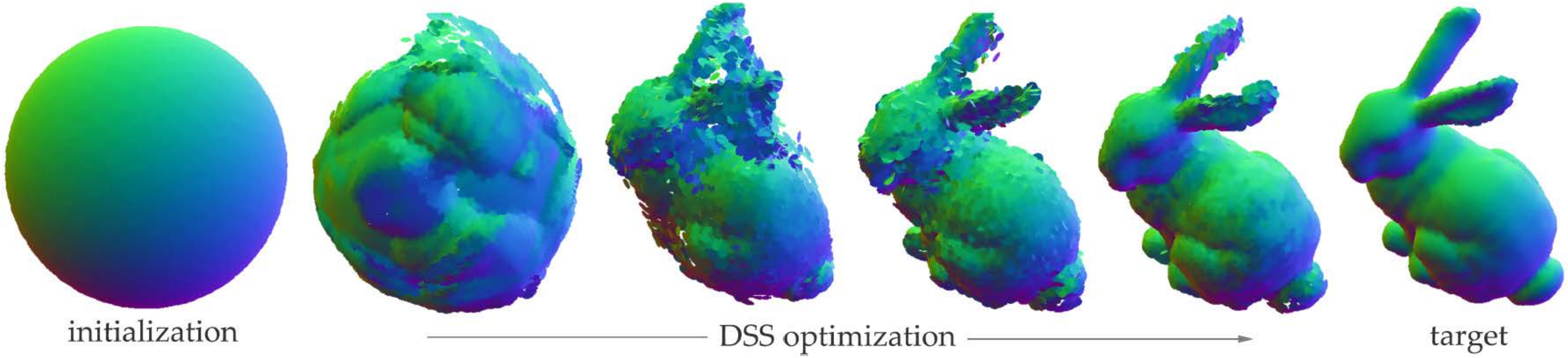
Differentiable Surface Splatting for Point-based Geometry Processing

WANG YIFAN, ETH Zurich, Switzerland
FELICE SERENA, ETH Zurich, Switzerland
SHIHAO WU, ETH Zurich, Switzerland
CENGIZ ÖZTIRELI, Disney Research Zurich, Switzerland
OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland



Presenter: Yuan Yao

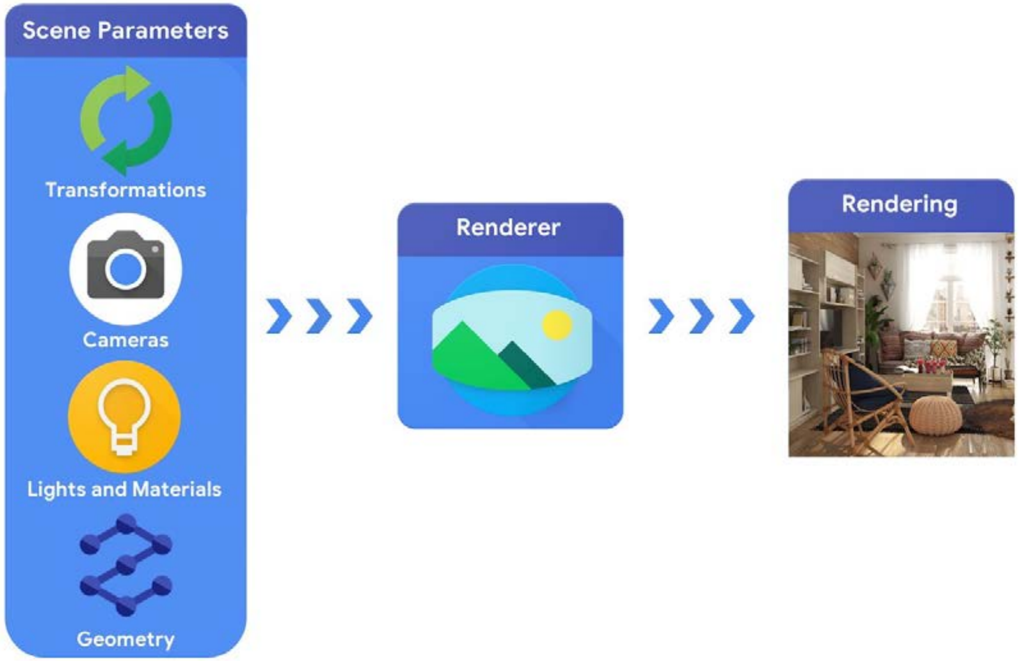
Sept. 17th 2019

Differentiable Surface Splatting(DSS)

- A high-fidelity differentiable renderer for point cloud.
  - Gradients for point locations and normals.
  - Regularization terms.
  - Application to inverse rendering for geometry synthesis and denoising.
  - Outperforming state-of-the-art.

What is DR(Differentiable Renderer) ?

- Takes scene-level information θ such as geometry, lighting, material and camera position as input, and outputs a synthesized image I.
- Any changes in the image I can be propagated to the parameter θ, allowing for image-based manipulation of the scene.



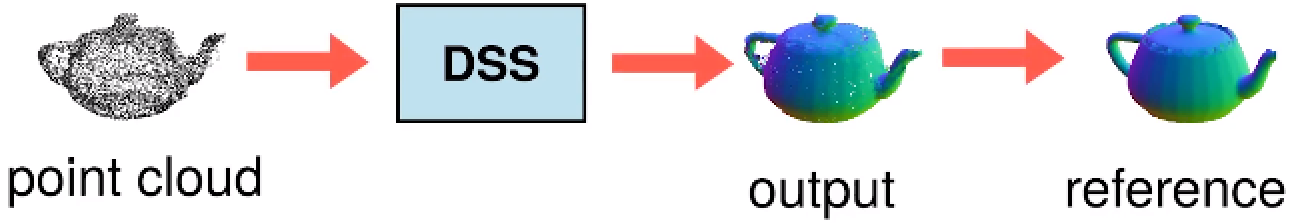
Existing DRs

method	objective	position update	depth update	normal update	occlusion	silhouette change	topology change
OpenDR	mesh	✓	✗	via position change	✗	✓	✗
NMR	mesh	✓	✗	via position change	✗	✓	✗
Paparazzi	mesh	limited	limited	via position change	✗	✗	✗
Soft Rasterizer	mesh	✓	✓	via position change	✓	✓	✗
Pix2Vex	mesh	✓	✓	via position change	✓	✓	✗
Ours	points	✓	✓	✓	✓	✓	✓

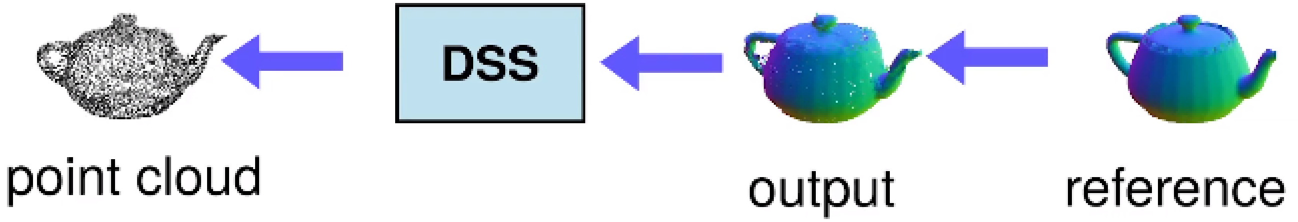
Table 1. Comparison of generic differential renderers. By design, OpenDR [Loper and Black 2014] and NMR [Kato et al. 2018] do not propagate gradients to depth; Paparazzi [Liu et al. 2018] has limitation in updating the vertex positions in directions orthogonal their face normals, thus can not alter the silhouette of shapes; Soft Rasterizer [Liu et al. 2019] and Pix2Vex [Petersen et al. 2019] can pass gradient to occluded vertices, through blurred edges and transparent faces. All mesh renderers do not consider the normal field directly and cannot modify mesh topology. Our method uses a point cloud representation, updates point position and normals jointly, considers the occluded points and visibility changes and enables large deformation including topology changes.

Method Overview

- **Forward Pass** : Generate 2D image from 3D scene-level information



- **Backward Pass**: The information flow from rendered image $\mathrm{I} = R(\theta)$ to the scene parameters θ based on approximating the gradient $\frac{dI}{d\theta}$.



- **Surface Regularization**: Avoid local minima. Repulsion term and projection term.

Forward Pass - Surface Splatting

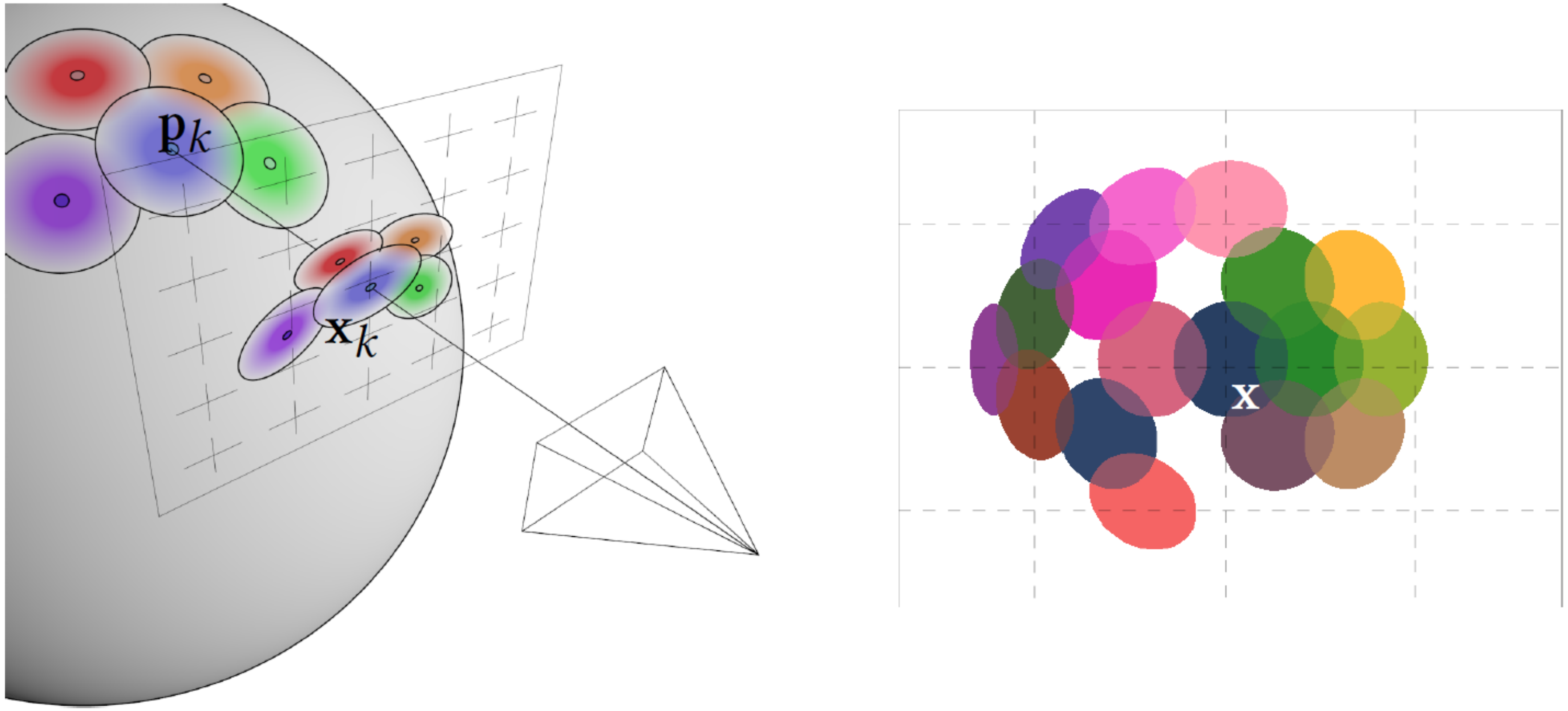


Fig. 2. Illustration of forward splatting using EWA [Zwicker et al. 2001]. A point in space $\mathbf{p}_k$ is rendered as an anisotropic ellipse centered at the projection point $\mathbf{x}_k$. The final pixel value $\mathbb{I}_\mathbf{x}$ at a pixel $\mathbf{x}$ in the image (shown on the right) is the normalized sum of all such ellipses overlapping at $\mathbf{x}$.

Forward Pass - Surface Splatting

- Isotropic Gaussian at position p for a point p_k:

$$\mathcal{G}_{\mathbf{p}_k,\mathbf{V}_k}(\mathbf{p}) = \frac{1}{2\pi|\mathbf{V}_k|^{\frac{1}{2}}} e^{(\mathbf{p}-\mathbf{p}_k)^\top \mathbf{V}_k^{-1}(\mathbf{p}-\mathbf{p}_k)}, \quad \mathbf{V}_k = \sigma_k^2 \mathbf{I}, \tag{2}$$

- Screen space elliptical Gaussian weight at x for x_k:

$$\begin{aligned} r_k(\mathbf{x}) &= \mathcal{G}_{\mathbf{V}_k}\left(\mathbf{J}_k^{-1}(\mathbf{x}-\mathbf{x}_k)\right) \\ &= \frac{1}{\left|\mathbf{J}_k^{-1}\right|} \mathcal{G}_{\mathbf{J}_k \mathbf{V}_k \mathbf{J}_k^\top}(\mathbf{x}-\mathbf{x}_k). \end{aligned} \tag{3}$$

- Final elliptical Gaussian weight:

$$\bar{\rho}_k(\mathbf{x}) = \frac{1}{\left|\mathbf{J}_k^{-1}\right|} \mathcal{G}_{\mathbf{J}_k \mathbf{V}_k \mathbf{J}_k^\top + \mathbf{I}}(\mathbf{x}-\mathbf{x}_k). \tag{4}$$

Forward Pass

- Resulting truncated Gaussian weight:

$$\rho_k(\mathbf{x}) = \begin{cases} 0, & \text{if } \frac{1}{2}\mathbf{x}^\top \left(\mathbf{J}\mathbf{V}_k\mathbf{J}^\top + \mathbf{I}\right)\mathbf{x} > C, \\ 0, & \text{if } \mathbf{p}_k \text{ is occluded,} \\ \bar{\rho}_k, & \text{otherwise.} \end{cases} \tag{5}$$

- Final pixel value:

$$\mathbb{I}_{\mathbf{x}} = \frac{\sum_{k=0}^{N-1} \rho_k(\mathbf{x})\, \mathbf{w}_k}{\sum_{k=0}^{N-1} \rho_k(\mathbf{x})}. \tag{6}$$

Forward Pass

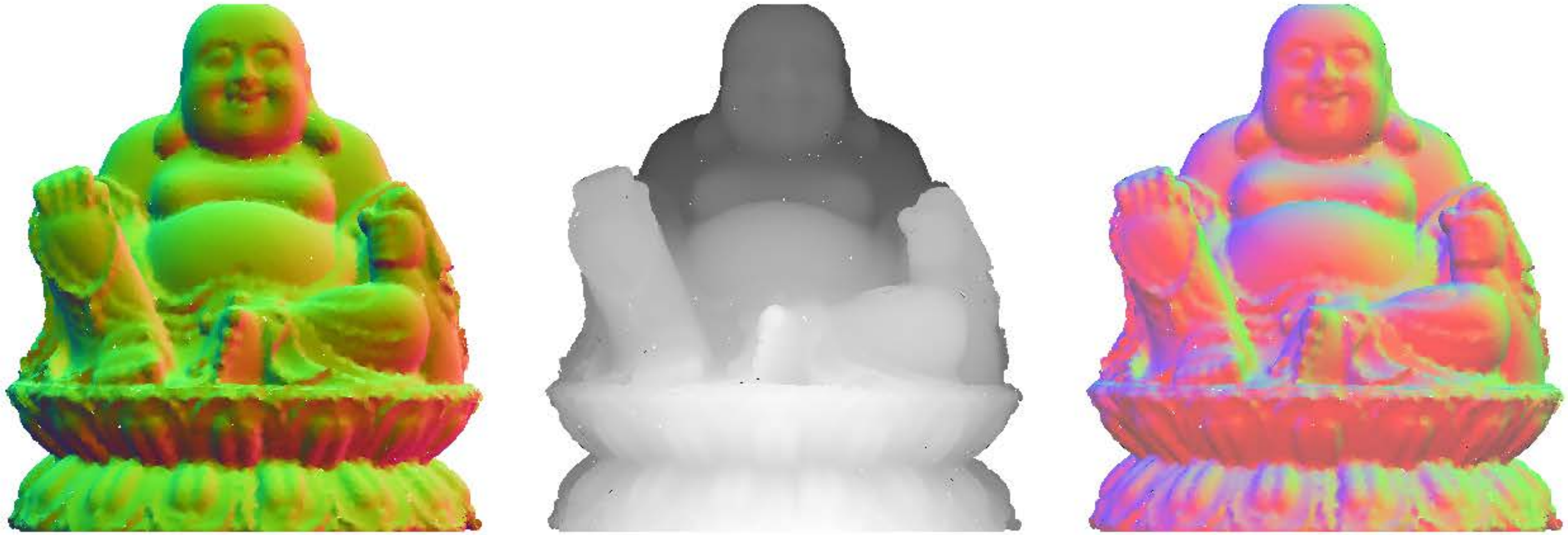


Fig. 3. Examples of images rendered using DSS. From left to right, we render the normals, inverse depth values and diffuse shading with three RGB-colored sun light sources.

Backward Pass

- Factorize the discontinuous ρ_k into fully differentiable term and discontinuous visibility term:

$$h_{\mathbf{x}}(\mathbf{p}_k) = \begin{cases} 0, & \text{if} \frac{1}{2}\mathbf{x}^\top \left(\mathbf{J}\mathbf{V}_k\mathbf{J}^\top + \mathbf{I}\right)\mathbf{x} > C, \\ 0, & \text{if } \mathbf{p}_k \text{ is occluded}, \\ 1, & \text{otherwise.} \end{cases} \quad (7)$$

- By omitting the impacts from normal to visibility(very small set of pixels), the chain rule is :

$$\frac{d\mathbb{I}_{\mathbf{x}}(\mathbf{w}_k, \bar{\rho}_k, h_{\mathbf{x}})}{d\mathbf{n}_k} = \frac{\partial \mathbb{I}_{\mathbf{x}}}{\partial \mathbf{w}_k}\frac{\partial \mathbf{w}_k}{\partial \mathbf{n}_k} + \frac{\partial \mathbb{I}_{\mathbf{x}}}{\partial \bar{\rho}_k}\frac{\partial \bar{\rho}_k}{\partial \mathbf{n}_k}, \quad (8)$$

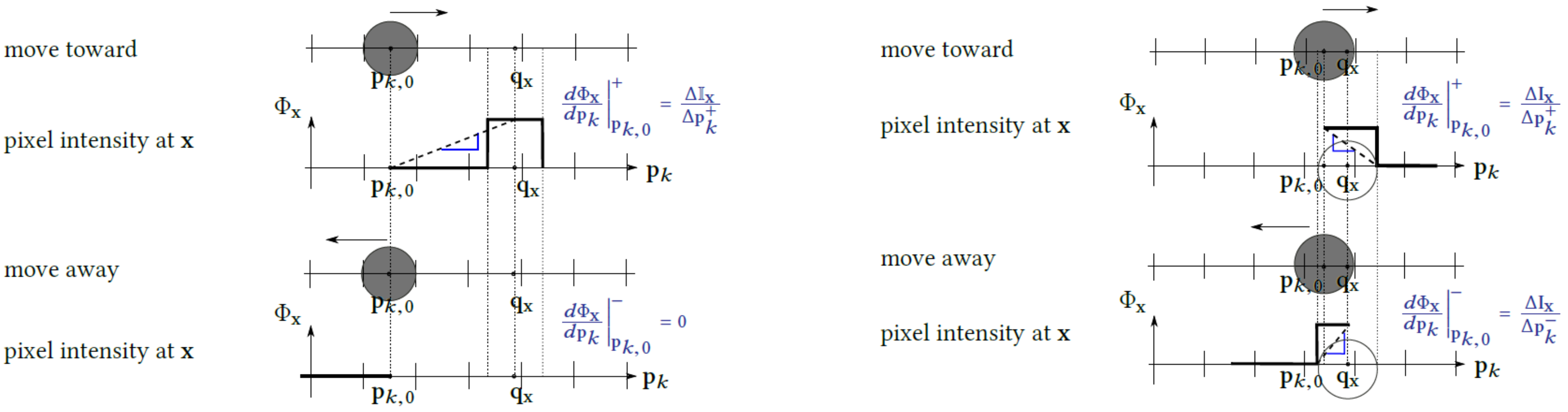
$$\frac{d\mathbb{I}_{\mathbf{x}}(\mathbf{w}_k, \bar{\rho}_k, h_{\mathbf{x}})}{d\mathbf{p}_k} = \frac{\partial \mathbb{I}_{\mathbf{x}}}{\partial \mathbf{w}_k}\frac{\partial \mathbf{w}_k}{\partial \mathbf{p}_k} + \frac{\partial \mathbb{I}_{\mathbf{x}}}{\partial \bar{\rho}_k}\frac{\partial \bar{\rho}_k}{\partial \mathbf{p}_k} + \frac{\partial \mathbb{I}_{\mathbf{x}}}{\partial h_{\mathbf{x}}}\frac{\partial h_{\mathbf{x}}}{\partial \mathbf{p}_k}, \quad (9)$$

Backward Pass

- Only need to focus on $\frac{\partial \mathbb{I}_{\mathrm{x}}}{\partial h_{\mathrm{x}}} \frac{\partial h_{\mathrm{x}}}{\partial \mathrm{p}_k}$, describes the change of a pixel intensity due to the visibility change of a point caused by its varying position.

Backward Pass

- 1D scenario



(a) The ellipse centered at $\mathbf{p}_{k,0}$ is not visible at $\mathbf{x}$.

(b) The ellipse centered at $\mathbf{p}_{k,0}$ is visible at $\mathbf{x}$.

Fig. 4. An illustration of the artificial gradient in two 1D scenarios: the ellipse centered at $\mathbf{p}_{k,0}$ is invisible (Fig. 4a) and visible (Fig. 4b) at pixel $\mathbf{x}$. $\Phi_{\mathbf{x},k}$ is the pixel intensity $\mathbb{I}_{\mathbf{x}}$ as a function of point position $\mathbf{p}_k$, $\mathbf{q}_{\mathbf{x}}$ is the coordinates of the pixel $\mathbf{x}$ back-projected to world coordinates. Notice the ellipse has constant pixel intensity after normalization (Eq. (6)). We approximate the discontinuous $\Phi_{\mathbf{x},k}$ as a linear function defined by the change of pixel intensity $\Delta\mathbb{I}_{\mathbf{x}}$ and the movement of the $\Delta\mathbf{p}_k$ during a visibility switch. As $\mathbf{p}_k$ moves toward $(\Delta\mathbf{p}_k^+)$ or away $(\Delta\mathbf{p}_k^-)$ from the pixel, we obtain two different gradient values. We define the final gradient as their sum.

Backward Pass

- The final gradient is defined as the sum of both

$$\left.\frac{d\Phi_x}{d\mathbf{p}_k}\right|_{\mathbf{p}_{k,0}} = \begin{cases} \frac{\Delta \mathbb{I}_\mathbf{x}}{\|\Delta \mathbf{p}_k^+\|^2+\epsilon} \Delta \mathbf{p}_k^+, & \mathbf{p}_k \text{ invisible at } \mathbf{x} \\ \frac{\Delta \mathbb{I}_\mathbf{x}}{\|\Delta \mathbf{p}_k^-\|^2+\epsilon} \Delta \mathbf{p}_k^- + \frac{\Delta \mathbb{I}_\mathbf{x}}{\|\Delta \mathbf{p}_k^+\|+\epsilon} \Delta \mathbf{p}_k^+, & \text{otherwise,} \end{cases} \tag{10}$$

- 3D scenario: similarly.

Surface Regularization

- Repulsion term: maximize distance between neighbors on a local projection plane.

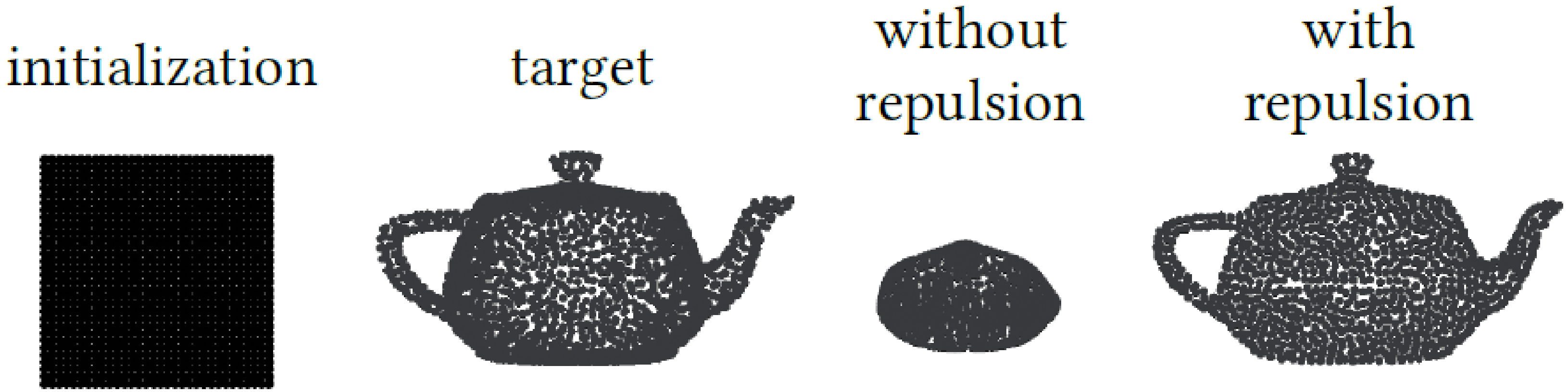


Fig. 8. The effect of repulsion regularization. We deform a 2D grid to the teapot. Without the repulsion term, points cluster in the center of the target shape. The repulsion term penalizes this type of local minima and encourages a uniform point distribution.

Surface Regularization

- Projection term: preserves clean surfaces by minimizing the distance from the point to the surface tangent plane.

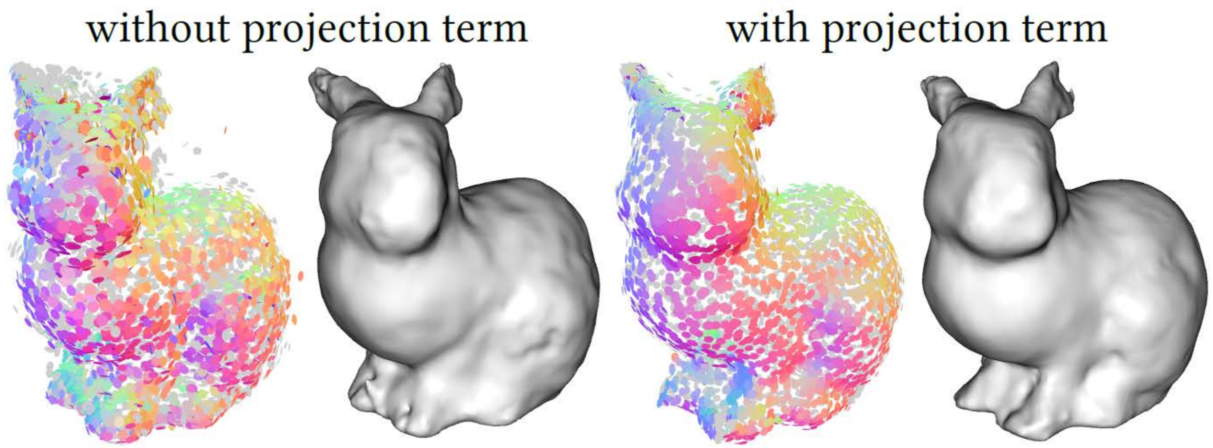


Fig. 9. The effect of projection regularization. The projection term effectively enforces points to form a local manifold. For a better visualization of outliers inside and outside of the object, we use a small disk radius and render the backside of the disks using light gray color.

Results

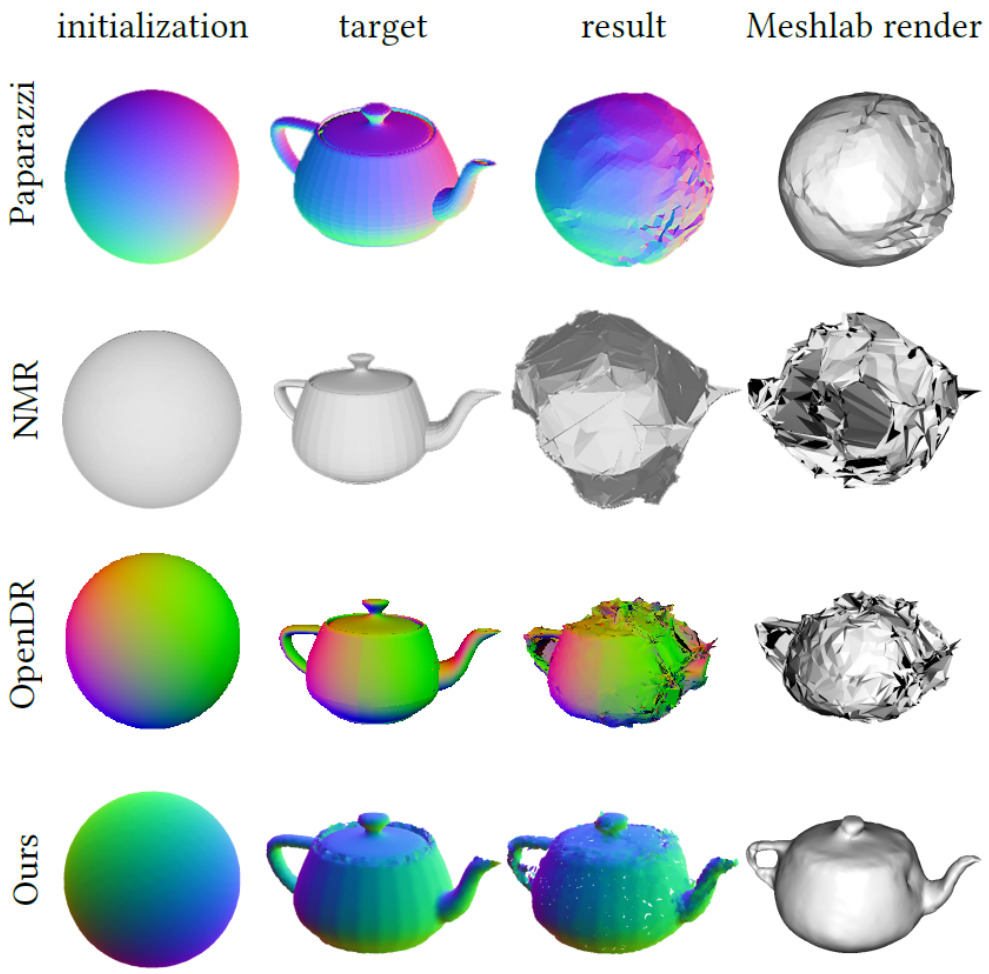


Fig. 10. Large shape deformation with topological changes, compared with three mesh-based DRs, namely Paparazzi [Liu et al. 2018], OpenDR [Loper and Black 2014] and Neural Mesh Renderer [Kato et al. 2018]. Compared to the mesh-based approaches, DSS faithfully recovers the handle and cover of the teapot thanks to the flexibility of the point-based representation.

Results

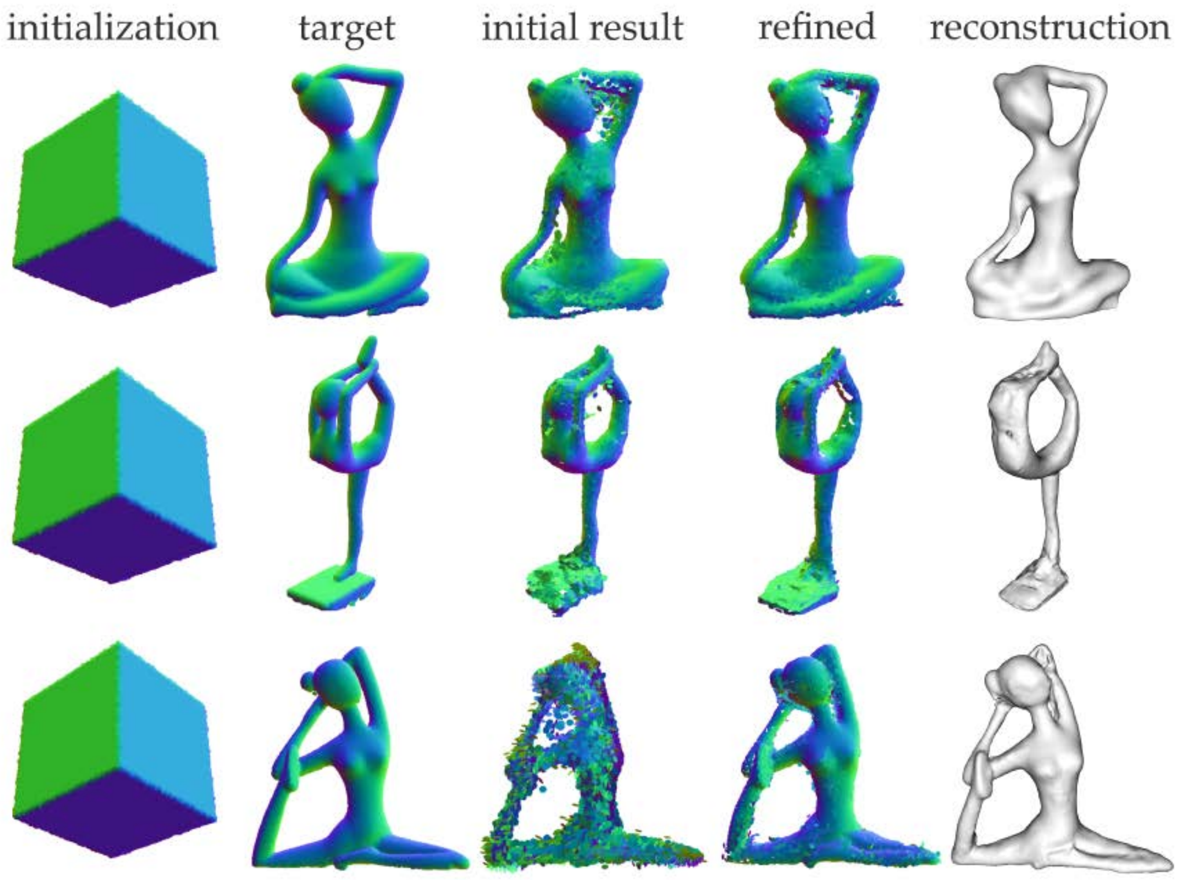


Fig. 11. DSS deforms a cube to three different Yoga models. Noisy points may occur when camera views are under-sampled or occluded (as shown in the initial result). We apply an additional refinement step improving the view sampling as described in Sec. 4.3.

Results

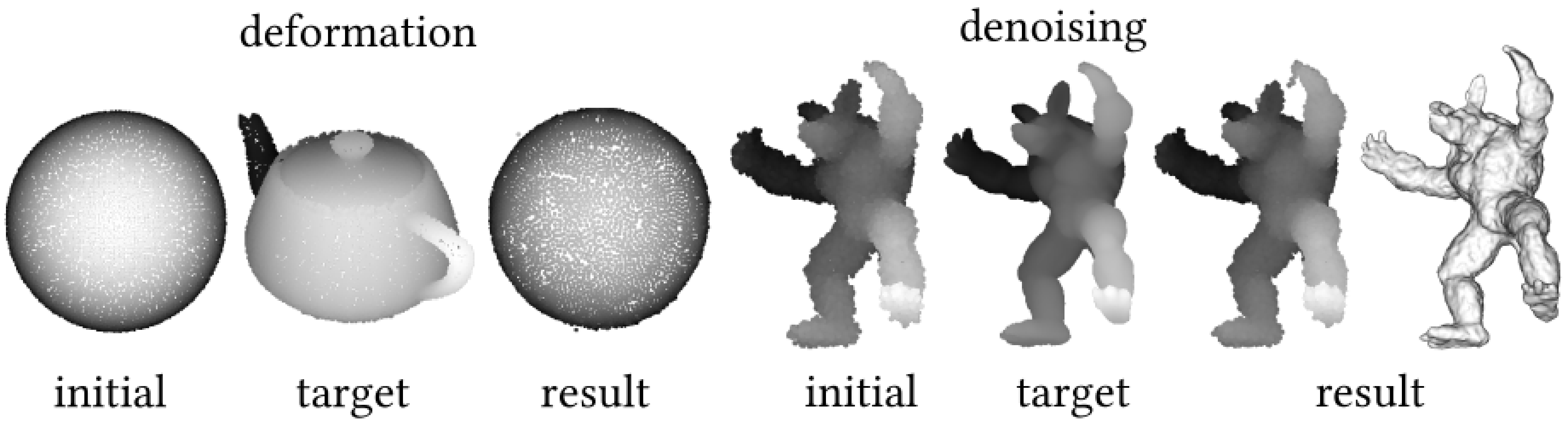


Fig. 12. A simple projection-based point renderer which renders depth values fails in deformation and denoising tasks.

Results - denoising

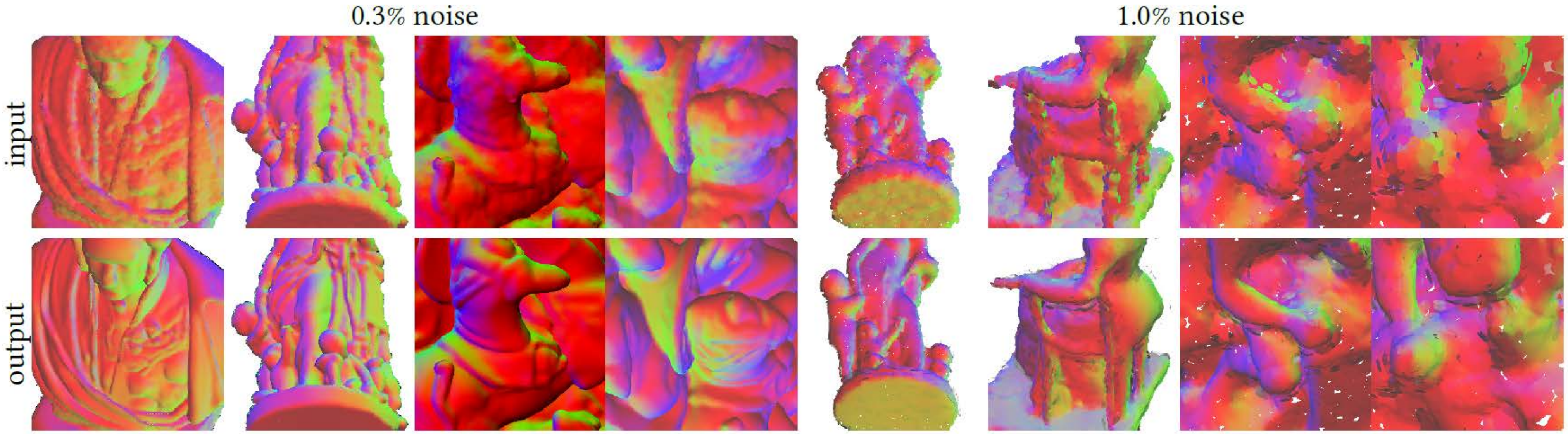


Fig. 15. Examples of the input and output of the Pix2Pix denoising network. We train two models to target two different noise levels (0.3% and 1.0%). In both cases, the network is able to recover smoothly detailed geometry, while the 0.3% noise variant generates more fine-grained details.

Results - denoising

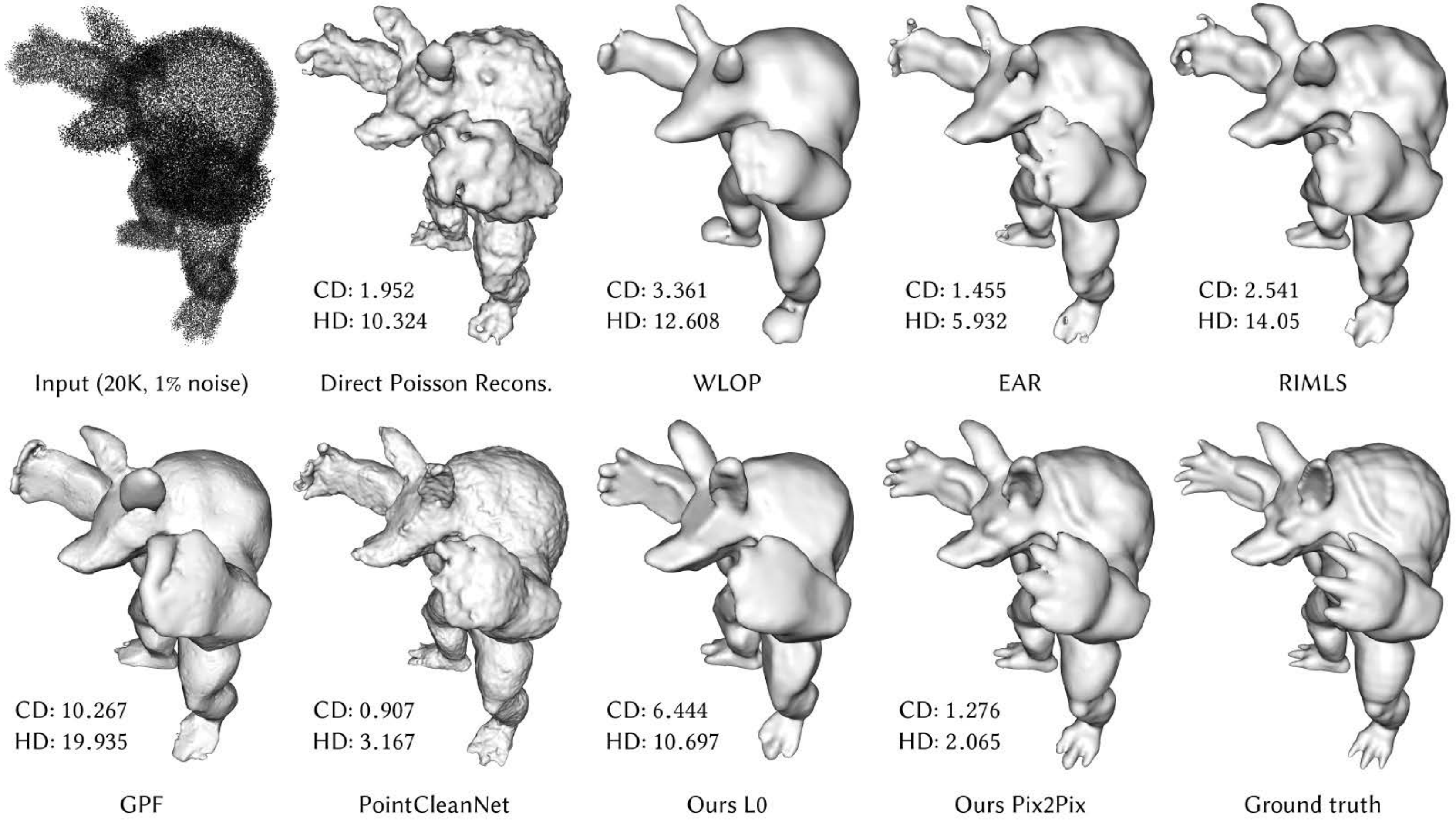


Fig. 17. Quantitative and qualitative comparison of point cloud denoising. The Chamfer distance (CD) and Hausdorff distance (HD) scaled by 10^{-4} and 10^{-3}. With respect to HD, our method outperforms its competitors, for CD only PointCleanNet can generate better, albeit noisy, results.

Discussion

Thank you!