Text2Mesh: Text-Driven Neural Stylization for Meshes

Oscar Michel[1*] Roi Bar-On[1,2*] Richard Liu[1*] Sagie Benaim[2] Rana Hanocka[1]

[1]University of Chicago [2]Tel Aviv University

1. Additional Results

Please refer to our project webpage for additional results. We show multiple views of a chair mesh synthesized with a wood style in Fig. 1 to demonstrate how our textures automatically align to the shape's sharp features and curves. We also give another showing the effectiveness of our displacement loss in Fig. 2. In Fig. 3 we show the more results of removing the color prediction from training.

2. Robustness to Mesh Quality

We learn to style high-resolution meshes, and thus are able to synthesize style with high fidelity. We show in Fig. 4 a 1670x2720 render of one of the stylized outputs we show in the main paper as a demonstration.

As mentioned in Section 3.1, our method is effective even on coarse inputs, and one can always increase the resolution of a mesh M to learn a neural field with finer granularity. In Fig. 6, we upsample the mesh by inserting a degree-3 vertex in the barycenter of each triangle face of the mesh. The network is able to synthesize a finer style by leveraging these additional vertices.

Additionally, our method is able to generalize its style to points on the object surface that were not seen during training. We show an example of this in Fig. 7, where a network trained on a coarse mesh is able to infer details across a subdivided surface.

Our method also performs well on low quality meshes. We show the result of testing our method on a mesh with non-manifold edges, self-intersections and boundary loops.

3. Choice of anchor view.

As mentioned in Section 3.3 of the main text, we select the view with the highest (i.e. best) CLIP similarity to the content as the anchor. There are often many possible views that can be chosen as the anchor that will allow a high-quality stylization. We show in Fig. 8 a camel mesh where the vertices are colored according to the CLIP score of the view that passes from the vertex to the center of the mesh. The color range is shown where the minimum and maximum values in the range are 0 and 0.4, respectively. We show in Fig. 9 a view with one of the highest CLIP scores and a view with one of the lowest. The CLIP score exhibits a strong positive correlation with views that are semantically meaningful, and thus can be used for automatic anchor view selection, as described in the main paper. This metric is limited in expressiveness, however, as demonstrated by the constrained range that the scores fall within for all the views around the mesh. The highest score for any view of the camel is 0.35 whereas the lowest score is still 0.2.

As mentioned in Section 3.3, n_θ, the number of sampled views, is set to 5. We show in Fig. 10 that increasing the number of views beyond 5 does little to change the quality of the output stylization.

4. Training and Implementation Details

4.1. Network Architecture

We map a vertex $p \in \mathbb{R}^3$ to a 256-dimensional Fourier feature. Typically 5.0 is used as the standard deviation for the entries of the Gaussian matrix $\mathbf{B}$, although this can be set to the preference of the user. The shared MLP layers N_s consist of four 256-dimensional linear layers with ReLU activation. The branched layers, N_d and N_c, each consist of two 256-dimensional linear layers with ReLU activation. After the final linear layer, a tanh activation is applied to each branch. The weights of the final linear layer of each branch are initialized to zero so that the original content mesh is unaltered at initialization. We divide the output of N_c by 2 and add it to $[0.5, 0.5, 0.5]$. This enforces the final color prediction c_p to be in range $(0.0, 1.0)$. We find that initializing the mesh color to $[0.5, 0.5, 0.5]$ (grey) and adding the network output as a residual helps prevent undesirable solutions in the early iterations of training. For the branch N_d, we multiply the final tanh layer by 0.1 to get displacements in the range $(-0.1, 0.1)$.

4.2. Training

We use the Adam optimizer with an initial learning rate of $5e^{-4}$, and decay the learning rate by a factor of 0.9 ev-

*Authors contributed equally.

Figure 1. Our method generates structured textures which automatically align to sharp features and curves. Prompt: 'A wooden chair'

ery 100 iterations. We train for 1500 iterations on a single Nvidia GeForce RTX2080Ti GPU, which takes around 25 minutes to complete. For augmentations Ψ_{global}, we use

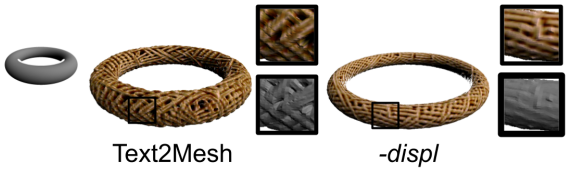


Figure 2. An example showing the effectiveness of the displacement loss

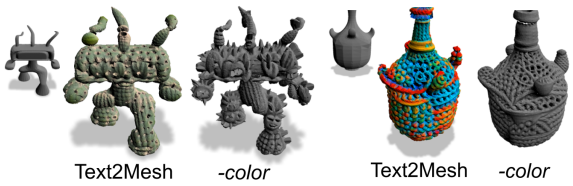


Figure 3. Training without predicting color.

a random perspective transformation. For Ψ_{local} we randomly crop the image to 10% of its original size and then apply a random perspective transformation. Before encoding images with CLIP, we normalize per-channel by mean $(0.48145466, 0.4578275, 0.40821073)$ and standard deviation $(0.26862954, 0.26130258, 0.27577711)$. We show an example of intermediate training stages in Fig. 11.

5. Baseline Comparison and User Study

Examples of results for our VQGAN baseline, as described in Section 4.3, are shown in Fig. 12 and Fig. 13, alongside our results.

(a) Our Method (b) VQGAN-CLIP

Figure 12. Prompt: 'A shoe made of cactus'

Figure 4. Rendering at 1670x2720 resolution. Prompt: ‘A colorful crochet vase’

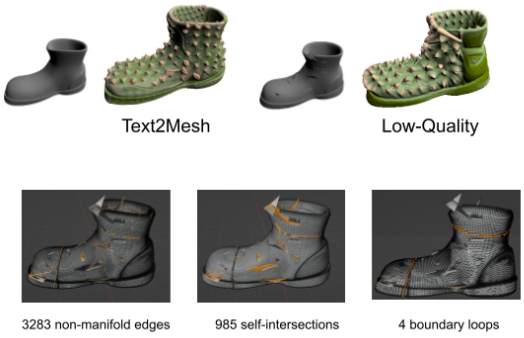


Figure 5. Training on a low quality mesh with non-manifold edges, boundary loops and self-intersections.

Figure 6. Style results over a coarse torus (left) and the same mesh with each triangle barycenter inserted as an additional vertex (right). Prompt: ‘a donut made of cactus’

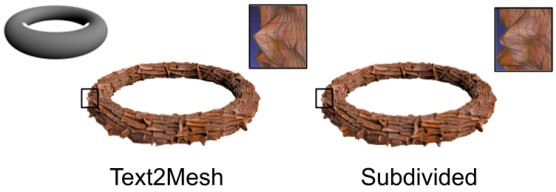
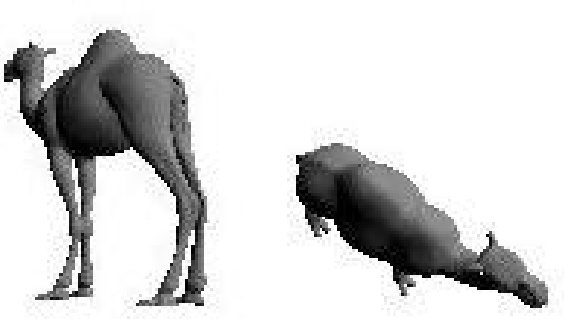


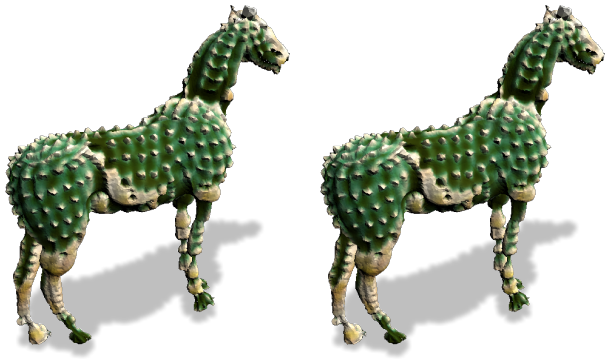
Figure 7. When trained on a coarse mesh, our method infers details on the object surface after it is subdivided.

Figure 8. CLIP scores for each vertex view.



(a) CLIP Score: 0.35 (b) CLIP Score: 0.20

Figure 9. Example views with CLIP similarities.



(a) # Views: 5 (Base) (b) # Views: 6

(c) # Views: 7

Figure 10. Style outputs sampling different # views. Prompt: 'A horse made of cactus'

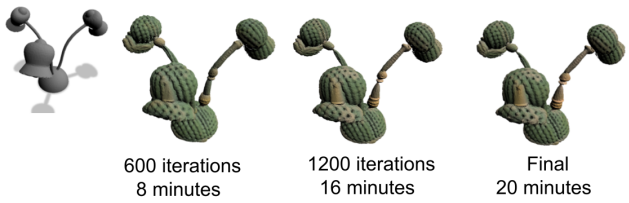


Figure 11. Intermediate training stages.

In addition, in Fig. 14, we provide screenshots of our user study, as shown for users.

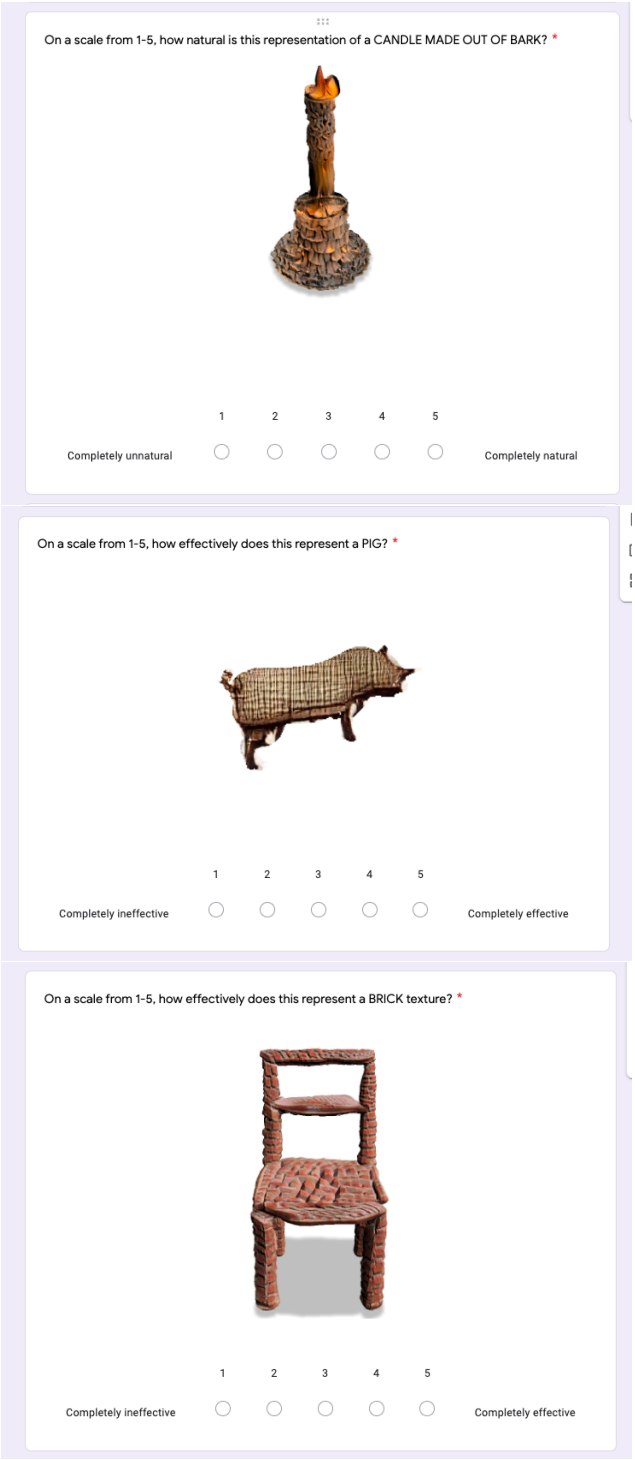


Figure 14. Example questions from user study

6. Societal Impact

Our framework utilizes a pre-trained CLIP embedding space, which has been shown to contain bias [1]. Since our system is capable of synthesizing a style driven by a target text prompt, it enables visualizing such biases in a direct and transparent way. We've observed evidence of societal bias in some of our stylizations. For example, the nurse style in Fig. 15 is biased towards adding female features to the input male shape. Our method offers one of the first opportunities to *directly* observe the biases present in joint image-text embeddings through our stylization framework. An important future work may leverage our proposed system in helping create a datasheet [2] for CLIP in addition to future image-text embedding models.

Figure 15. Our method enables visualizing the biases in the CLIP embedding space. Given a human male input (source in Figure 3), and target prompt: 'a nurse', we observe a gender bias in CLIP.

References

[1] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021.

[2] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.