Pacific Graphics (2023)
R. Chaine, Z. Deng, and M. H. Kim (Editors)

Short Paper

Supplemental document for "Text2Mat: Generating Materials from Text"

Zhen He[1] Jie Guo[1] Yan Zhang[1] Qinghao Tu[1] Mufan Chen[1] Yanwen Guo[1] Pengyu Wang[2] and Wei Dai[2]

[1]Nanjing University, State Key Lab for Novel Software Technology, China
[2]Dimension 5 Techs, China

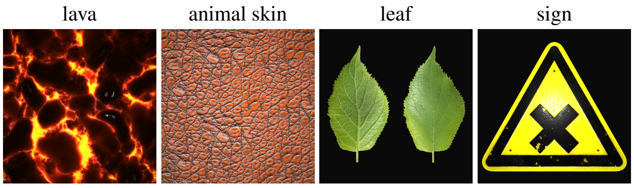

Figure 1: *Some uncommon and unreasonable materials which are not considered in our dataset.*

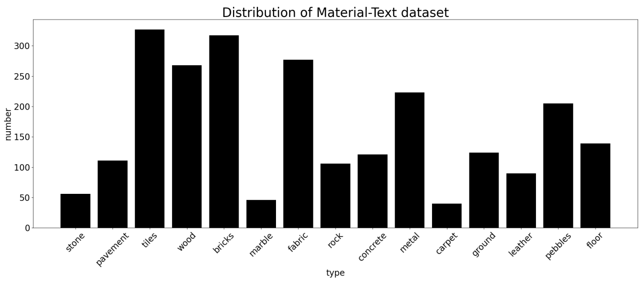

Figure 2: *The histogram of our dataset distribution*

1. More Details of Our Material-text Dataset

We collected PBR mateirals from four 3D asset sites[Tex32; Dem23; Hav23; Sha23]. In total, over 4000 different material samples with CC0 Liscene were collected, each containing corresponding SVBRDF parameter maps and a corresponding type label, with 1K (1024) resolution for the samples collected from 3D Textures[Tex32] and 4K (4096) resolution for the samples collected from the other three sites. These 4000+ textures include a wide variety of materials. We filtered out materials that were not common in life or had unreasonable content as listed in Fig. 1, such as laval, animal skin etc.

As is shown in Fig. 2, our dataset mainly consists of 15 types of materials, they are stone, pavement, tiles, wood, bricks, marble, fabric, rock, concrete, metal, carpet, ground, leather, pebbles, floor.

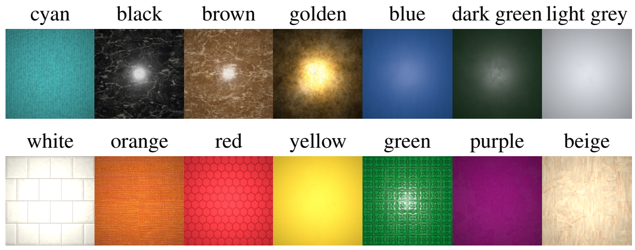

Figure 3: *Examples of rendering results with materials of different basic colors in our dataset.*

1.1. Color Label

When it comes to color label, we do not intend to use the very complex web color standard or other standards for the definition of color attributes, as the semantic definition of a color is a subjective human perception, and very slight color changes are difficult to distinguish in simple language, and the perception of light varies between people's retinas. Therefore, after examining all the material data collected and referring to the existing color description tags in some of the data, in order to simplify the complex color representation and to be able to distinguish between colors, we have chosen some basic colors[BK91] that are commonly known to people (shown in Fig. 3), specifically red, orange, yellow, green, cyan, blue, purple, pink, brown, grey, black, white, golden and beige. Different colors can be combined, such as blue grey, grey white, etc. For the same color we use light and dark to distinguish. Finally, Each material sample is labelled by hand with the appropriate color.

1.2. Texture Label

We make use of the attributes of DTD[CMK*14] dataset to help us describing texture features of material, but we observe that DTD textures are biased toward describing unstructured texture features. To solve this problem, we introduce 9 new attributes. Fig. 4 shows specific attributes describing the materials with structured and regular textures in our dataset, including herringbone, I-shaped, staggered, rectangular, cubic, diamond, polygonal, triangular and

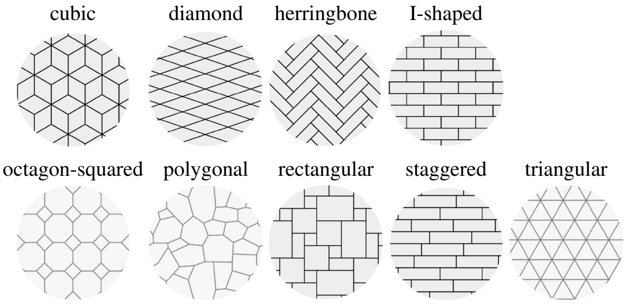

Figure 4: *Our texture attribute descriptions for regular materials.*

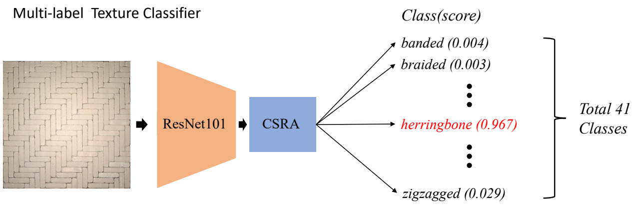

Figure 5: *Example of Multi-label texture classification: a beige pavement is classified to label "herringbone".*

octagon-squared. In most cases, these attributes are applied on materials such as tiles, bricks, pavements, fabric, etc.

1.3. Multi-label Classifier

After introducing the new attributes, we identified a total of 41 texture category labels (listed in Tab. 1) and processed the DTD dataset against these labels (filtering, internet collection expansion). Then, We trained a classifier (displayed in Fig. 5) consisting of a pre-trained ResNet101 [HZRS16; DDS*09] with a CSRA[ZW21] module for multi-label classification on the processed DTD dataset. When performing classification on our data, the classifier usually outputs single class label, sometimes 2 or 3. These labels are mainly auxiliary to the annotation, but generally we keep all the labels unless they do not match the appearance and geometry of the material very much.

1.4. Sample Annotation

In Fig. 6, We show some examples of text label on materials in our dataset.

2. More Results and discussion of Text2Mat

We show more comparison results used for two user studies in Sec. 5.1 and Sec. 5.2 and discuss the limitations of Text2Mat.

2.1. More Comparison Results in user studies.

More comparison results on image generation of SD[RBL*22] and Text2Mat are displayed in Fig. 7, and on material generation of Poly[Inf23] and Text2Mat in Fig. 8

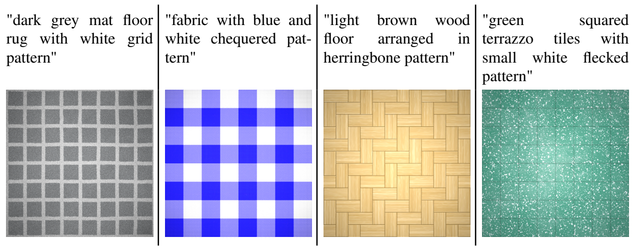

Figure 6: *The text label of materials.*

2.2. Limitation and Discussion

Despite we can generate materials matching the text input with Text2Mat through two stages, there still remain some limitations. First, Text2Mat is trained on the Text-Mateiral dataset based on the pre-trained Stable Diffusion, and the samples in this dataset are only a very small fraction compared to the dataset used in SD training, so Text2Mat is limited in the number of materials it can generate. Second, Text2Mat generates materials in two stages, where we generate representation in latent space based on the text input in stage I and directly recover the corresponding SVBRDFs from the representation in stage II. When generating metal or metal-like materials, sometimes the correct metallicity map cannot be reconstructed due to the absence of any priori information. Third, in most cases, Text2Mat can produce SVBRDF maps without burn-in artifacts, due to the strong representation ability of SD and the guidance of proper text descriptions. Burn-in artifacts do occasionally appear in some cases and removing these artifacts would be our further work. Retrieving materials from a library with a large number of node graphs can be time-consuming, and later adjustments may require additional expertise. While generating various materials directly from simple forms of text can be much more efficient, our proposed Text2Mat is able to generate a variety of results based on the same description at the same time (depending on the random seed set and the number of samples to be generated). This allows the user to pick the version of the material they really want.

References

[BK91] Berlin, Brent and Kay, Paul. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991 1.

[CMK*14] Cimpoi, Mircea, Maji, Subhransu, Kokkinos, Iasonas, et al. "Describing Textures in the Wild". *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, 3606–3613. doi: 10.1109/CVPR.2014.461 1.

[DDS*09] Deng, Jia, Dong, Wei, Socher, Richard, et al. "Imagenet: A large-scale hierarchical image database". *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, 248–255 2.

[Dem23] Demes, Lennart. *ambientCG*. https://ambientcg.com. 2023 1.

[Hav23] Haven, Poly. *Poly Haven*. https://polyhaven.com. 2023 1.

[HZRS16] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. "Deep residual learning for image recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778 2.

[Inf23] Infinity, Poly. *Poly*. https://withpoly.com/browse/textures. 2023 2, 4.

Table 1: *The total 41 texture labels used for classification.*

banded	braided	bumpy	chequered	cracked	crosshatched	cubic	diamond	dotted
grid	grooved	herringbone	honeycombed	interlaced	I-shaped	knitted	lacelike	lined
octagon-squared	paisley	perforated	pitted	pleated	polka-dotted	polygonal	porous	rectangular
spiralled	staggered	stratified	striped	studded	swirly	triangular	woven	flecked
frilly	meshed	scaly	wrinkled	zigzagged				

[RBL*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. "High-resolution image synthesis with latent diffusion models. 2022 IEEE". *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 10674–10685 2, 4.

[Sha23] SHARETEXTURES. *ShareTextures*. https://www.sharetextures.com. 2023 1.

[Tex32] TEXTURES, 3D. *3D Textures*. https://3dtextures.me. 2023.2 1.

[ZW21] ZHU, KE and WU, JIANXIN. "Residual attention: A simple but effective method for multi-label recognition". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 184–193 2.

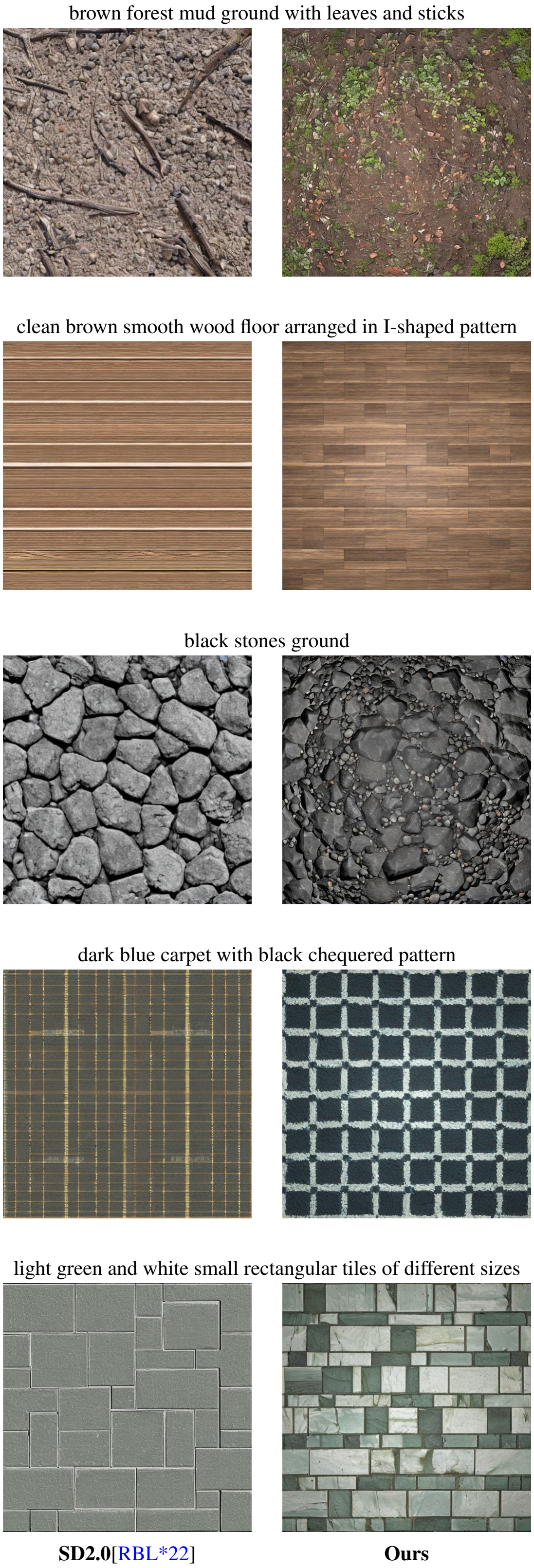

Figure 7: *Comparison to SD2.0 [RBL*22]. We show more results on text-to-image generation of SD2.0 and our Text2Mat.*

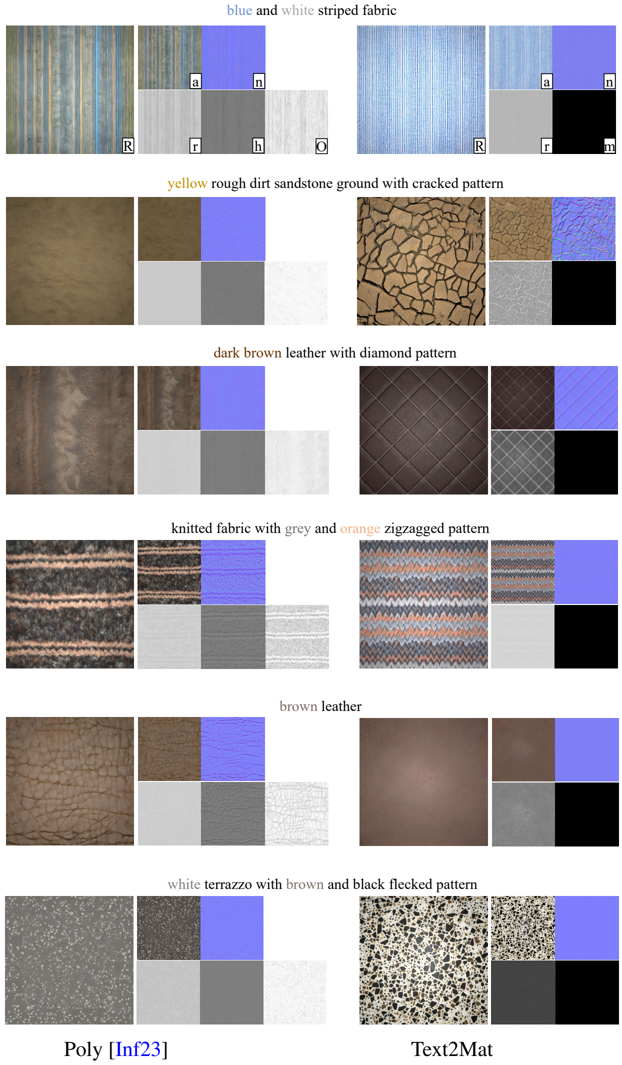

Figure 8: *Comparison to Poly [Inf23]. We show more results on text-to-material generation of Poly and our Text2Mat. The a, n, r, h, m, O, R denote albedo, normal, roughness, height, metallicity, ambient occlusion and rendering images, respectively.*