Supplementary Material for "TexGaussian: Generating High-quality PBR Material via Octree-based 3D Gaussian Splatting"

Bojun Xiong[1*†], Jialun Liu[2*], Jiakui Hu[3†], Chenming Wu[2], Jinbo Wu[2], Xing Liu[2], Chen Zhao[2], Errui Ding[2], Zhouhui Lian[1‡]

[1]Wangxuan Institute of Computer Technology, Peking University, China
[2]Baidu VIS
[3]Institute of Medical Technology, Peking University, China

1. Preliminary of 3D Gaussian Splatting

Gaussian splatting employs a collection of 3D Gaussians to represent 3D data. Specifically, each Gaussian is formally defined as:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ represents the spatial mean of 3D Gaussian and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ denotes the covariance matrix. The covariance matrix $\boldsymbol{\Sigma}$ of a 3D Gaussian is analogous to describing the configuration of an ellipsoid. Thus, the covariance matrix $\boldsymbol{\Sigma}$ is decomposed into a scaling matrix $\boldsymbol{S}$ and a rotation matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{R}^\top. \tag{2}$$

To allow independent optimization of both factors, they are stored separately: a 3D vector $\boldsymbol{s}$ for scaling and a quaternion $\boldsymbol{q}$ to represent rotation. During the rendering process, the 3D Gaussians are projected onto a 2D plane. With the intrinsic matrix $\boldsymbol{K}$ and extrinsic matrix $\boldsymbol{T}$, the 2D mean $\boldsymbol{\mu}'$ and covariance $\boldsymbol{\Sigma}'$ are defined as follows:

$$\boldsymbol{\mu}' = \boldsymbol{K}[\boldsymbol{\mu}, 1]^\top, \quad \boldsymbol{\Sigma}' = \boldsymbol{J}\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^\top\boldsymbol{J}^\top. \tag{3}$$

Here, $\boldsymbol{J}$ represents the Jacobian of the affine approximation of the projective transformation. Each 3D Gaussian is associated with an opacity value $\boldsymbol{o}$ and a view-dependent color $\boldsymbol{c}$, determined by a set of spherical harmonics coefficients. In our model, the multi-view rendered images of albedo map do not depend on the selected viewpoints. As a result, we just use three-channels RGB on each 3D Gaussian to represent the view-independent colors instead of original spherical harmonics, and we exclude the positional parameter $\boldsymbol{\mu}$ because each 3D Gaussian is fixed at the center of each finest leaf node of the constructed octree. All the parameters can be collectively denoted by Θ_0 with:

$$\Theta_{0i} = \{\boldsymbol{o_i}, \boldsymbol{s_i}, \boldsymbol{q_i}, \boldsymbol{c_i}\}, \tag{4}$$

representing the parameters for the i-th Gaussian.

Moreover, to encode the PBR material parameters, we append additional two parameters: roughness $\boldsymbol{r}$ and metallic $\boldsymbol{m}$ at the end of the original Gaussian parameters. To render multi-view images of these two attributes, we concatenate $\boldsymbol{r}$ and $\boldsymbol{m}$ with previous parameters to obtain:

$$\Theta_{1i} = \{\boldsymbol{o_i}, \boldsymbol{s_i}, \boldsymbol{q_i}, \boldsymbol{r_i}\},\ \Theta_{2i} = \{\boldsymbol{o_i}, \boldsymbol{s_i}, \boldsymbol{q_i}, \boldsymbol{m_i}\}. \tag{5}$$

Then, all the 3D Gaussians are paired with these two new parameters Θ_{1i} and Θ_{2i}, rendered from multiple viewpoints to get multi-view roughness map and metallic map for further training.

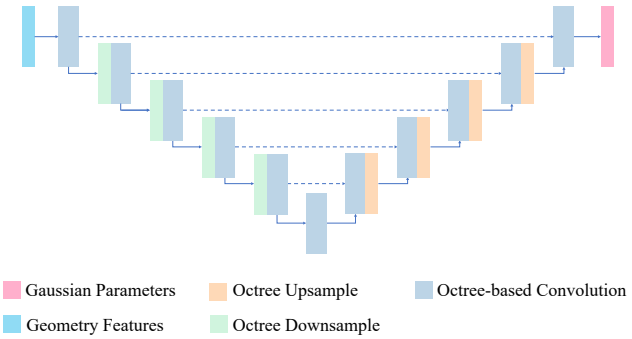

Figure 1. The network architecture of the octree-based 3D U-Net we used to train our unconditional RGB texture generation model.

*Denotes equal contribution.
†This work was partly done when Bojun and Jiakui interned in Baidu.
‡Corresponding author. E-mail: lianzhouhui@pku.edu.cn

This work was supported by the National Natural Science Foundation of China (Grant No.: 62372015), Center For Chinese Font Design and Research, and Key Laboratory of Intelligent Press Media Technology.

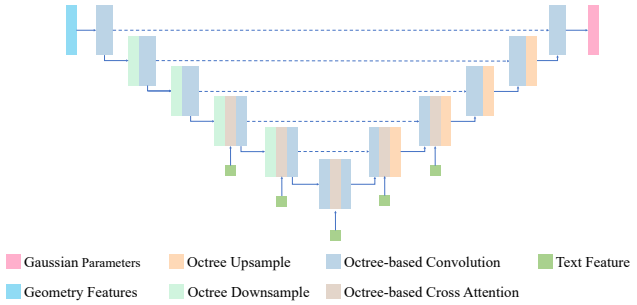

Figure 2. The network architecture of the octree-based 3D U-Net we used to train our text-conditioned PBR material generation model.

2. Network Details

Unconditional RGB Texture Generation The network architecture of the octree-based 3D U-Net we used in unconditional RGB texture generation is shown in Fig. 1. The U-Net has five hierarchical levels, corresponding to octree depths of 8, 7, 6, 5 and 4, with resolutions of $256^3, 128^3, 64^3, 32^3, 16^3$. The feature dimensions are set to $32, 64, 128, 256, 256$ respectively. The number of channels for input and output features is 4 and 11, respectively, due to the lack of material information.

Text-conditioned PBR Material Generation The network architecture of the octree-based 3D U-Net we used in text-conditioned PBR material generation is shown in Fig. 2. The U-Net has five hierarchical levels, corresponding to octree depths of 8, 7, 6, 5 and 4, with resolutions of $256^3, 128^3, 64^3, 32^3, 16^3$. The feature dimensions are set to $64, 128, 256, 512, 512$ respectively. The number of channels for input and output features is 4 and 13, respectively, as described in the main manuscript. The text feature is fed to U-Net via the octree-based multi-head cross attention mechanism. The cross attention layers are only inserted at the least two down-sampling blocks, the middle block and the two first up-sampling blocks to save GPU memory.

3. More Results

Our method is capable of generating diverse materials given different text prompts for a single mesh. Fig. 3 shows the PBR materials and the rendering results of the same mesh generated from different text prompts by our proposed TexGaussian. These results demonstrate that our method is able to generate diverse materials of different styles that align well with the text prompts and 3D objects with high fidelity.

We provide more generated results in Fig. 4.

Figure 3. Diverse material generation. Our method can generate different materials with different text prompts on the same mesh.

Figure 4. More generative results of our method on different input 3D models and text prompts.