Solid Texture Synthesis from 2D Exemplars

Team 04

VISHWESH VHAVLE and MRISHIKA NAIR

1 INTRODUCTION

Texture synthesis is an interesting and an important problem in the field of computer graphics and is one of the most essential techniques for realistic image synthesis. Texture synthesis techniques are thus of much interest. For our term project in CSE333: Computer Graphics we will implement a method for synthesizing a 3D solid texture from a 2D exemplar, based on the paper Solid Texture Synthesis from 2D Exemplars, SIGGRAPH 2007[2].

A 2D exemplar is an exemplar image of a target texture and our goal is to generate high-quality samples of the target texture in 3D. The texture in 3D is what we refer to as solid texture. Solid textures provide texture information not only on surfaces but also throughout the entire volume occupied by a solid object. Solid texture synthesis or STS is the process of mapping the 2D exemplar to 3D solid texture. 3D textures are difficult to obtain, this makes STS even more interesting while being an extremely challenging task.

2 LITERATURE REVIEW

Solid texture synthesis has attracted considerable research interest in the field of computer graphics and vision since it was proposed by Perlin [4] and Peachey [3]. We come a long way since then and most modern Solid texture synthesis solutions like [6] and [5] are now using deep learning techniques like GANs to generate seamless solid textures from 2D exemplars that look very natural and are computationally much better on the fly once the models are trained.

The scope of this project will remain limited to using a non-deep learning approach for solid texture synthesis. Keeping this in mind, a popular and effective approach for 2D texture synthesis is to simply copy patches of the given exemplar to generate a much larger 2D texture optimized to reduce seams between the patches. This approach does not extend well to 3D textures as we need to generate a value from the exemplar for each voxel such that the solid texture looks natural even when sliced through.

The paper Solid Texture Synthesis from 2D Exemplars [2] proposes using techniques from non-parametric texture synthesis together with a global histogram matching approach. The proposed solution has a global texture energy function which we try to optimally minimize in order for the solid texture to gradually deviate from the provided exemplar such that we get a solid texture that looks seamless and natural for most exemplars.

To ensure that the optimization process makes use of the full potential of a given 2D exemplar, the optimizer must not get stuck in a local minima i.e. starts using the same patch of exemplar again and again. To prevent this, authors use a popular digital image processing technique called Histogram Matching. Histogram matching ensures the use of the entire exemplar and the authors note that the final result is much better in quality and performance.

Authors' address: Vishwesh Vhavle, vishwesh20156@iiitd.ac.in; Mrishika Nair, mrishika20389@iiitd.ac.in.

2D exemplars contain the basic three R, G, and B color channels for each texel, however, the paper experiments with the use of more channels, which are displacement, shininess, and specularity. Which the authors note is very non-trivial to accomplish and sometimes not possible for certain textures. Another important provision that the paper provides is synthesis control which allows us to fine-tune the final output for a particular texture.

3 MILESTONES

S. No.	Milestone	Member	Status
	Mid evaluation		
1	Render complex 3D models like Stanford Bunny	Vishwesh Vhavle	Done
2	Collect 2D exemplars and perform basic 2D to 3D mapping	Mrishika Nair	Done
3	Implement stochastic texture synthesis algorithm with texture energy optimization. (Shared Milestone)	Vishwesh Vhavle Mrishika Nair	Done
	Final evaluation		
4	Adding Histogram Matching to improve Solid Optimization (Shared Milestone)	Vishwesh Vhavle Mrishika Nair	Done
5	Use the algorithm to generate texel values for the mesh from exemplar	Vishwesh Vhavle	Done
6	Add Shininess and Specularity	Mrishika Nair	Done

4 APPROACH

We initially rendered a 3D model of the Stanford Bunny from the PLY open-source Stanford PLY repository, and then we collected some 2D texture exemplars and performed 2D to 3D spherical mapping. Later, we tried to wrap the texture image around the object, but since the bunny was constructed by the interpolation of many triangles, we had to further divide our texture exemplars into a combination of pixels following the stochastic texture synthesis algorithm and create a much larger texture file. The final texture image was created using the non-parametric texture synthesis algorithm, by stochastically selecting neighbourhoods from the exemplar and sampling from the neighbourhoods such that the new patch is similar to the previous patch. We compare the patches by calculating the RMS distance between the images.

Next, we added Histogram Matching to improve texture synthesis. The Histogram Matching score of the result texture if a candidate patch is selected is determined using histogram matching of the result texture with candidate patch and exemplar. If the score is above a certain threshold, the patch is allowed in the result. We need noticed better qualitative results as the result texture is synthesised by sampling more diverse neighbourhoods so as to maximise the richness of the texture.

We extended this algorithm to simultaneously generate 3 output textures whose energies at each texel is optimized. For our solid texturing, we calculate the normal at each triangle, then align the normal to the closest axis, and finally based on this, we select a texture from one of three planes to shade the triangle with. This is implemented in the vertex shader itself

We further used the computed per-vertex normals for the mesh and included the shininess and specular lighting. To enhance the realism of the object, we tried including a parallax map, which refers to altering the texture coordinates so that there appears to be a height difference in the texture. To offset the texture coordinates at the fragment position, we scaled the fragment-to-view direction vector by the height at that particular fragment to get those coordinates at the surface. But this approximation showed a discrepancy in areas where heights changed frequently. To solve this, we tried transforming the view direction vector to tangent space by calculating a TBN matrix(combination of tangent, bitangent, normal). Then we took the inverse of this matrix to bring it to the same space, so that the tangent, bitangent vectors and the surface's texture coordinates point in the same direction regardless of the surface's orientation.

5 RESULTS

We observed that indeed the non-parametric texture synthesis algorithm [1] is prone to getting stuck in local minima. In order to deal with this without histogram matching, we simply changed the neighbourhoods after 10 failed attempts with the RMS error higher than the threshold initially. Later, we implemented Histogram Matching which handled the issue. The Histogram Matching also improved the final texture's richness. The final results of the solid texturing are visualized below.

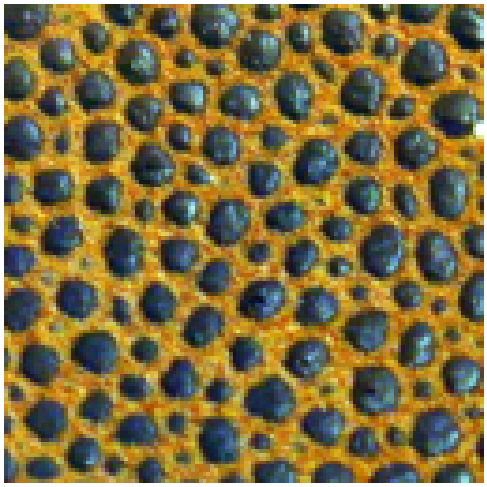

Fig. 1. Exemplar Texture

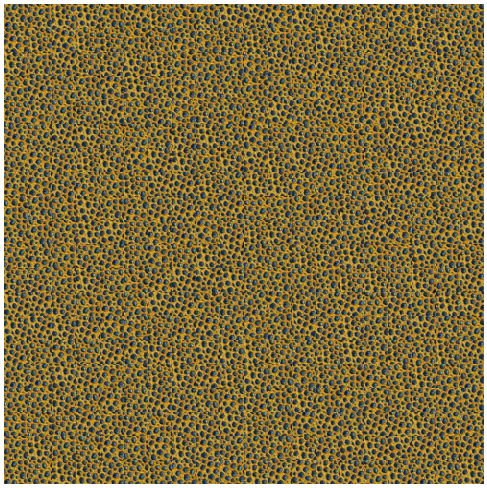

Fig. 2. Final Texture

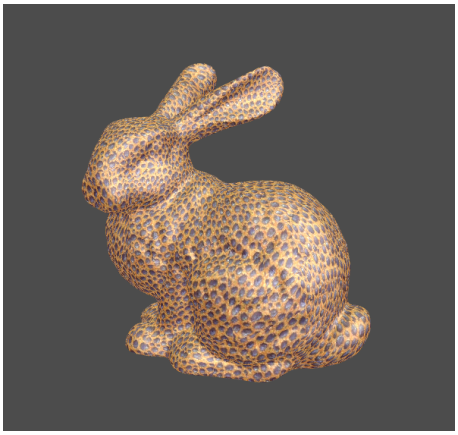

Fig. 3. Textured Stanford Bunny

6 CONCLUSION

The implementation allows any OBJ file to be loaded and by simply selecting the texture exemplar, we can solid texture the model. Thus we have so far implemented all the milestones we set out to accomplish. We would like to appreciate and acknowledge the course instructor Dr. Ojaswa Sharma and the teaching assistants, Shounak Chatterjee and Gaurav Rai, for providing us with the boilerplate code and offering guidance whenever we asked for it.

REFERENCES

[1] A.A. Efros and T.K. Leung. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2. 1033–1038 vol.2. https://doi.org/10.1109/ICCV.1999.790383

[2] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. 2007. Solid texture synthesis from 2D exemplars. In *ACM SIGGRAPH 2007 papers (SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 2–es. https://doi.org/10.1145/1275808.1276380

[3] Darwyn R. Peachey. 1985. Solid texturing of complex surfaces. *ACM SIGGRAPH Computer Graphics* 19, 3 (July 1985), 279–286. https://doi.org/10.1145/325165.325246

[4] K. Perlin. 1995. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics* 1, 1 (March 1995), 5–15. https://doi.org/10.1109/2945.468392

[5] Tiziano Portenier, Siavash Arjomand Bigdeli, and Orcun Goksel. 2020. GramGAN: Deep 3D Texture Synthesis From 2D Exemplars. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 6994–7004. https://proceedings.neurips.cc/paper/2020/hash/4df5bde009073d3ef60da64d736724d6-Abstract.html

[6] Xin Zhao, Jifeng Guo, Lin Wang, Fanqi Li, Junteng Zheng, and Bo Yang. 2022. STS-GAN: Can We Synthesize Solid Texture with High Fidelity from Arbitrary Exemplars? https://doi.org/10.48550/arXiv.2102.03973 arXiv:2102.03973 [cs, eess].