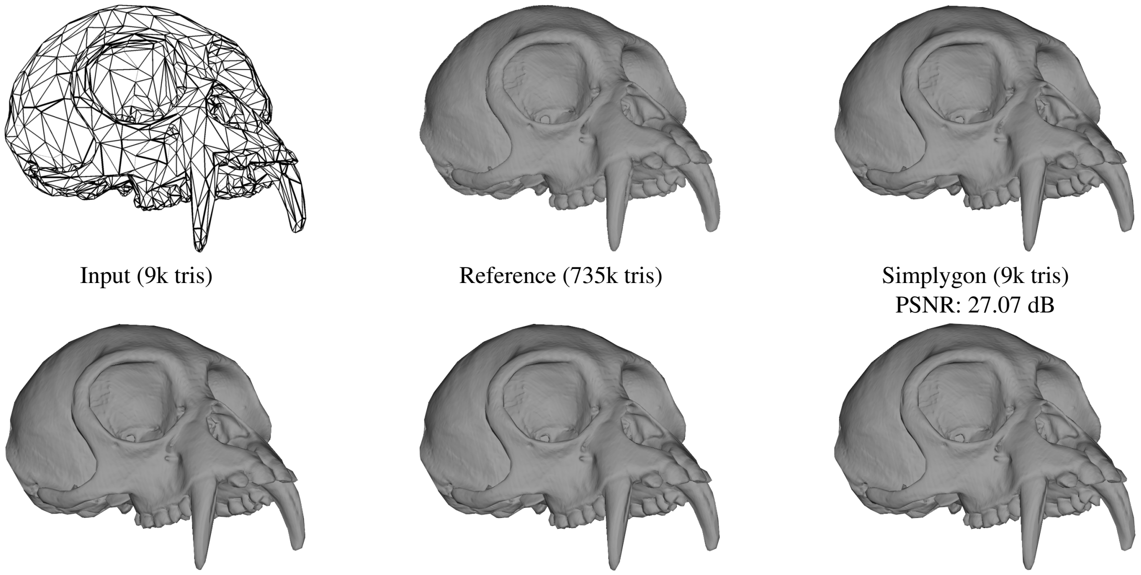
Eurographics Symposium on Rendering (2021)
A. Bousseau and M. McGuire (Guest Editors)

Appearance-Driven Automatic 3D Model Simplification

J. Hasselgren[1], J. Munkberg[1], J. Lehtinen[1,2], M. Aittala[1] and S. Laine[1]

[1]NVIDIA Research
[2]Aalto university



Input (9k tris)

Reference (735k tris)

Simplygon (9k tris)
PSNR: 27.07 dB

Substance (9k tris)
PSNR: 27.11 dB

Our, normals only (9k tris)
PSNR: 26.22 dB

Our, normals & positions (9k tris)
PSNR: 28.30 dB

Figure 1: *A comparison with normal map baking. The starting point is a reduced base mesh with 9k triangles (reduced from the 735k triangle reference in Simplygon 8.3) and we bake a normal map from a 735k triangle reference using the normal bakers of Simplygon 8.3 and Substance Painter v2020.2.2. To our knowledge, these bakers only optimize the normal map, and leave the base mesh geometry unmodified. In our version, we jointly optimize the base geometry and normals based on rendered image observations.*

1. Comparison with Normal Map Bakers

In Figure 1 we compare our approach with two production quality normal map bakers (Simplygon 8.3 and Substance Painter v2020.2.2) on a scanned skull, Hylobates sp.: Cranium[†], courtesy of the Smithsonian [Smi18]. The normal map resolution is 2048 × 2048 texels for all versions and all versions use the same reduced input mesh with 9k tris (generated in Simplygon). To our knowledge, both Simplygon and Substance generate the normal map by ray tracing from the reduced mesh to the high resolution reference. In contrast, our version is optimized from image observations using a resolution of 2048 × 2048 pixels for 10k steps (randomized viewing conditions and lighting) in our differentiable rasterizer. As can be seen, given that we optimize both vertex positions of the base mesh and normal map texels from image observations, we obtain a slightly higher-quality result on the same triangle bud-

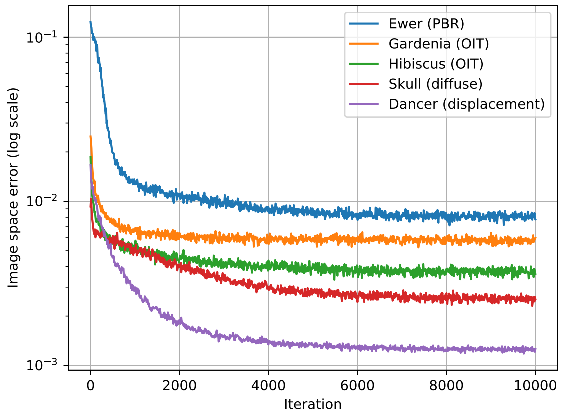
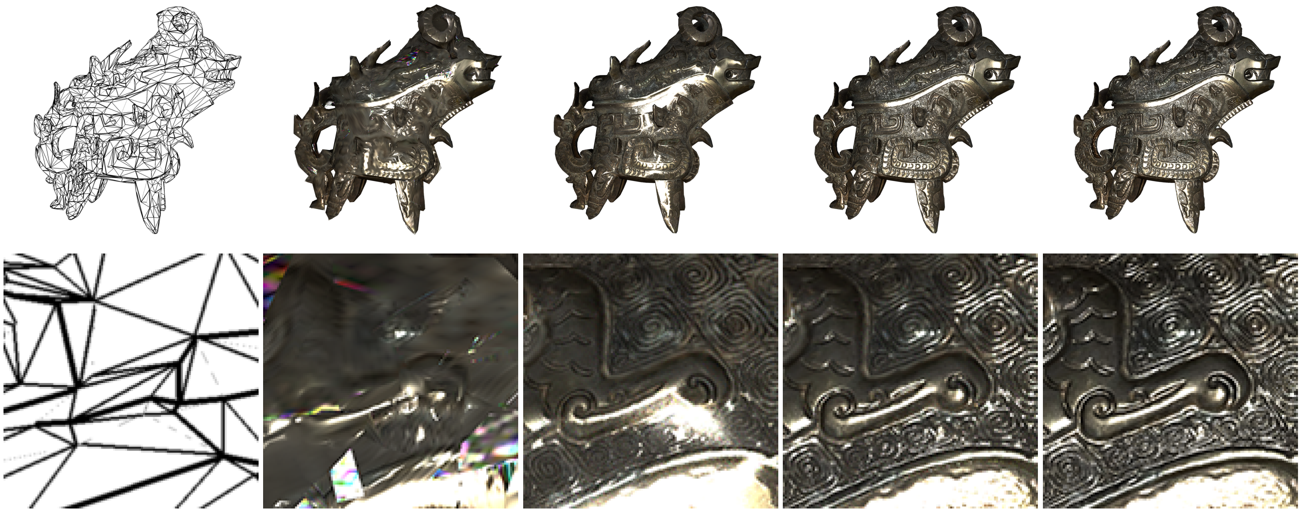


Figure 2: *Training convergence plots for five optimization examples from the paper.*

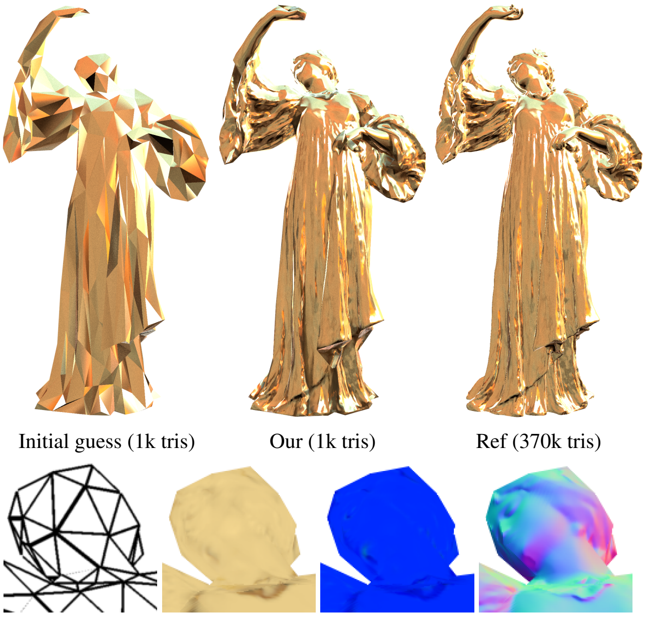
[†] https://3d.si.edu/object/3d/hylobates-klossii-cranium:4b963018-7642-431e-99ba-603b51bd158f

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.



Initial guess (7k tris)	w/o Laplacian	w/o normal map	Our	Reference (300k tris)
PSNR (dB)	18.13	18.15	24.49	
FLIP	0.124	0.127	0.0588	

Figure 3: *The Ewer bronze sculpture, courtesy of the Smithsonian 3D Digitization project [Smi18]. The reference mesh consists of 300k triangles, normal maps, textured base color and a bronze metal material. We start from a reduced version of the reference mesh with 7k triangles, with randomized material parameters. In this case, we obtain a high quality result, closely approximating the reference. The insets highlight the results obtained when we disable either the normal map or the Laplacian regularizer during optimization. We note that both components are critical to obtain high quality results. Without the normal map, we lose high-frequency detail. Without the Laplacian regularizer, the mesh is malformed, which subsequently also blurs the material parameters.*



Initial guess (1k tris) Our (1k tris) Ref (370k tris)

Mesh Diffuse map Specular map Normals

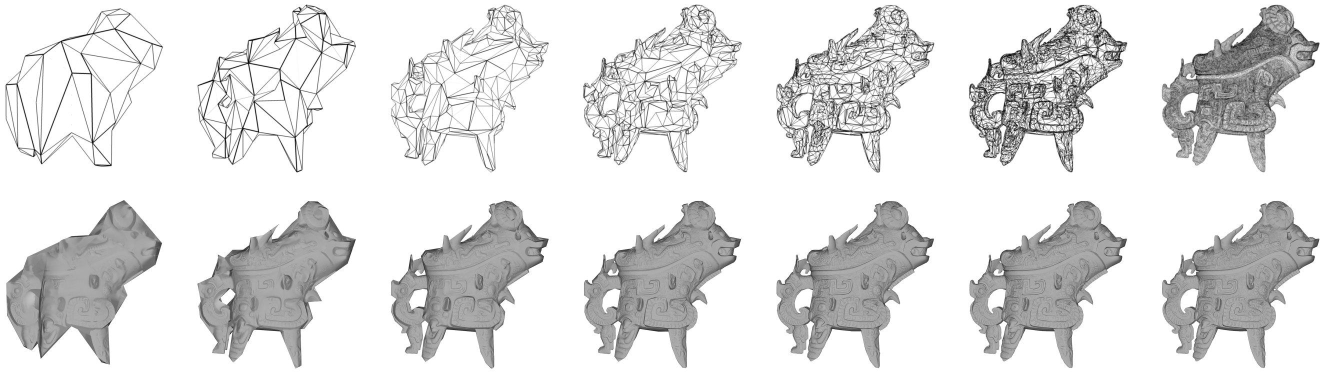
Figure 4: *We optimize shape and appearance for a decimated version (0.3% tris remaining) of the dancer. The insets show the geometry and materials terms of our latent representation. Here, we place the model in a new lighting configuration (an HDR environment probe), to show that it generalizes well.*

get. If we restrict the optimization to only normal map texels, the quality is slightly lower (cf. “Our, normals only” versus “Ours, normals & positions”).

2. Convergence

In Figure 2 we show training convergence plots for five examples from the paper. The normal map baking example (Skull) is the easiest task, with diffuse shading, low dynamic range and joint optimization of the vertex positions and the tangent space normal map texels. The displacement map example (Dancer) performs joint optimization of shape, normal- and displacement maps and uses a lower initial learning rate. The aggregate geometry examples, Gardenia and Hibiscus, are harder, as they include order-independent transparency and PBR shading. Additionally, in those examples, we used an MSE loss function to minimize the differences against the reference when transferring the optimized assets to a path tracer, and the MSE loss is more sensitive to outliers than the tone mapped L_1 variant (see main paper) used in the other examples. Finally, we show the Ewer sculpture, which includes high-frequency specular materials and high dynamic range lighting. Note that the convergence plot show the image space *training* error, and as we randomize the light and camera position each training iteration, some remaining noise is expected throughout the training. We note that all examples converge nicely.

Learning rate is scheduled as $\mathrm{lr}_i = \mathrm{lr}_0 \cdot 10^{-kt}$, where t is the iteration, lr_0 is the initial learning rate, and $k = 0.0002$. When starting from a coarse initial guess, e.g., a sphere or billboard cloud, we typically use a high initial learning rate, e.g., $\mathrm{lr}_0 = 0.01$, to allow for



132 tris	500 tris	1.2k tris	2.9k tris	7k tris	18k tris	Reference
18.42	20.78	23.77	26.36	29.44	30.14	PSNR
0.3825	0.1086	0.0404	0.0209	0.0149	0.0137	Chamfer-L^1

Figure 5: *Influence of initial guess. We show six different input meshes, ranging from 130 to 18k triangles, all trying to approximate a reference mesh with 300k triangles (Ewer model courtesy of the Smithsonian 3D Digitization project [Smi18]). The Chamfer-L^1 scores are multiplied with a factor 10^3.*

large mesh deformations. When starting from an auto-decimated mesh, we want to fine-tune the result without jumping out of the already good local minima, which leads us to using lower values for $\mathrm{lr}_0 \in [0.001, 0.003]$. We note that mini-batching is highly beneficial to reduce gradient noise, particularly during the early phases of optimization. Gradient noise is problematic for vertex positions as it can cause the mesh to fold or self-intersect, especially in highly tessellated regions. In practice, we use batch sizes between one and eight.

3. Influence of Normal Map and Laplacian

In Figure 3 we show how the use of normal map and Laplacian regularizer during the optimization influences the quality of our results. We report PSNR and FLIP [ANA*20] scores. We run 10k steps of optimization at a resolution of 2048×2048 pixels. All material textures are initialized to random values, except for the normal map, which is initialized to $(0,0,1)$. Normal maps help capture micro-detail, which is clearly visible in the insets. The Laplacian regularizer helps stabilize optimization and improves mesh quality. Without it, large initial optimization steps may cause mesh corruption or self-intersections which are hard to recover from.

4. Material breakdown

Figure 4 shows a breakdown of the geometry and material parameters, diffuse texture, specular (orm) texture and normal map for the dancer statue. We additionally place the model in a new lighting configuration (an HDR environment probe) to show that it generalizes well.

5. Mesh Decimation: Varying Triangle Count in the Reduced Mesh

In Figure 5 we study quality as function of the triangle count in our initial guess for the Ewer model. All reduced versions are generated in Simplygon 8.3. As can be seen, small details and silhouettes benefit greatly from increased triangle count. This result also shows the importance of optimizing the normal map. Details that are not part of the silhouette are captured well even at low triangle counts. Even the model with 500 triangles reasonably estimates the overall appearance of the reference mesh and could be used as a distant level of detail.

6. Tessellation as Level of Detail

Tessellation is often used as a level of detail scheme. Here, we optimize for a single *common* base mesh, displacement map and normal map, with the objective that their renderings reproduce the reference at all levels of tessellation. Figure 6 shows increasing levels of tessellation on the dancer model from a base mesh with 1k triangles, optimized with 2048×2048 pixels resolution and 5k iterations. We use edge-midpoint-tessellation, wherein each subdivision step quadruples the triangle count. From the insets we note that, as expected, silhouette edges and details are improved as tessellation is increased.

Note that the fingers on the right hand of the figure are never accurately captured even at the highest level of tessellation. Here, the limitation is in the base mesh. Displacement mapping cannot introduce concavities, and since the hand is not sufficiently modeled in the base mesh it cannot be recreated through displacement mapping. This would require increasing the polygon count of the base mesh, or possibly through an artist improving the initial guess to have higher tessellation in this region.

7. Approximating Aggregate Detail: Comparison with Stochastic Simplification

In Figure 7, we compare against stochastic simplification of aggregate detail. Following Cook et al. [CHPR07], we stochastically

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

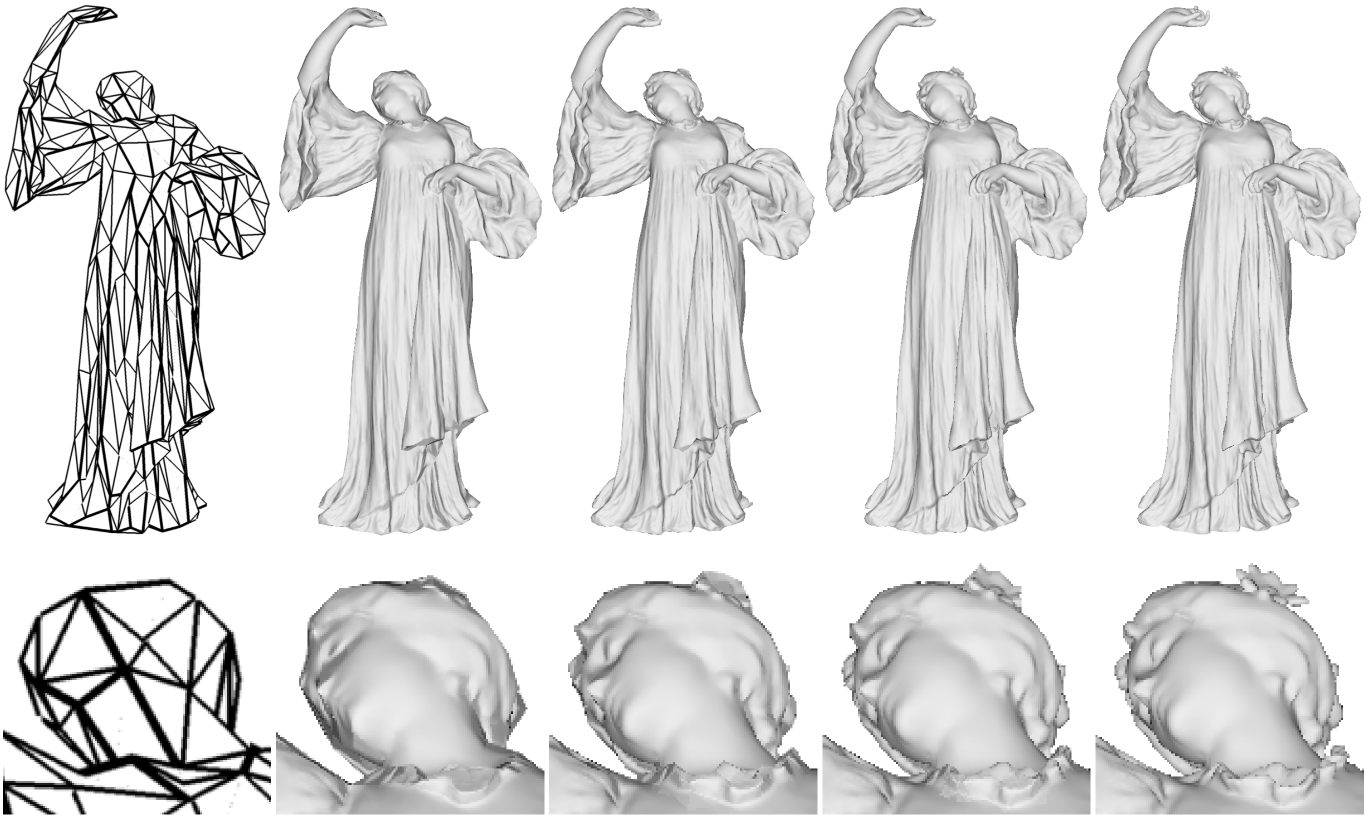


Figure 6: *A level of detail example with different levels of tessellation rendered from a single base mesh. All levels of tessellation use the same base mesh, displacement map and normal map. As can be seen in the insets, silhouette and details are improved with increased tessellation.*

remove 90% ($\lambda = 0.1$) of the leaves from one instance of the Disney Moana Island Gardenia asset [Wal18], and adjust the element area of the reduced model by scaling each leaf uniformly with a factor $\sqrt{1/\lambda}$. Finally, the contrast of each leaf texture is adjusted by modifying the color of its texels as $c_i' = \bar{c} + \sqrt{\lambda}(c_i - \bar{c})$, where $\bar{c}$ is the average color of the texture.

Our version is optimized from image observations using a resolution of 2048 × 2048 pixels for 10k steps (randomized viewing conditions and lighting) in our differentiable rasterizer. We start from an initial guess with 6.5k triangles where each leaf geometry is replaced with a quad and material parameters are randomized. Please refer to the paper for a visual example of the input mesh. We visualize both a single model, and a larger scene with 3000 instances, rendered in a path tracer. Our version, with more aggressive reduction (99.6% of the triangles removed), still produces a high quality approximation by automatically moving geometry details into transparency- and normal maps. The shape and material parameters are optimized from image observations, so no heuristic for scaling or contrast adjustments is needed.

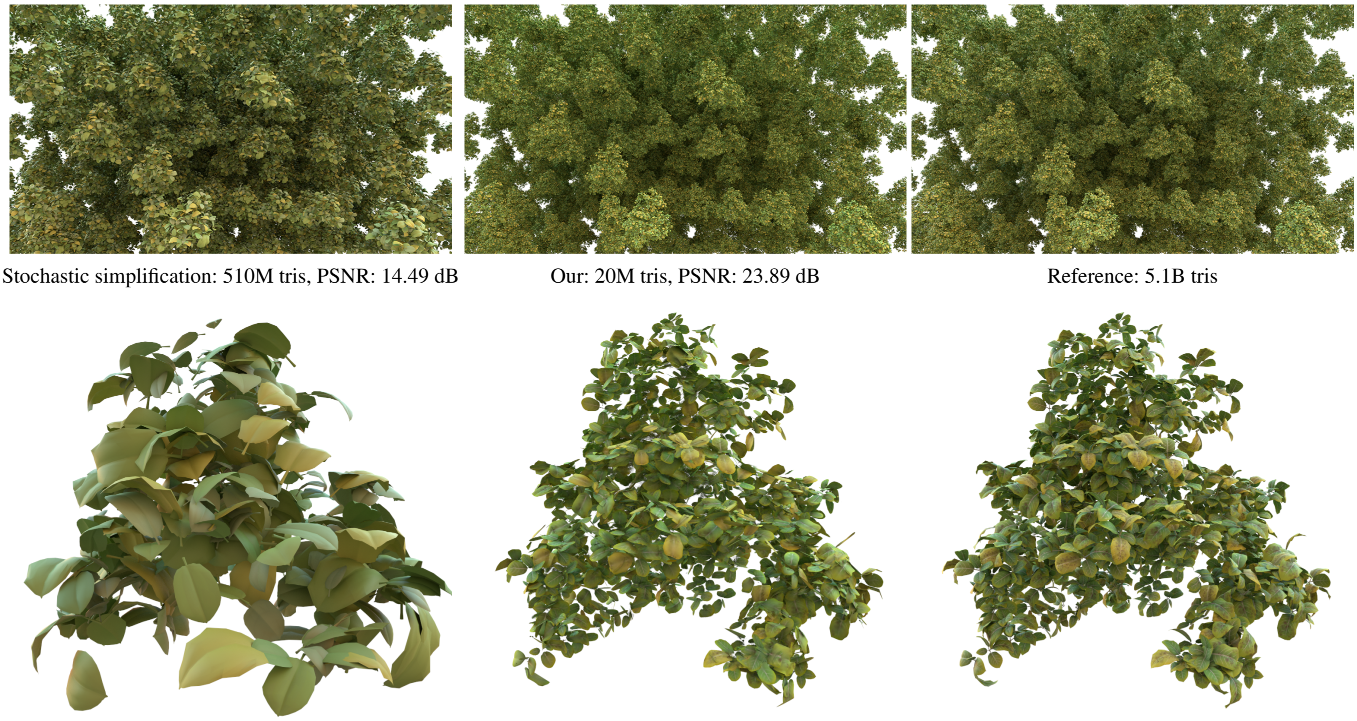
8. Learning Mesh and Materials from Implicit Surfaces

In Figure 9 we show an example of learning shape and materials to approximate a signed distance field rendered using ray marching. We adapt a version of the ShaderToy “Elephant“ from Inigo Quilez[‡], modified to isolate the main object. Figure 10 shows a harder example with subsurface scattering, based on the ShaderToy “Snail“ from Inigo Quilez[§], modified to isolate the main object.

We use a sphere with 12k triangles as an initial guess for the rasterizer, and optimize at a resolution of 2048 × 2048 pixels for 10k steps. The appearance of the shaded result matches the reference well, and the sphere deforms to a reasonable mesh (please refer to the wireframe inset). However, we note that this example is limited by the quality of the initial guess, and further efforts would be required to generalize to more complex assets.

[‡] https://www.shadertoy.com/view/MsXGWr

[§] https://www.shadertoy.com/view/ld3Gz2



Stochastic simplification: 510M tris, PSNR: 14.49 dB

Our: 20M tris, PSNR: 23.89 dB

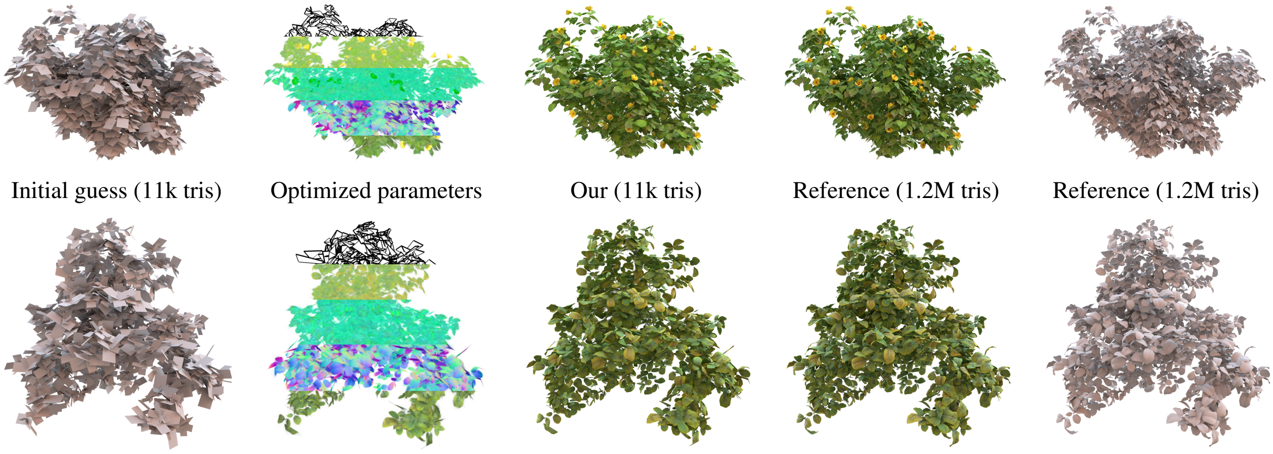
Reference: 5.1B tris

170k tris
Stochastic simplification

6.5k tris
Our model

1.7M tris
Reference

Figure 7: *Our approach compared to stochastic simplification of aggregate detail [CHPR07].*



Initial guess (11k tris)

Optimized parameters

Our (11k tris)

Reference (1.2M tris)

Reference (1.2M tris)

Initial guess (6.5k tris)

Optimized parameters

Our (6.5k tris)

Reference (1.7M tris)

Reference (1.7M tris)

Figure 8: *Approximating aggregate geometry. We start from a low-polygon mesh and jointly optimize shape, material parameters, and transparency. The shaded results are rendered in a path tracer to illustrate that our results generalize across renderers.* ***Top row:*** *The leaves and flowers of the "isHibiscus" asset (1.2M triangles), approximated by 11k tris.* ***Bottom row:*** *The leaves from the "isGardenia" asset (1.7M triangles), approximated by 6.5k triangles. The models are taken from the Moana Island Scene [Wal18], a publicly available data set courtesy of Walt Disney Animation Studios.*

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

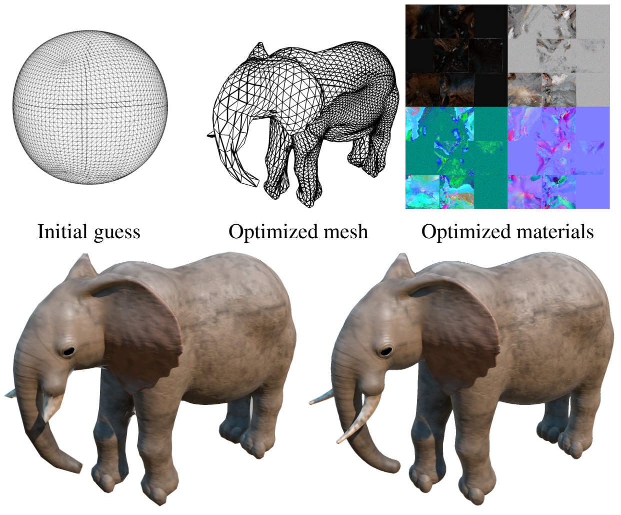


Figure 9: *We extract a mesh and materials from a ray marched distance field: the ShaderToy "Elephant," from Inigo Quilez. We initialize the optimization process by a sphere with random material parameters and learn shape and material parameters such that our rasterized model resembles the ray marched reference.*

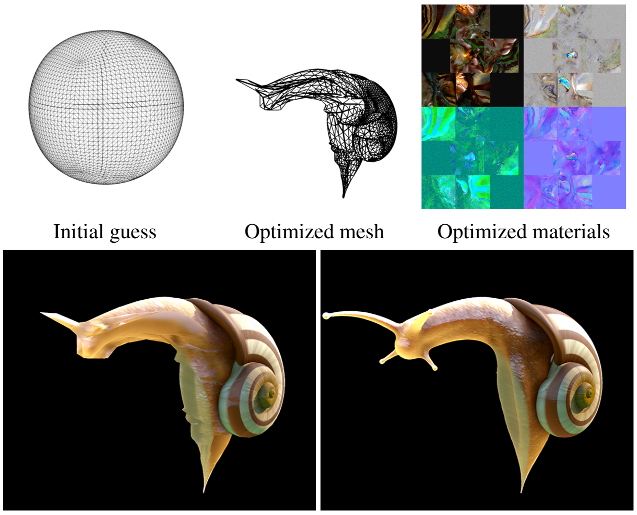


Figure 10: *Similar to Figure 9, we extract a mesh and materials from a ray marched distance field: the ShaderToy "Snail," from Inigo Quilez.*

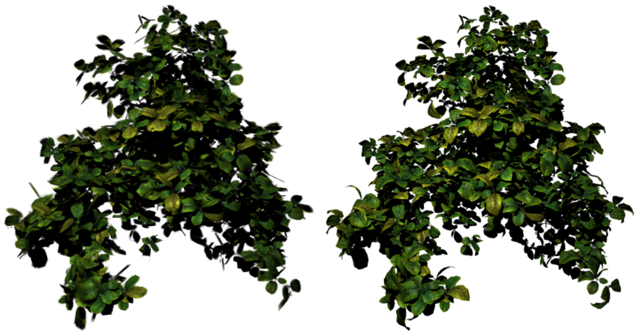

Our rasterized model (6.5k tris)
Path traced reference (1.7M tris)

Figure 11: *We optimize a mesh and materials to have the appearance of a path traced reference, when rendered in a rasterizer. This can be used to convert between different material models, or as a simple way of baking shading into material terms. Our optimized model is rasterized with 1 spp + post-processing antialiasing. The path traced reference is rendered with 256 spp.*

9. Baking path traced lighting

In Figure 11, we use the ViSII path tracer [MTBW20] to generate reference images for the aggregate geometry decimation example. We use a static light position during optimization: materials are effectively converted to our material model and shadows are baked into the textures, creating a plausible rasterized approximation of the path traced reference. Note that we control the viewing conditions in the reference images, and use matching configurations in the optimization.

10. Shape and Appearance Prefiltering

Figure 12 is an extension of Figure 2 in the main paper, and shows the dancer model specifically optimized for four different rendering resolutions. Our results match the appearance of the antialiased reference well, considering the difference in sample count. As expected, geometric detail and shading are gradually smoothed as the rendering resolutions decrease, even though all results are generated from the same initial guess.

References

[ANA*20] Andersson P., Nilsson J., Akenine-Möller T., Oskarsson M., Åström K., Fairchild M. D.: FLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 3*, 2 (2020), 15:1–15:23. 3

[CHPR07] Cook R. L., Halstead J., Planck M., Ryu D.: Stochastic Simplification of Aggregate Detail. *ACM Trans. Graph. 26*, 3 (2007). 3, 5

[MTBW20] Morrical N., Tremblay J., Birchfield S., Wald I.: ViSII: Virtual scene imaging interface, 2020. https://github.com/owl-project/ViSII/. 6

[Smi18] Smithsonian: Smithsonian 3D Digitization, 2018. https://3d.si.edu/. 1, 2, 3

[Wal18] Walt Disney Animation Studios: Moana island scene (v1.1), 2018. http://technology.disneyanimation.com/islandscene/. 4, 5

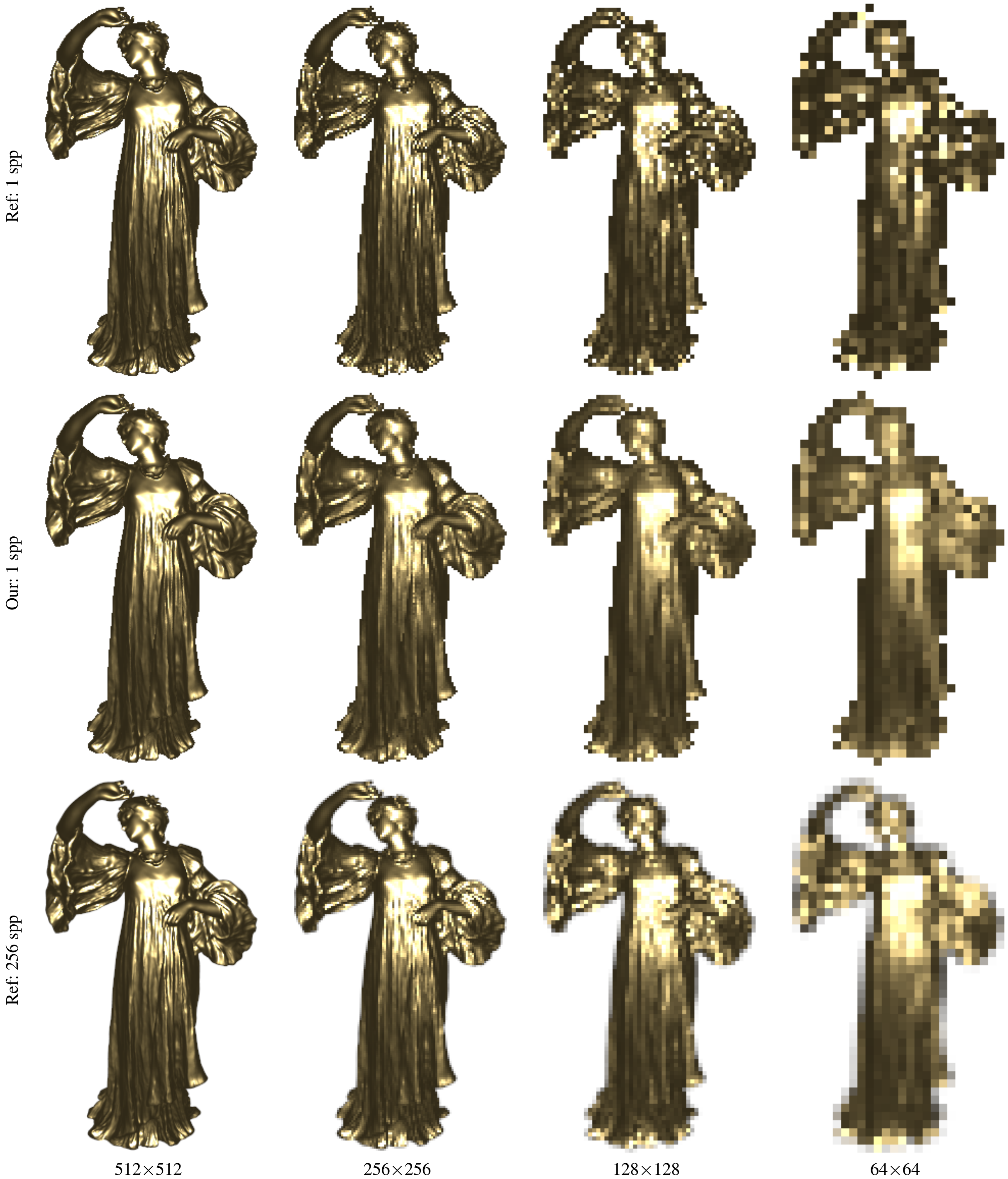


Figure 12: *We perform shape and appearance prefiltering by optimizing for a particular rendering resolution. Here, we show our results for four different resolutions. As expected, geometric details and shading are smoothed as rendering resolution decrease.* ***Top:*** *The reference dancer model rendered at 1 spp, note the aliasing.* ***Middle:*** *Our optimized model with prefiltered shape and appearance rendered at 1 spp.* ***Bottom:*** *The reference dancer model rendered, with antialiasing, at 256 spp.*