3D Neural Sculpting (3DNS): Editing Neural Signed Distance Functions (Supplementary Material)

Petros Tzathas[1] **Petros Maragos**[1] **Anastasios Roussos**[2,3]

[1]School of Electrical & Computer Engineering, National Technical University of Athens, Greece
[2]Institute of Computer Science (ICS), Foundation for Research & Technology - Hellas (FORTH), Greece
[3]College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK

1. Other Brush Types

In the main paper (Section 4.2) we used a single brush template, which was based on a quintic polynomial. Here, we describe a more general approach.

1.1. Smoothstep Functions

In computer graphics, the need to transition smoothly from one real number to another arises very frequently. For this purpose, various functions are used, which take the value 0 for $x < 0$, the value 1 for $x > 1$, and go from 0 to 1 in the interval $[0, 1]$ in an continuously differentiable increasing manner, with vanishing derivatives at 0 and 1. In graphics, such a function is usually called *smoothstep*. Refer to Inigo Quilez's site [3] for a presentation of some smoothstep functions.

1.2. Radially Symmetric Brushes

If f is a smoothstep function, then we can define a brush template that is radially symmetric, as follows:

$$b_T(\boldsymbol{x}) = f\left(1 - \|\boldsymbol{x}\|\right) \tag{1}$$

Any function from [3] can be used. The brush we use for our experiments is of this type. Besides smoothsteps, any function f defined over $[0, 1]$ that has value 0 at $x = 0$ and a maximum value of 1 can be used in the equation above.

1.3. Vector Brushes

A possible extension we can make to our brush formulation is to allow the brush template function to take vector values instead of just scalars. Such an extension would allow to create brushes that twists the surface locally around the interaction point.

2. Pre-training

During the initial training of the networks so that they represent the unedited shapes we encountered a problem.

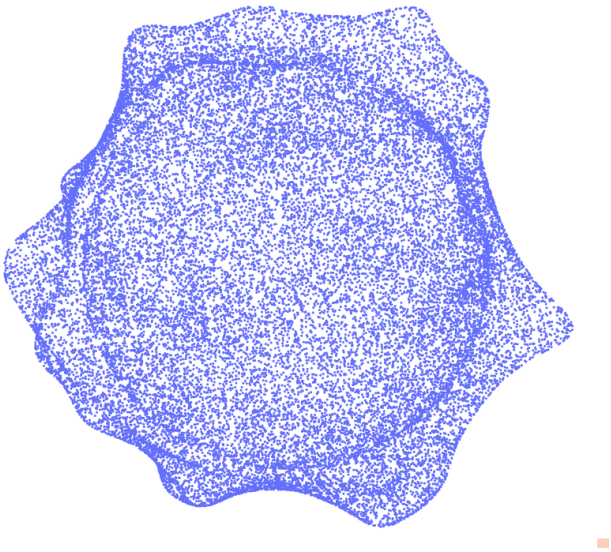

Figure 1: Sampled zero-level set of a neural network that was unsuccessfully trained to represent a sphere.

Sometimes the training fails. Figure 1 shows the sampled zero-level set of a network that was (unsuccessfully) trained to represent a sphere. A sphere can actually be discerned. Nevertheless, there also exists an outer shell. This situation manifested for different shapes as well. We hypothesize that this is due to the nature of the loss function, the network's size, and poor initialization of the network weights.

The loss function (equations 3-6) does not specify the value the network should take anywhere other the zero-level set. Depending on initialization, the network can take negative values outside the surface. The fact that we want the network function to be positive outside the surface and negative inside is expressed in the loss function only through the term for the normals and, hence, in a local fashion. Early

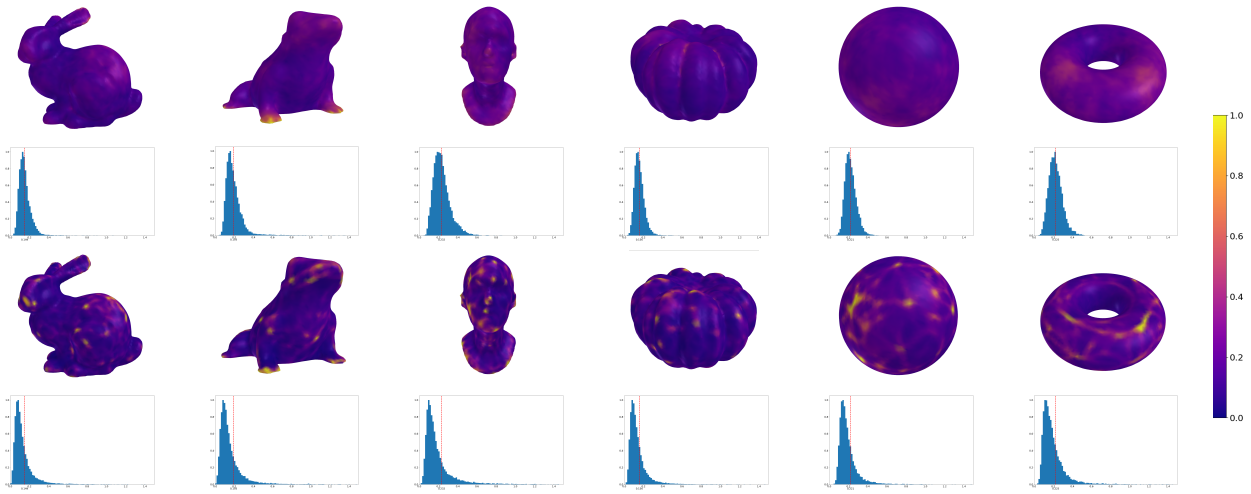

Figure 2: The results of PDF estimation for all the shapes in the dataset. For the top two rows our proposed algorithm was used, while for the bottom two the naive approach. For an explanation of the figures refer to the main paper (Section 5.5). Please zoom in for details.

(a) Dining chair (b) Chair (c) Armchair (d) Sofa (e) Vase

4dd46b9657c0e998b4d5420f7c27d2df *02e76cb4f1039c482eb499cc8fbcd* *c5d880efc887f6f4f9111ef49c078dbe* *bcff6c5cb4127aa15e0ae65e074d3ee1* *13375f8fce3142e6597d391ab6fcc1*

Figure 3: ShapeNet shapes used for surface editing comparisons. Beneath each caption the ShapeNet model ID of the corresponding shape is written in italics.

in training, this normal term will lead to the network taking positive values outside the surface but only close to the boundary. Far from the surface the network function continue to be negative. This creates the outer shell we see in the figure. Once the shell is created, the eikonal term L_{eik} (equation 5) and the term that penalizes small values of the network function L_{es} (equation 6) lead to a local minimum.

The training that proceeds can be regarded through a variational point of view. L_{eik} acts as a constraint on the network function forcing it to be an SDF, which means that the outer shell created will change as a surface in a continuous manner during gradient descent. The minimization of L_{es} requires that the area of the outer shell decreases and so the shell shrinks. When the outer shell is close enough to the surface of the shape which the network is trained to represent the shell cannot shrink further because of the network's limited capacity. Between the shell's surface and the shape's surface there exists a discontinuity in the SDF's gradient. This discontinuity, of course, exists even when the shell is far from the shape, however when it gets close the network's gradient is required to change too abruptly which is not possible for its size. Thus, the training is stuck at a local minimum.

We notice that larger networks sizes and/or more complex shapes help to avoid the issue. The solution we opted for is pre-training the network. That is, before starting the actual training we execute 100 iterations using the following loss function:

$$L_{pre} = \mathbb{E}_q \{ \left| f_\theta(\boldsymbol{x}) - \|\boldsymbol{x}\| \right| \} \tag{2}$$

where θ is the parameter vector, f_θ is the network function, and q is the uniform distribution in a bounding box. This lead to the network function being positive and so avoids the creation of the shell.

3. Additional PDF Estimation Results

In the main paper, we provided PDF Estimation results only for the bunny and the sphere in the main paper (Section 5.5). In Figure 2, we provide them for all the shapes in

Shape	Mean Chamfer Distance $\times 10^3$ (↓)					
	Over whole surface			Inside interaction area		
	Ours	Naive	Simple Mesh	Ours	Naive	Simple Mesh
Dining chair	**7.256**	20.545	8.260	**8.843**	11.288	10.703
Chair	**8.498**	30.620	8.650	**8.741**	19.072	15.344
Armchair	**14.152**	19.433	14.311	**5.085**	10.942	23.654
Sofa	**11.160**	18.268	11.899	**4.477**	8.623	23.940
Vase	**9.063**	15.565	10.345	**10.477**	15.713	16.725
Average	**10.026**	20.8862	10.693	**7.525**	13.128	18.073

Table 1: Comparison of our editing method with and without model samples (Ours and Naive, respectively) and direct mesh editing on a mesh with equivalent size (Simple Mesh). Chamfer distances are computed with 100000 points. The mean for each shape is taken over 10 independent edits.

the dataset. Here, as well, it is clear the our proposed algorithm produces a more uniform distribution than the naive approach.

4. Additional Surface Editing Comparisons

We also run the surface editing comparison experiment for five shapes from the ShapeNet dataset [1] shown in Figure 3. Since the ShapeNet meshes do not have adequately high resolution to capture the desired edits accurately and in order to have a fair comparison between the approaches, we consider as ground truth a high resolution mesh constructed by Marching Cubes [2]. This is similar to the process that we adopted for the Sphere and Torus shapes in the experimental comparisons of the main paper. We follow the same protocol of experimental comparisons as in Sec. 5.6 of the main paper and the results are reported in Table 1. We observe that, once again, our method outperforms the compared approaches consistently for all shapes and with respect to both metrics used (i.e. Mean Chamfer Distance over the whole surface and inside the interaction area).

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. `https://shapenet.org`.

[2] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *COMPUTER GRAPHICS*, 21(4):163–169, 1987.

[3] Inigo Quilez. Smoothstep functions, 2022. `https://iquilezles.org/articles/smoothsteps/`.