DIST: Rendering Deep Implicit Signed Distance Function with Differentiable Sphere Tracing

Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, Zhaopeng Cui
Deep Implicit Signed Distance Functions (DeepSDF)

- Infinite-Resolution
- Lightweight

Voxel-based Representation  
[Tatarchenko et al.]

DeepSDF Representation  
[Park et al.]

No differentiable renderer for DeepSDF!

Feedforward Rendering Results

LR texture | 32x HR texture | HR Relighting | HR 2\textsuperscript{nd} View
Optimization over Shape Code
DIST - Differentiable Sphere Tracing

How to make feedforward efficient?

What ray convergence criteria is the best setup?

Efficient feedforward of sphere tracing algorithm

Gradient computation on the rendered silhouette

What can we do to resolve the GPU memory burden?

How to deal with non-differentiable rendered silhouette?
DIST Feedforward – Naive Sphere Tracing

For each camera ray, march along the ray direction at each step with the queried SDF value until convergence.
We start the sphere tracing over an image with \( \frac{1}{4} \) resolution, and split each ray twice during the marching process, which saves computation at the early stage.
DIST Feedforward – Aggressive Marching

Setting step size $\alpha > 1$ incurs bouncing between both sides.

Setting step size $\alpha > 1$ speeds up convergence.

$$|d(1 - \alpha \sin \theta)^k| < \epsilon$$

$$k > k_{min} = \frac{\log \epsilon - \log d}{\log |1 - \alpha \sin \theta|}$$
DIST Feedforward – Convergence Criteria

A large threshold $\epsilon$ causes dilation, while a small threshold leads to erosion.

We stop the marching once the SDF value is smaller than $1/2 \epsilon$. 

A large threshold $\epsilon = 5 \times 10^{-2}$ causes dilation, while a small threshold $\epsilon = 5 \times 10^{-8}$ leads to erosion.
The computation becomes affordable while the results remain almost unchanged.
DIST Backward – Recursive Gradients

Each query location depends on the previous ones, incurring recursive gradients.

\[ d = \alpha \sum_{n=0}^{N-1} f(\mathbf{p}^{(n)}) + (1 - \alpha) f(\mathbf{p}^{(N-1)}) = d' + e \]

\[ \frac{\partial d'}{\partial \mathbf{z}} \bigg|_{z_0} = \alpha \sum_{i=0}^{N-1} \left( \frac{\partial f_{\theta}(\mathbf{p}^{(i)}(\mathbf{z}), \mathbf{z})}{\partial \mathbf{z}} \right) \bigg|_{z_0} \]

This term is omitted as it empirically has less impact on the optimization process.
We make use of the nice property of signed distance function to optimize the nearest surface geometry.
Optimization over Camera Parameters

Given a fixed shape, our differentiable renderer can successfully backpropagate gradients to the camera parameters with respect to 2D observations.
Results - Reconstruction from Sparse Depth Images

<table>
<thead>
<tr>
<th></th>
<th>dense</th>
<th>50%</th>
<th>10%</th>
<th>100pts</th>
<th>50pts</th>
<th>20pts</th>
</tr>
</thead>
<tbody>
<tr>
<td>sofa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepSDF</td>
<td>5.37</td>
<td>5.56</td>
<td>5.50</td>
<td>5.93</td>
<td>6.03</td>
<td>7.63</td>
</tr>
<tr>
<td>Ours</td>
<td>4.12</td>
<td>5.75</td>
<td>5.49</td>
<td>5.72</td>
<td>5.57</td>
<td>6.95</td>
</tr>
<tr>
<td>Ours (mask)</td>
<td>4.12</td>
<td>3.98</td>
<td>4.31</td>
<td>3.98</td>
<td>4.30</td>
<td>4.94</td>
</tr>
<tr>
<td>plane</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepSDF</td>
<td>3.71</td>
<td>3.73</td>
<td>4.29</td>
<td>4.44</td>
<td>4.40</td>
<td>5.39</td>
</tr>
<tr>
<td>Ours</td>
<td>2.18</td>
<td>4.08</td>
<td>4.81</td>
<td>4.44</td>
<td>4.51</td>
<td>5.30</td>
</tr>
<tr>
<td>Ours (mask)</td>
<td>2.18</td>
<td>2.08</td>
<td>2.62</td>
<td>2.26</td>
<td>2.55</td>
<td>3.60</td>
</tr>
<tr>
<td>table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepSDF</td>
<td>12.93</td>
<td>12.78</td>
<td>11.67</td>
<td>12.87</td>
<td>13.76</td>
<td>15.77</td>
</tr>
<tr>
<td>Ours</td>
<td>5.37</td>
<td>12.05</td>
<td>11.42</td>
<td>11.70</td>
<td>13.76</td>
<td>15.83</td>
</tr>
<tr>
<td>Ours (mask)</td>
<td>5.37</td>
<td>5.15</td>
<td>5.16</td>
<td>5.26</td>
<td>6.33</td>
<td>7.62</td>
</tr>
</tbody>
</table>
Results - Reconstruction from Video Sequences

Synthetic #1

Lin et al.

Ours

Real-world #1

Lin et al.

Ours

Synthetic #2

Lin et al.

Ours

Real-world #2

Lin et al.

Ours

Code and Demo are available here

http://b1ueber2y.me/projects/DIST-Renderer/
DIST: Rendering Deep Implicit Signed Distance Function with Differentiable Sphere Tracing

Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, Zhaopeng Cui

Motivation & Pipeline
The recently proposed deep implicit signed distance function [1] is effective on representing 3D shapes. Advantages: infinite resolution, lightweight, etc.

No differentiable renderer exists for this representation, making it infeasible to be optimized over 2D observations.

DIST – Feedforward

Naive Sphere Tracing
For each camera ray, march at each step with the queried SDF value until convergence.

Coarse-to-fine Strategy
We start the sphere tracing over an image with ¼ resolution, and split each ray twice during the marching process, which saves computation at the early stage.

Aggressive Marching

Convergence Criteria
Take fixed length $l = 100m$, sensor size $S = 500m$, resolution $R = 32^2$, maximum depth $l_{max} = 100m$. We use $\epsilon = 10^{-4}$. A large threshold causes dilation, while a small threshold leads to erosion.

DIST - Backward

Memory issue caused by Recursive Gradients
Each query location depends on the previous one, incurring recursive gradients. We make approximations over sphere tracing by omitting high-order gradients.

Differentiable Silhouette
We make use of the nice property of signed distance function to optimize the nearest surface geometry.

Feedforward Rendering

Optimization over Camera Parameters

Reconstruction from Video Sequences

Results on synthetic data

Results on real data

Reconstruction from Sparse Depths

Full-Resolution: http://b1ueber2y.me/projects/DIST-Renderer/pdf/4986-poster.pdf