scene synthesis in the past two decades. For example, some works aimed at summarizing the structure or object relationship of indoor scenes for furniture-level objects [3, 4, 5], while others focused on the small object arrangement to facilitate indoor scene creation [6, 7, 8]. The relations between human and indoor objects also have been studied as priors for determining the indoor layouts (e.g., [9, 10, 11]). More and more recent works have paid their attention to the deep network for indoor scene synthesis with the models trained by large-scale indoor scene datasets (e.g., [12, 13]).

The other important task for interior design is to explore proper materials/textures to indoor objects, which always requires lots of manual operations. Recently several works have been proposed to assist the materials/colors suggestion to relieve the workload (e.g., [1, 2, 14]). On the other hand, some works have also studied how to label, recognize, and even synthesize
Figure 2. Two user interfaces of our system.

![User Interfaces](image)

materials to facilitate the interior color theme design (e.g., [15, 16, 17]). Despite the above-mentioned material suggestion methods that can promote the interior color theme design, the outputs of these methods however still need further rendering via some commercial tools. Our system, on the contrary, is directly integrated into the Unity3D thus to make the designed color theme can be previewed in real time.

Among various important factors that impact the quality of the designed indoor color themes, color harmonic which means are sets of colors that are aesthetically pleasing in terms of human visual perception, has been studied for a long time. For example, Cohen-or et al. [18] presented a method that enhances the harmony among the colors of a given photograph or of a general image, while remaining faithful, as much as possible, to the original colors. Odonovan et al. [19] studied color compatibility theories using large datasets, and develops new tools for choosing colors. Lu et al. [20] proposed a hierarchical unsupervised learning approach to learn the compatible color combinations from a large dataset. Kim et al. [21] presented a novel method for automatic color assignment based on theories of color perception. Our method does not directly tackle this problem. Instead, it provides a user interface to allow the user to edit the textures with proper colors in order to design a proper color theme for the given 3D indoor scene.

3. System Overview

As illustrated in Figure 2, our system provides two user interfaces. Both of them are established on Unity3D, where the given 3D scene is presented to the user with editable indoor objects and their components. Once an object/component is selected by the user, a panel of candidate texture maps will appear to assist the user change the original color theme of the given room (Figure 2-(a)). The texture maps can be synthesized through the texture editing UI as shown in Figure 2-(b). This user interface aims at assisting the user to create textures with user-specified colors and extending the dataset of texture maps.

These two user interfaces work independently. That means the user can perform an off-line mission that creates various texture maps to enrich the system dataset. Note that this operation does not need to have a 3D indoor scene input. Similarly, when the user designs the color theme of a given room through our major UI, s/he does not have to use the texture editing UI, unless none of the candidate texture maps in the dataset could meet the demands of the user. Besides the easy-to-use system interface, we also develop a workflow to deal with texture synthesis. Aiming at reusing the existing textures, our system decomposes a texture map into its main color and grey details (e.g., pattern/line). Since we also consider the material parameters in the combinations, a small number of grey textures and material types would produce various texture maps via the synthesis workflow of our system. This purpose is achieved by converting the texture image to HSB (Hue, Saturation, Brightness) color space. After that, we perform a
Figure 3. (a): Examples of the grey texture images. (b): Colorized texture maps with user-specified colors for the associated texture images.

4. Texture Synthesis

As an independent part of our system, the texture synthesis function aims at extending the datasets of texture maps with user-expected samples. This is achieved by decomposing the texture map to the main color, grey texture image and the associated material parameters (e.g., Figure 3-(a)). As shown in Figure 3-(b), we could extend the existing textures to generate more color variations through texture synthesis. The user interface has a color picker to assist the users to select a certain color as the main color of the synthesized texture. Users can first adjust a bar to determine the hue of the color, and then select a certain color from the window where gradient colors are presented with different saturation and brightness. Next to the color picker, there is a list of grey texture images that can be selected by the users. The synthesized texture map will be shown below the color picker, once a color and a grey texture image are selected. After that, the users should choose a type of material from a drop-down menu of the UI. The associated material parameters are pre-defined with a semantic label (e.g., glass, wood, etc.) to help the user-specified the proper material parameters to the synthesized texture map.

Since our goal in this step is to generate a new texture with the hue of the user-specified color and details (e.g., pattern/line) of the original texture image. Both the texture image and the user-specified color are converted to HSB color space, and then be mixed together in terms of each pixel. Let $H_c$, $H_t$ be the values of the hue of the selected color and a certain pixel of the texture image, respectively. Similarly, the values of their saturation and brightness can be denoted as $S_c$, $S_t$, $B_c$, and $B_t$. Assuming that $H_m$, $S_m$, $B_m$ are the HSB values of the associated pixel in the synthesized texture map, these values can be calculated with the following equation:

$$
H_m = H_c, \\
S_m = \omega_1 \cdot S_t + (1 - \omega_1) \cdot S_c, \\
B_m = \omega_2 \cdot B_t + (1 - \omega_2) \cdot B_c,
$$

where $\omega_1$ and $\omega_2$ are the weights (we set $\omega_1 = 0.5$ and $\omega_2 = 0.5$ in our implementation).

The synthesized texture maps still need specified material parameters in order to support the real-time indoor scene rendering. Therefore, our UI also has a drop-down menu with some
material semantic labels. In our implementation, we consider 13 kinds of materials including wood, leather, tile, etc. All these labels are pre-specified with a series of parameters such as albedo, shininess, reflectivity, etc. In this manner, once a material label is specified to a synthesized texture map, we can directly use such data to the model of the 3D scene for visualization. Note that our system also allows the user to create solid color textures with no pattern/line.

5. Color Theme Editing

The created texture maps still need a system to be attached to the indoor objects or their components, and show the effects in real time as well. To this end, the major UI of our system is constructed based on Unity3D to tackle the given 3D indoor scene. The 3D indoor scene model needs a pre-processing to enable the color theme editing and increase the quality of its real-time rendering views. First, we encode the material parameters as the features, thus the components with the same textures/materials can be clustered into groups. In this way, the components with the same texture can be jointly edited, thus to improve the efficiency of the color theme design. When the designer wants to change the texture or color of a certain object component, s/he can just click it and call a menu with a list of candidate texture maps (with specified material parameters). The design can then select one of them to replace the original texture and material of the component, and immediately see the result thanks to the real-time rendering of Unity3D.

Unity3D has some strategies to support real-time rendering such as light baking. After input a 3D scene model into Unity3D, we should first set the lighting environment for better rendering quality. Light baking is then auto-performed to create the lightmaps to decrease the computational cost. Even though light baking should be performed for each time texture map changed, the efficient baking algorithm of Unity3D could ensure that the rendering is performed in real time. Moreover, since the 3D scene is rendered in real time, the designer can observe the current color theme of the scene in a free-form view, and detect which part should change its texture map to refine the color theme. As an interactive system, all the above-mentioned stuff focus on assisting the designer to see and change the textures/materials within a 3D scene. In a word, aiming at providing a toolkit for the designer to fulfill their work, our system concerns more about how to simplify the workflow and required user operations, rather than automatically generating color themes which are various, but can hardly be edited by the human designers.

6. Experiment and Discussion

In this section, we first show some experimental results with designed indoor scenes whose color themes are created by our system. Then we compare the time cost between our system and the commercial software, i.e., Adobe 3Dmax, which is popular in interior design.

In Figure 4-Left, we show a 3D indoor scene with four different color themes designed by our system. In each row, we show the snapshots with the same view conditions but different color themes (from left to right). We also show the snapshots of this scene with different views in different rows to better illustrate the effect of the color theme editing. Note that these snapshots are rendered in real time based on Unity3D, namely, the designer can immediately view the result once the texture map of a certain object/component is changed. On average, our system took about 30 seconds to synthesize a texture map, and about 20 minutes to complete the color theme design for middle size indoor scenes with 20-30 different objects, on a PC with Intel Core i5-8400 2.80GHz CPU, 16GB RAM.

To demonstrate the usability and efficiency of our system in interior color theme design, we conducted a user study to compare the productiveness of interior design in terms of color theme determination between our system and Adobe 3Dmax. A skilled designer who is proficient in 3D modeling software is invited in this user studies. We asked the participant to design 5
candidate color themes for the given room without using any software. After that, the participant respectively was asked to use our system and 3Dmax to change the associated textures/materials following the designed color theme. The time costs of generating indoor scenes with designed color themes between the two systems are summarized in Figure 4-Right. We can see that our system can help the designer to change the color theme of a scene faster.

**Limitations.** Our current system still has several technical limitations. First, since our system tackles the solo components with materials and textures, over-segmented with too many components might slow down the pre-processing of our system. Therefore, we would ignore the tiny components in our implementation. Second, the main purpose of our system is to assist the user to quickly synthesize the required textures with user-specified colors and material parameters, we do not consider any priors to ensure the harmony of the designed color theme. Some related works [1, 2] can be employed to suggest harmonious colors based on the user inputs. These works would benefit our system and improve the efficiency of the interactive interior color theme design. Lastly, due to the limited quality of the real-time rendering technique of Unity3D, the presented indoor scenes might have chromatic aberration rather than what they should be in the real world. For this reason, after the color theme is designed through our system, using off-line rendering software such as 3Dmax to check the design results are still meaningful.

7. **Conclusion**

In this paper, we propose a novel interactive system for interior color theme editing. The proposed system provides a texture editing UI to allow the user to synthesize texture maps with user-specified color, grey texture image, and material parameters, and a major UI that presents a 3D scene whose object/component textures can be easily changed and rendered in real time. We demonstrate that our system can promote the productiveness of interior color theme design, through several experiments and user studies that compares our method with the commercial interior design software.

In the future, we interest in extending our system to more factors about the color theme. For example, the color style would be further considered when attaching the created textured map to the indoor objects/components. Namely, when the user edits the texture of a Chinese chair in ancient style, the geometry of the chair could be used as a cue to determine the proper texture the user should choose. Besides, the objects with style-related textures can be jointly edited to improve the design productiveness. We believe that interactive systems for the interior color theme would open a door to intelligent interior design, and release the designers from the heavy workload.

---

**Figure 4.** Left: Gallery of the indoor scene with different color themes edited by our system. Right: Comparison results of the user study.
Acknowledgements

We thank the anonymous reviewers for the constructive comments. This work was partially supported by grants from the NSFC (No.61902032, No.61902014), the National Key Research and Development Program of China (2018YFC0831003), the Open Project Program of the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (VRLAB2019B01), and the Fundamental Research Funds for the Central Universities (2020RC17).

References

[1] K. Chen, K. Xu, Y. Yu, T. Wang, and S. Hu. Magic decorator: automatic material suggestion for indoor digital scenes. ACM Transactions on Graphics, 34(6):232, 2015.
[2] J. Zhu, Y. Guo, and H. Ma. A data-driven approach for furniture and indoor scene colorization. IEEE Transactions on Visualization and Computer Graphics, 24(9):2473–2486, 2018.
[3] M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. ACM transactions on graphics (TOG), 30(4):34, 2011.
[4] K. Xu, R. Ma, H. Zhang, C. Zhu, A. Shamir, D. Cohen-Or, and H. Huang. Organizing heterogeneous scene collections through contextual focal points. ACM Transactions on Graphics (TOG), 33(4):1–12, 2014.
[5] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. ACM Transactions on Graphics (TOG), 36(6):201, 2017.
[6] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, and H. Zhang. Action-driven 3d indoor scene evolution. ACM Transactions on Graphics (TOG), 35(6):173–1, 2016.
[7] L. Majerowicz, A. Shamir, A. Sheffer, and H. H. Hoos. Filling your shelves: Synthesizing diverse style-preserving artifact arrangements. IEEE TVCG, 22(2):1138–1148, 2016.
[8] L.-F. Yu, S.-K. Yeung, and D. Terzopoulos. The clutterpalette: An interactive tool for detailing indoor scenes. IEEE TVCG, 20(11):1507–1518, 2014.
[9] M. Fisher, M. Savva, Y. Li, and P. Hanrahan. Activity-centric scene synthesis for functional 3d scene modeling. ACM Transactions on Graphics (TOG), 34(6):179, 2015.
[10] M. Savva, A. X. Chang, P. Hanrahan, and M. Fisher. Pigraphs: learning interaction snapshots from observations. ACM Transactions on Graphics (TOG), 35(4):139, 2016.
[11] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In CVPR’18, pp. 5899–5908, Salt Lake City, U.S., 2018.
[12] K. Wang, M. Savva, A. X. Chang, and D. Ritchie. Deep convolutional priors for indoor scene synthesis. ACM Transactions on Graphics (TOG), 37(4):70, 2018.
[13] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, C. Tu, B. Chen, D. Cohen-Or, and H. Zhang. Grains: Generative recursive autoencoders for indoor scenes. ACM Transactions on Graphics (TOG), 38(2):12, 2019.
[14] M. G. Chajdas, S. Lefebvre, and M. Stamminger. Assisted texture assignment. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 173–179, Washington, D.C., U.S., 2010.
[15] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: a richly annotated catalog of surface appearance. Acm Transactions on Graphics, 32(4):111, 2013.
[16] S. Bell, P. Upchurch, N. Snavely, and K.Bala. Material recognition in the wild with the materials in context database. In CVPR, pp. 3479–3487, Boston, U.S., 2015.
[17] K. Zsolnai fer, P. Wonka, and M. Wimmer. Gaussian material synthesis. ACM Transactions on Graphics, 37(4):76, 2018.
[18] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu. Color harmonization. ACM Transactions on Graphics, 37(4):76, 2018.
[19] P. O’donovan, A. Agarwala, and A. Hertzmann. Color compatibility from large datasets. ACM Transactions on Graphics, 30(4):63, 2011.
[20] P. Lu, Z. Kuang, X. Peng, and R. Li. Discovering harmony: A hierarchical colour harmony model for aesthetics assessment. In ACCV, pp. 452–467, Singapore, 2014.
[21] H. Kim, M. Yoo, H. Kang, and I. Lee. Perceptually-based color assignment. Computer Graphics Forum, 33(7):309–318, 2014.