# Interactive Material Annotation on 3D Scanned Models leveraging Color-Material Correlation

Wataru Kawabe
The University of Tokyo
Japan
wkawabe@iis.u-tokyo.ac.jp

Taisuke Hashimoto
Preferred Networks
Japan
hashimotot@preferred.jp

Fabrice Matulic
Preferred Networks
Japan
fmatulic@preferred.jp

Takeo Igarashi
The University of Tokyo
Japan
takeo@acm.org

Keita Higuchi
Preferred Networks
Japan
khiguchi@preferred.jp

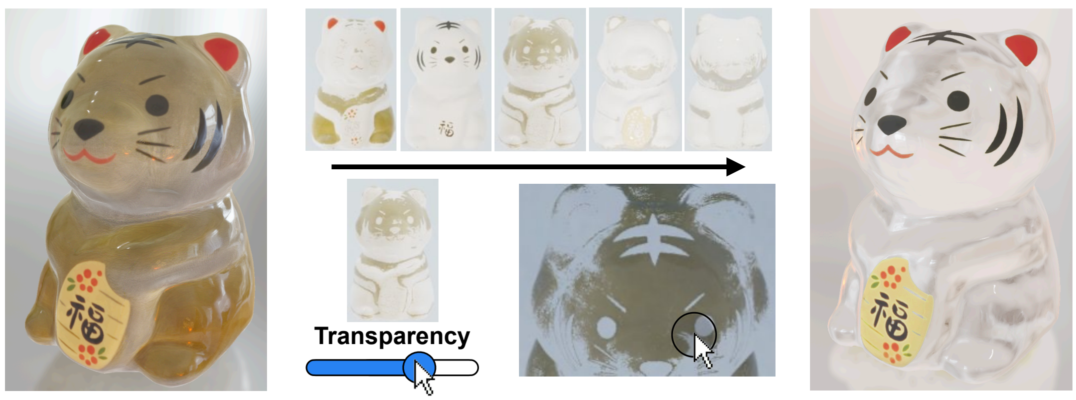


**Figure 1: The purpose of the proposed system is to make the material annotation process to 3D models more efficient and accurate. To achieve the goal, we leveraged the tendency that the distribution of colors is closely related to that of materials and introduced two approaches based on it.**

## ABSTRACT

3D scanning has made it possible to generate 3D models from real objects. Although 3D scanning can capture an object's shape and color texture, it is still technically difficult to analyze and reproduce material properties such as metalness, roughness, and transparency. Therefore, they need to be explicitly annotated after the scanning process. However, existing methods are highly labor-intensive such as a simple brush painting that requires delicate and inefficient handwork. To make this process more efficient and accurate, we propose a system that mitigates the costs by introducing a texture-aware annotation pipeline. This method is based on the observation that material distribution is correlated to color distribution. We segment the 3D surface into areas based on color similarity and let users annotate materials using the segmentations as masks. In an empirical user study, the participants could make quality annotations in a short time.

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interaction design*; • **Information systems**;

## KEYWORDS

3D scanning, 3D material annotation, graphical user interface

**ACM Reference Format:**
Wataru Kawabe, Taisuke Hashimoto, Fabrice Matulic, Takeo Igarashi, and Keita Higuchi. 2023. Interactive Material Annotation on 3D Scanned Models leveraging Color-Material Correlation. In *SIGGRAPH Asia 2023 Technical Communications (SA Technical Communications '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3610543.3626170

## 1 INTRODUCTION

3D scanning, a technology to analyze the appearance of real-world objects and reconstruct them as 3D models, is increasing its usefulness and popularity with advances in both algorithms and hardware. The representative application of 3D scanning is, for example, rich content creation such as virtual reality, movie production, video


*SA Technical Communications '23, December 12–15, 2023, Sydney, NSW, Australia*

ACM ISBN 979-8-4007-0314-0/23/12…$15.00
https://doi.org/10.1145/3610543.3626170

games, and presentation of products on e-commerce sites. In such applications aiming at enhancing the user experience, one of the requirements is that the generated 3D model is as close to reality as possible.

Factors that determine the appearance of an object include the texture of the surface as well as the shape itself. Although accurately capturing the shape with 3D scanning has become feasible to some extent, there are still technical challenges in reproducing a realistic look of its surface caused by the light (i.e., texture). A particularly challenging part is to reproduce the material properties of the original object such as metalness, roughness, and transparency since the differences in appearance are caused by light reflection, diffusion, or refraction, and analyzing it with 3D scanning is currently tricky.

To reproduce the flow of the light, it is currently necessary to explicitly annotate parameters that define the material distribution on the captured 3D surface. One of the possible methods for this is brush painting, where users scale and rotate the model in the 3D view and coordinate the brush size to paint the desired location. The problem here is that brush painting is costly for human annotators since clearly keeping the borders with a brush requires delicate manual labor. We also cannot ignore the amount of effort involved in brushing the entire surface of the model. Another method is to use image editing software such as Photoshop [1] in order to directly paint on a texture map. However, such systems do not allow users to observe the 3D view during annotation, even though an intuitive understanding of the correspondence between texture maps and 3D surfaces is almost impossible. Finally, we can annotate materials on a 3D surface efficiently with Adobe Substance 3D Painter [2]. This system is designed for newly created, manually defined 3D models, and lacks the tools to edit post-scan 3D models and their imperfect texture maps. Because of these difficulties, material annotation without any assistance is labor-intensive, even for expert workers.

This paper introduces a user interface (UI) that mitigates such costs with efficient annotation strategies (Fig. 1). We assume that the capturing process provides geometry and color distribution (texture). We design our system based on our observation that areas sharing the same color on a 3D surface are often associated with the same material. This observation led us to introduce a method to coordinate material parameters collectively for color-sharing areas (**segment-wise annotation**). However, the distribution of materials and colors do not necessarily coincide perfectly. For such a case, we also include the brush painting method that explicitly annotates within the color-sharing area while avoiding the others (**masked paint annotation**). By combining the two methods, users can finish the entire annotation process efficiently and accurately with less effort compared to the existing unassisted painting method.

## 2 RELATED WORK

There are interactive tools or systems that assist manual editing of textures on the 3D surface. Software dealing with 3D environments, such as Unity [Haas 2014] and Blender [3], enable us to apply texture annotation on the 3D surface and change materials. [Zwicker et al. 2002] proposed a system to alter the shape and appearance of 3D point models by cleaning, texturing, sculpting, carving, filtering, and resampling. Layerpaint [Fu et al. 2010] made it possible to draw long strokes across different depth layers. These tools, systems, and methods can be used for material annotation. However, efficiently specifying only the area the user wants to annotate is not yet implemented.

The problem of material annotation can also be considered as a question of how to correctly partition a 3D surface into different meaningful parts. [Huang 2022] proposed an end-to-end framework for detecting edges on the 3D surface according to its texture. 3D surface segmentation [Boubolo et al. 2021] is a more semantic approach to achieve the same goal. If we consider an image can be pasted onto a 3D surface to represent its appearance, image segmentation methods, such as Lazy Snapping [Li et al. 2004], might also be applicable. The problem here is that the division itself does not always work appropriately and there are cases where users may want to determine the distribution of materials according to their own preferences.

## 3 PROPOSED SYSTEM

### 3.1 Overview of UI

The overview of the GUI is shown in Fig. 2. The target 3D model is shown in the upper left area. (a) Users can change the camera angle by clicking and rotating the orbit control button or zooming in and out with the mouse wheel. (b) The upper right corner shows the color and material maps. The one titled *color* indicates the UV map of base color, which is obtained in the process of 3D scanning. The three maps (*Metalness*, *Roughness*, and *Transparency*) are the material maps that are updated each time the user makes an annotation. These are grayscale images, with a pixel value of 255 representing a material value of 1.0, and 0 being a material value of 0.0.

Based on the assumption that the color distribution and the material distribution are linked to some extent, the system segments the surface into layers grouped by color and utilizes them for the upcoming process. (c) The images at the bottom of the screen in Fig. 2 are the segmentation results. Users can display a certain color layer in a 3D view by clicking one of them. While each layer is visible in the 3D view, colors belonging to the other layers are invisible. Also, multiple color layers can be displayed simultaneously by selecting them while holding down the Shift button.

(d) For the masked paint annotation, users can change the properties of the brush, specifically the material type, its value, and the brush radius. (e) By clicking the undo button, users can return the 3D model to its previous state.

### 3.2 Interaction Flow

*3.2.1 Segment-wise Annotation.* Referring to the segmentation result, users can annotate a certain material value collectively for the entire area that belongs to a single color layer. By manipulating the sliders, users can adjust the material value for the corresponding color area (Fig. 3). The aim of this interaction method is to finish a large part of the work by roughly annotating the whole surface, even with inadequate details. If users feel that the colors are not well-segmented (e.g., colors perceived to be different are included in the same layer), they further split the layer into two different ones by clicking the split button. In contrast, if the colors are too

[1]https://www.adobe.com/jp/products/photoshop.html
[2]https://www.adobe.com/jp/products/substance3d-painter.html
[3]https://www.blender.org/

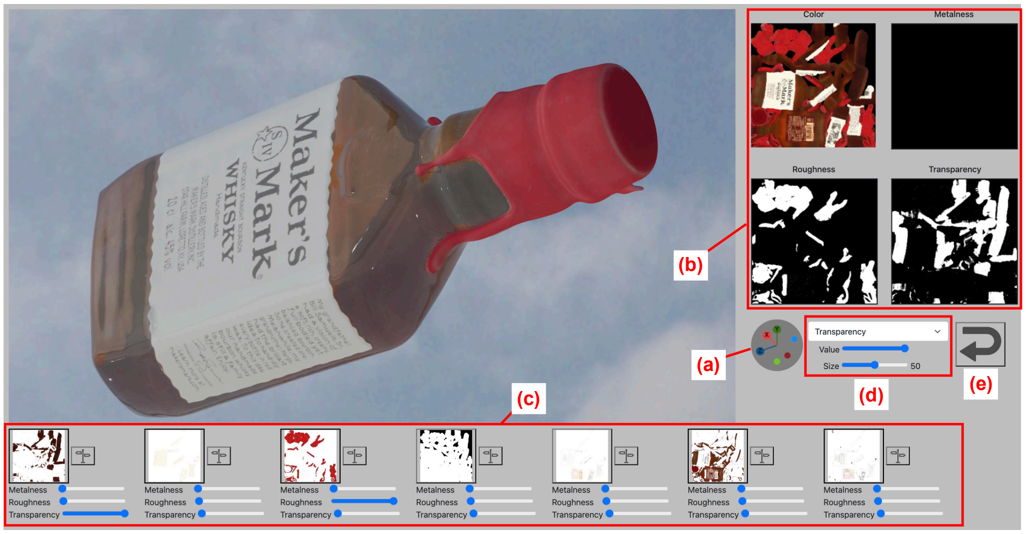


**Figure 2: The overview of our proposed system. (a) Users can change the camera angle with this button. (b) They can view the UV map and the material maps they are annotating. (c) The images are color layers indicating the color segmentation result of the UV map. Users can change the material value for each layer. They also can merge or split layer(s). (d) The brush properties can be changed here. (e) This is the undo button.**

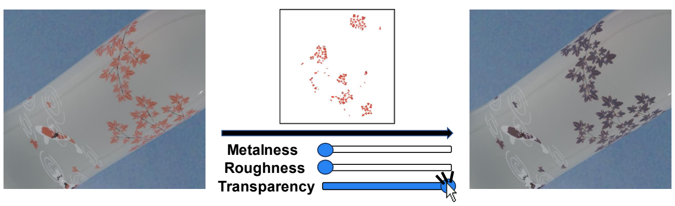


**Figure 3: The concept of the segment-wise annotation. Users can designate the material value for the whole area sharing the same color layer with the range input.**

**Figure 4: The concept of the masked paint annotation. (Left) The original 3D model. (Middle) One color layer contains two different areas: the lid and a part of the bottle. (Right) Users can modify the material values within the same color layer. For example, they can paint the bottle (circled in blue) separately.**

fragmented, they can merge different color layers by dragging and dropping one layer onto another.

*3.2.2 Masked Paint Annotation.* The masked paint annotation is used for completing the details. Specifically, this feature is used when multiple different materials are distributed within a certain color layer, or when color segmentation itself works poorly according to the effect of shadows or light reflections. By selecting certain color layer(s) and projecting it/them on the 3D view, users can additionally modify the details with brush strokes within the area while avoiding the others (Fig. 4). By intentionally masking the area of irrelevant color layers, they can make fine adjustments without paying attention to the precise borders. Users can change the size of the brush and the material value to be painted.

## 3.3 Technical Details

*3.3.1 Color Segmentation.* For color segmentation, we extract the RGB values of each pixel in the UV map, and the three-dimensional vectors were clustered with an unsupervised machine-learning method. We use K-means [Lloyd 1982] in scikit-learn [4] and set $k$ (i.e., the number of clusters) as 7, but the algorithm and the number of clusters are arbitrary. For example, it is possible to apply a deep learning-based color segmentation method, but since the UV map itself has no visual semantics as in natural images, we found the improvement in accuracy was marginal when compared with K-means. Considering that K-means works quickly with minimal computational resources unlike deep learning-based methods, we decided to use K-means for system implementation and user study as a representative of a number of candidate methods. We empirically select $k = 7$ because this number is enough to segment the colors of common objects without over- or under-segmentation. Note that future work should investigate what value of $k$ best correlates with the color and the material distributions. If users click the split button, the new color layers are calculated by K-means with $k$ as 2 within the segment.

[4] https://scikit-learn.org/

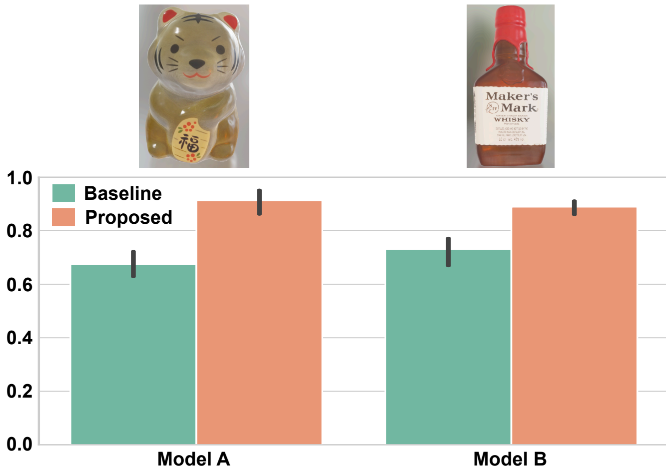


**Figure 5: The two 3D models used in the study (above) and the ratio of the correctly annotated pixels in the UV map (below). Error bars represent 95% confidence intervals.**

*3.3.2 Web App Implementation.* The 3D-related implementations are written with Threejs [5], a WebGL-based [Cozzi 2015] library for javascript. The backend is implemented with Flask [6], a Python library for web application development. In the background of the 3D model, we prepared four directional lights moving at different speeds on different trajectories in order to make it easier for users to observe the light reflection.

# 4 USER STUDY

## 4.1 Procedure

A user study was conducted to assess the efficacy of a system designed to facilitate more accurate and efficient annotation compared to a baseline system utilizing simple brush painting. The study involved 16 participants (nine women), who were all familiar with computer operations. Two 3D models were selected for the study (Fig. 5). The participants were asked to use both the proposed and baseline systems to annotate the materials of the 3D models according to a predefined target material distribution. A within-subjects method was employed, and the order of the system and model was varied among the participants to offset order and combination effects. Each trial was given a time limit of 10 minutes.

## 4.2 Results

Figure 5 indicates the ratio of correctly annotated pixels in the UV map in terms of all three materials: metalness, roughness, and transparency. We observe that significantly higher values are recorded for the proposed system for both 3D models ($p < 0.01$, Mann–Whitney U test). This indicates that our system is suitable for accurate annotation in a short time.

The notable qualitative results are shown in Fig. 6. The difficulty for model A is that the participants should avoid annotating facial parts (e.g., eyes) while annotating the underlying surface. With the baseline system, the participant had difficulty in accurately painting between the parts. The proposed system enables painting while keeping the boundaries clear by mainly utilizing the segment-wise method. The spots on the bottom of model A are almost impossible to annotate precisely within ten minutes only with a simple brush. We observe that the quality of capturing the outline of the spots is much better with the proposed system. A challenging part of model B is the lid, which requires painting the whole area from different angles and being aware of the border with the bottle. The participant failed to annotate it while keeping the border with the baseline system. With the proposed system, the entire lid can be annotated with the segment-wise annotation and the boundary can be accurately painted with the masked-paint annotation.

[5]https://threejs.org/
[6]https://flask.palletsprojects.com/en/3.0.x/

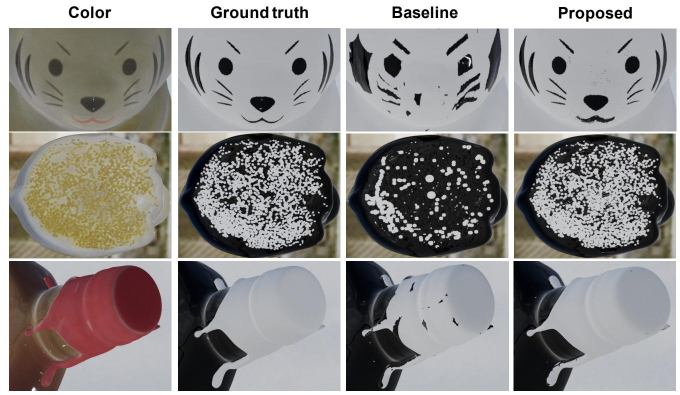


**Figure 6: Qualitative examples. Note that the material values are converted to greyscale colors to visualize them clearly.**

# 5 CONCLUSION

We proposed a system that facilitates fast and accurate material annotation on the surface of 3D scanned models. Our user study revealed that the system overcomes the shortcomings of a simple brush painting, which is one of the conventional annotation methods. As a secondary effect, participants could work with less mental strain with the proposed system. This paper would pave the way for better material annotation tools/methods of 3D-scanned models.

# REFERENCES

Louis Boubolo, Maxime Dumont, Serge Brosset, Jonas Bianchi, Antonio Ruellas, Marcela Gurgel, Camila Massaro, Aron Aliaga Del Castillo, Marcos Ioshida, Marilia Yatabe, et al. 2021. Flyby CNN: a 3D surface segmentation framework. In *Medical Imaging 2021: Image Processing*, Vol. 11596. SPIE, 627–632.

Patrick Cozzi. 2015. *WebGL Insights*. CRC Press. http://www.webglinsights.com/.

Chi-Wing Fu, Jiazhi Xia, and Ying He. 2010. Layerpaint: a multi-layer interactive 3D painting interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 811–820.

John K Haas. 2014. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE* 483 (2014), 484.

Yi-Jheng Huang. 2022. Detecting color boundaries on 3D surfaces by applying edge-detection image filters on a quad-remeshing. *Computer Animation and Virtual Worlds* (2022), e2051.

Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. 2004. Lazy snapping. *ACM Transactions on Graphics* 23, 3 (2004), 303–308.

Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.

Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. 2002. Pointshop 3D: An interactive system for point-based surface editing. *ACM Transactions on Graphics* 21, 3 (2002), 322–329.