Supplementary Material for Text2Scene: Text-driven Indoor Scene Stylization with Part-aware Details

Inwoo Hwang[1], Hyeonwoo Kim[1], and Young Min Kim[1,2,*]
[1]Department of Electrical and Computer Engineering, Seoul National University
[2]Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University

In the supplementary materials, we elaborate the implementation details for our Text2Scene (Sec. A), the impact of seed (Sec. B), motivation of global context (Sec. C) additional visual results (Sec. D, Sec. E, Sec. F) and through algorithm (Sec. G).

A. Implementation details

A.1. Network Architecture

The input to the LNSF is the positional encoding of the 3D coordinate, coefficients of the eigenfunctions of mesh, and part segment id, and the output is the color of the vertex. Each vertex $p \in \mathbb{R}^3$ is mapped to a 256-dimensional Fourier feature applying by $\gamma(p) = [\cos(2\pi Bp), \sin(2\pi Bp)]$ where B is randomly sampled from $\mathcal{N}(0, 5^2)$. Also, the coefficients of the eigenfunctions corresponding to the top 128 eigenvalues are used to reflect the intrinsic geometry. The neural network for LNSF consists of five 256-dimensional layers. For activation, $\tanh$ is used for the last layer, and ReLU is used for others. In addition, the weight of the final layer is set to zero; thus, the colors of all vertices meshes are initially set to a constant color.

For part-aware geometric deformation in Sec. 3.2.3, we branch the last layer into two while having the first four layers in common, and each last layer outputs the color and displacement respectively [3].

A.2. Training Details

For all experiments, we use the Adam optimizer [1] with an initial learning rate of 5×10^{-4} and the learning rate decay factor as 0.9 for every 100 iterations. We sample camera poses on a hemisphere with a radius r. The viewing angles are sampled from Gaussian distribution with $\sigma = \pi/4$ around the front view of objects within the elevation angle of $[10°, 80°]$ and the azimuth angle of $[0°, 360°]$. As mentioned in Sec. 3.2.3. for detailed stylization, we apply random perspective transformations and random cropping and observe $10 - 20\%$ of the original rendered image. A color range of each point is set to 0 to 1 by adding half of the output of LNSF through $\tanh$ with the gray color $[0.5, 0.5, 0.5]$.

For parameters, we set $r = 2.0$ for rendering, $\lambda_1 = 0.2$ for structure stylization and $\lambda_{th} = 3.0$ for merge segments. Also, we set λ_2, λ_3 as 0.2 for base color assignment and $\alpha = 0.3$ for detailed stylization. For all experiments, we use a pre-trained CLIP learned with a ViT-B/32 backbone [6].

A.3. Design Choice for Structure Stylization

In Sec. 3.1, we stylize structure by retrieving texture from a pre-defined texture set using MATch [7]. Here, we show undesirable artifacts when the structural components are not separately handled and were created using CLIP [6]. We generate walls directly from the designed MLP or optimize the weights of a CNN that translates a fixed random noise z to an output image [4, 8]. As shown in Fig. A.1, we observe various artifacts, such as a number of bricks or grass. On the other hand, MATch [7] successfully samples a candidate of structure components.

Figure A.1. Wall texture generation through MLP, CNN, and MATch [7]. Generation with CLIP embedding results in creating texture with multiple objects and perspective distortions, which is not appropriate for texture patterns for structural components with a flat geometry. The text 'a wall' is used for CLIP.

B. Impact of Seed, Limitation and Diversity

From the same text prompt, CLIP can generate different 2D images, and this property extends when we use CLIP embedding to discover part information or stylize 3D assets. The additional degree of freedom stems from the random

*Young Min Kim is the corresponding author.

seed, as observed with the convergence pattern of generation using CLIP. Figure B.2 shows different part discovery results from the same initial super segments. Different part segments are sometimes merged into one based on the convergence condition and might fail to discover the correct part information.

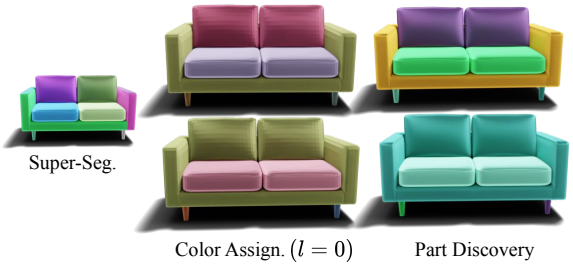

Figure B.2. From the same initial super segments, different parts can be observed through random seeds.

The different random seeds can result in diverse results after the part discovery. Figure B.3 shows the stylization results of the same input text with different seeds. Figure B.4 shows stylized scenes of different seeds with the same input target image I_t and the style description t_s.

Figure B.3. From the same text prompt, we can generate diverse objects through different random seeds. 'a book of Harry Potter and the Sorcerer's Stone' (top), and 'a sofa, minimal style' (bottom) is used for description, respectively.

C. Motivation of Using Text for the Global Context

We discuss our choice of the shared text description to enforce the clip loss in the structure stylization (Sec. 3.1) and the part-level base color assignment for the object colors (Sec. 3.2.2).

For the structure stylization, we use an additional text prompt T_s, 'a structure of a room' to provide the context with the clip loss. Figure C.5 shows randomly generated structure and their resulting clip scores against the text prompt T_s, $\mathcal{L}_{\text{clip}}(I_s, T_s)$. The samples indicate that the resulting clip scores, to some extent, reflect how natural the texture choices are. Therefore, we can exclude unnatural stylization of the structural components with the simple text.

Figure B.4. Diverse scenes could be generated through different random seeds.

Figure C.5. $\mathcal{L}_{\text{clip}}(I_s, T_s)$ with the renderings of structures from randomly sampled choices of textures.

For objects within the scene, we assign part-level base colors with the text prompt T, which represents the types of the scene, such as 'a bedroom', and an additional global signal $\mathcal{L}_{\text{clip}}(I, T)$. Figure C.6 shows the results when each object is independently stylized using only the object clip loss compared to stylization with the additional global clip loss at the base color assignment step. The color loss between the target image, $\mathcal{L}_{\text{hist}}(I, I_t)$ is not used. With the global clip loss, we learn the assignments of the scene with harmonious colors of the scene-level rendering. Also, as shown in the results of the user study in the main paper (Table. 1), we generate a more realistic texture for the holistic scene with the color histogram of the target image in addition to the global clip loss.

D. Additional Results of Part Discovery

Our realistic stylization greatly benefits from the stable part discovery to assign different textures. Figure D.7 con-

w/o global clip loss

w/ global clip loss

Figure C.6. Base color assignment results (left), and the final results with additional learned details (right). With the global clip loss, more harmonious base colors are assigned from which additional details are generated. The color histogram loss against the target image is not used.

tains intermediate results of the entire pipeline, starting from the initial super-segments followed by iterations of merged segments until convergence. Figure D.8 contains more results of the discovered parts of various objects. It shows that the results correctly capture different parts that conventionally are colored with different materials or textures.

We observed a 76.6% success rate for the part recovery without the help of any part dataset. Note that there is no public dataset available for 'texture parts', which are different from functional parts or semantic segmentation, and we evaluated with a manually annotated dataset.

E. Comparison to Other Text-to-3D Methods

Parallel to our method, several methods create 3D contents from text input. such as DreamFusion [5] and Latent-NeRF [2]. DreamFusion [5] creates 3D contents in NeRF representation. As shown in Figure E.9, the created volumetric representation tends to be blurry for highly structured objects, such as beds, while we enjoy more explicit texture boundaries. We also include a comparison against Latent-NeRF [2], which supports the mode of updating the UV-texture map of a mesh using the score distillation loss proposed by [5]. The color distribution generated with our method exhibits superior visual quality over their texture map as our results show clear boundaries obtained from the part discovery step.

Figure D.7. Intermediate results of part discovery. From the initial super-segments (left), we show segment $\{s_{ik}^l\}$ and assigned color $c(s_{ik}^l)$ at l^{th} iteration and finally show the part discovery results (right) for each row. Empirically the process converges within two iterations. Random colors are assigned to each segment.

Figure D.8. Part discovery results for diverse categories of objects. Random colors are assigned to discovered parts.

Figure E.9. Result from [5], [2], and ours with discovered part in order.

F. Additional Stylization Results

Although it was not obvious from the main manuscript, our stylized objects contain plausible texture even when observed from diverse points, as shown in Fig. G.10. The individual objects can be weakly bound by the text description

as shown in Fig. G.11.

Since we stylize the entire object, we can easily manipulate the 3D scene through object removal, replication, or relocation (Fig. G.12). Also, we show additional stylized results over various scenes (Fig. G.13), or target image and style descriptions (Fig. G.14). Finally, for the scene stylization, our target image can be replaced to natural photographs (Fig. G.15).

G. Text2Scene: Algorithm

We provide the algorithm to describe the flow of the entire pipeline in Algorithm 1.

References

[1] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[2] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 3

[3] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021. 1

[4] Christopher Olah, Ludwig Schubert, and Alexander Mordvintsev. Feature visualization. *Distill*, 2017. 1

[5] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[7] Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekeur, Radomir Mech, and Wojciech Matusik. Match: Differentiable material graphs for procedural material capture. *ACM Trans. Graph.*, 39(6):1–15, Dec. 2020. 1

[8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017. 1

Figure G.10. We stylize whole 3D objects and provide a diverse view of stylized objects.

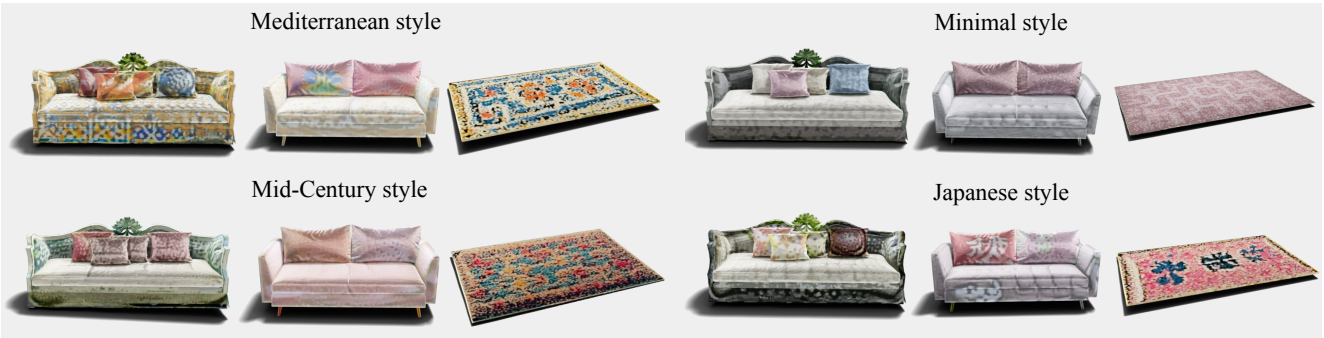

Figure G.11. Object stylization results in the specific text description.

Figure G.12. From original scene (left), we can manipulate scene by object removal, replication and relocation.

Figure G.13. Additional stylize results for various scenes. We validate our algorithms for four scenes.

Figure G.14. Results of the same scene with different stylization, using different target images and style texts.

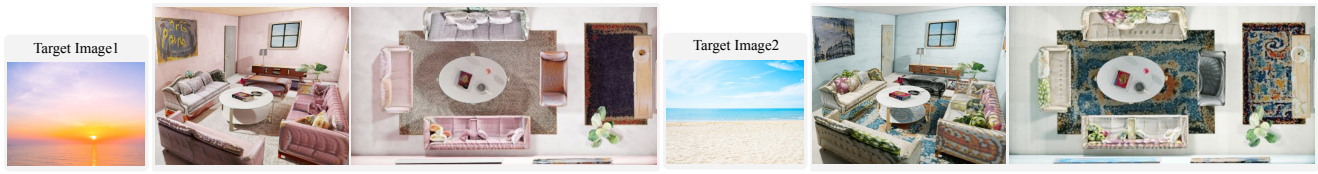

Figure G.15. Results of setting the target image as a natural photograph.

Algorithm 1 Overall pipeline of Text2Scene

Input: 3D scene $\mathcal{S} = \{\mathcal{W}, \mathcal{O}\}$ where $\mathcal{W}$ is a structure components and $\mathcal{O} = \{\mathbf{M}_i\}$ is a set of 3D mesh objects,
Corresponding class labels and optionally have text descriptions for each $\mathbf{M}_i$,
Target image $\mathbf{I}_t$, specific appearance style description $\mathbf{t}_s$

Output: Stylized 3D Scene $\mathcal{S}$ reflecting object specific information and given conditions

Structure Stylization $\mathcal{W}$ (Sec. 3.1.)

$Score = \emptyset$
for i $= 1, 2, \ldots, N$ **do**
  $\mathcal{W} \leftarrow$ `set_texture` ▷ Sample texture from texture set
  $\mathbf{I}_s \leftarrow$ `render`$(\mathcal{W})$
  $Score = Score \cup \{criteria(\mathbf{I}_s, \mathbf{I}_t, \mathbf{T}_s)\}$
end for
$[\mathcal{W}^*, \_\_] \leftarrow$ `top-1`$[Score]$

Part Discovery for each 3D object (Sec. 3.2.1.)

for i $= 1, 2, \ldots, |\mathcal{O}|$ **do**
  $\{\mathbf{s}_{ik}^0\} \leftarrow$ `superseg`$(\mathbf{M}_i)$ ▷ Initial super-segments
  $l = 0$
  while num of segments does not decrease **do**
    Set $\mathbf{c}(\mathbf{s}_{ik}^l)$ as grey
    Generate a graph $\mathcal{G}_i^l$
    for iter $= 1, 2, \ldots, L$ **do**
      $\mathbf{I}_{M_i} \leftarrow$ `render`$(\mathbf{M}_i)$
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{clip}}(\mathbf{I}_{M_i}, \mathbf{T}_{i,c})$
      Update $\mathbf{c}(\mathbf{s}_{ik}^l)$
    end for
    Update graph $\mathcal{G}_i^l$
    $\{\mathbf{s}_{ik}^{l+1}\} \leftarrow$ `merge`$(\{\mathbf{s}_{ik}^l\})$ ▷ Merge segments
    $l = l + 1$
  end while
end for
Discovered Part $\{\mathbf{s}_{ik}\}$

Part-level Base Color Assignment (Sec. 3.2.2.)

$\forall \mathbf{M}_i \in \mathcal{O}$, set $\mathbf{c}(\mathbf{s}_{ik})$ as grey
for iter $= 1, 2, \ldots, L$ **do**
  $\mathbf{I} \leftarrow$ `render`$(\mathcal{W}^*, \mathcal{O})$
  $\mathbf{I}_{M_i} \leftarrow$ `render`$(\mathbf{M}_i)$, $\forall \mathbf{M}_i \in \mathcal{O}$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{color,scene}} + \mathcal{L}_{\text{clip,scene}}$
  Update $\mathbf{c}(\mathbf{s}_{ik})$, $\forall \mathbf{M}_i \in \mathcal{O}$
end for
Part-level Assigned Color $\mathbf{c}(\mathbf{s}_{ik})$

Detailed Stylization (Sec. 3.2.3.)

for i $= 1, 2, \ldots, |\mathcal{O}|$ **do**
  for iter $= 1, 2, \ldots, L$ **do**
    $\mathbf{M}_i \leftarrow$ `update_local_texture`$(\mathcal{F}_{i,\theta}(\mathbf{M}_i))$ ▷ Local Neural Style Field (LNSF)
    $\mathbf{I}_{M_i} \leftarrow$ `render`$(\mathbf{M}_i)$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{clip}}(I_{M_i}, T_i^+)$
    Update LNSF $\mathcal{F}_i$ parameters
  end for
end for
Stylized Object $\mathbf{M}_i$