iSDF: Real-Time Neural Signed Distance Fields for Robot Perception

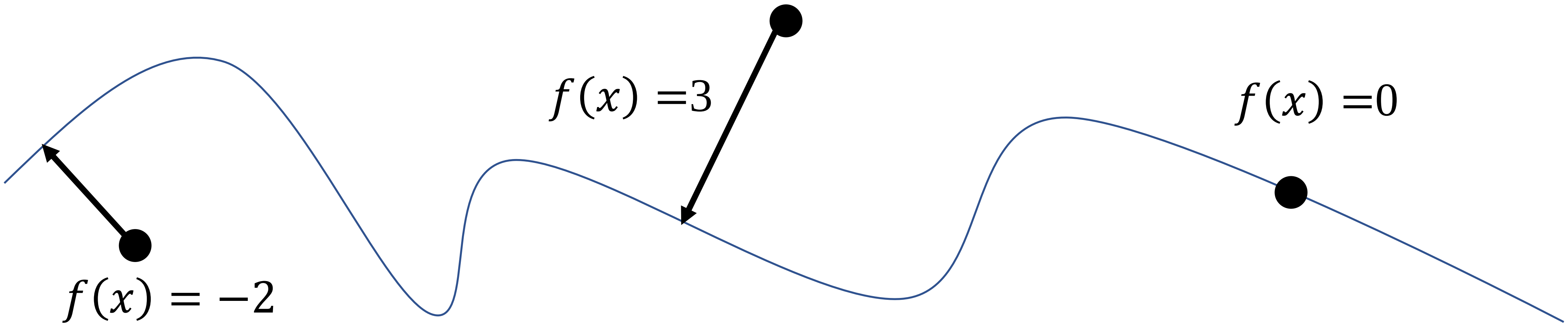
Joseph Ortiz (Imperial College London),
Alexander Clegg (Facebook AI Research),
Jing Dong (Facebook),
Edgar A Sucar (Imperial College London),
David Novotny (Facebook AI Research),
Michael Zollhöfer (Facebook Reality Labs),
Mustafa Mukadam (Facebook AI Research)

presented by Yancheng Liang
07/19/2022

Background: SDF

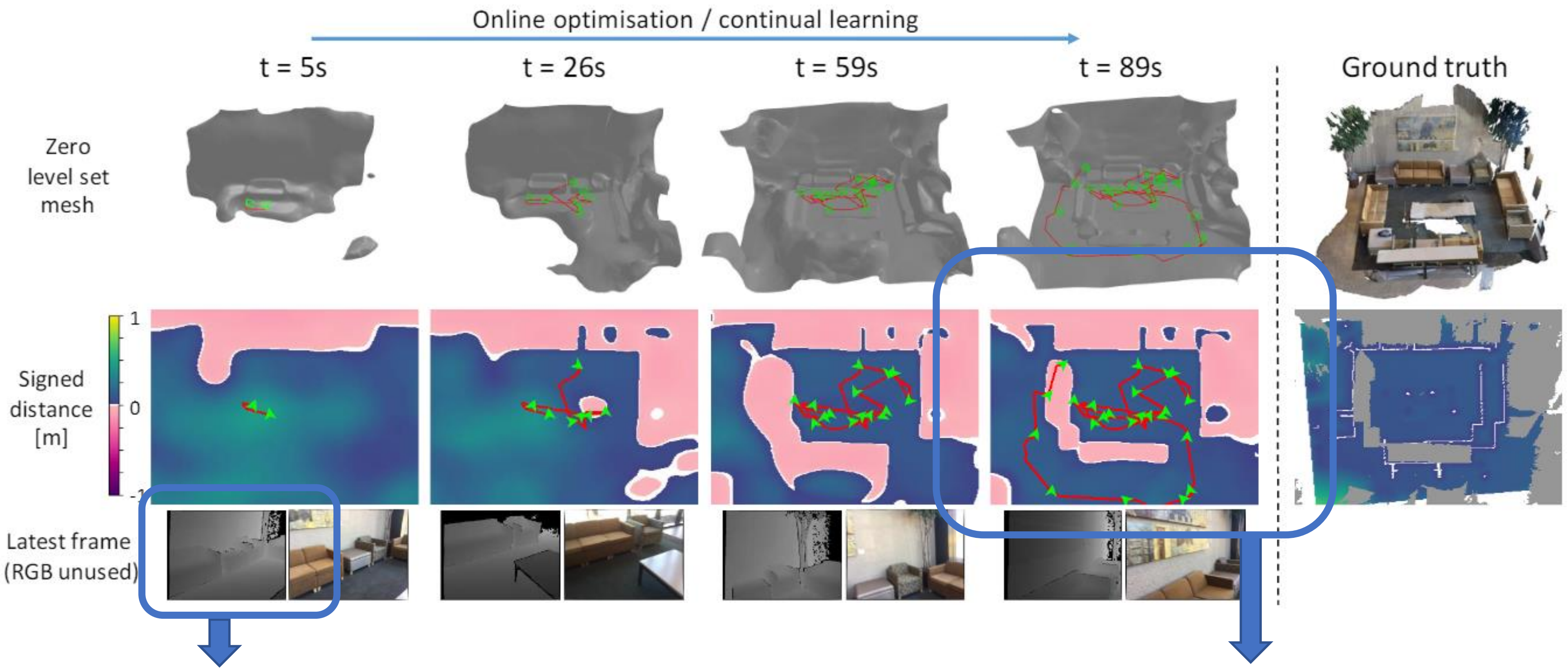
- Signed Distance Fields (SDFs)

- $f(x)$: $R^3 \rightarrow R$ projects a 3D points to a scalar
  - which is the distance to the nearest surface point
  - positive for points in the free space, zero for surface points and negative for points inside objects



(Vision Information) -> (Safety)

Depth Image Stream -> Neural SDF



Vision Information as Input: Depth + RGB

SDF as Output

SDF is represented by a neural network that takes coordinates $x \in R^3$ as input and a scalar $y \in R$ as output

On online learning framework

0. Randomly initialize a NN to represent SDF

Then in each iteration:
1. Collect a new frame of depth image
2. Sample training data from current and past frames
3. Update NN

Prior Works

Update Occupancy Map

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	1
0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	1	1	0

New frame comes

Both are voxel-based, and thus time and space-consuming

Update SDF with online BFS (still voxel-based)

Update TSDF

?	?	?	?	?	2	2	2	2	?
?	?	?	?	2	1	1	1	1	2
?	?	?	2	1	?	0	0	-0	1
?	?	2	1	0	0	-1	-1	-1	0
?	?	2	1	0	-1	?	?	-1	0
?	?	2	1	0	-1	?	?	-1	0
?	?	2	1	0	-1	?	?	-1	0
?	?	2	1	0	-1	?	?	-1	0
?	?	?	2	1	0	-1	-1	-1	0
?	?	?	?	2	1	0	0	0	1

How To Learn SDF f? Four Parts.

- $Loss = E_x[L_{distance}\big(b(x), f(x)\big) + L_{orientation}\big(g(x), \nabla f(x)\big) + L_{property}(f, x)]$

How to sample training points?

How to estimate the true distance $b(x)$?

How to estimate the orientation $g(x)$?

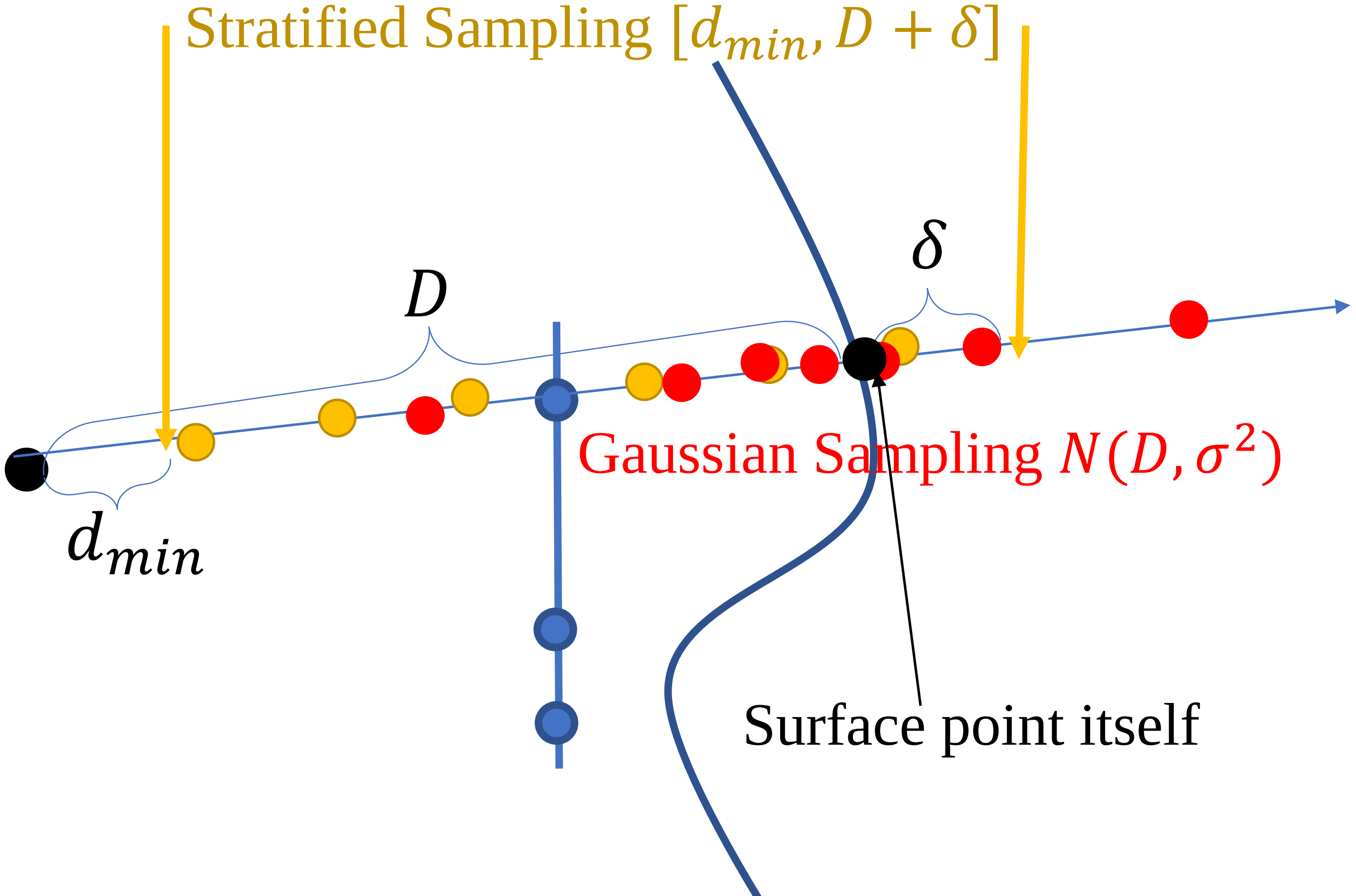
Fact: $|\nabla f(x)| = 1$ almost everywhere

Training Points Batch Sampling

$$Loss = E_x[L_{distance}(b(x), f(x)) + L_{orientation}(g(x), \nabla f(x)) + L_{property}(f, x)]$$

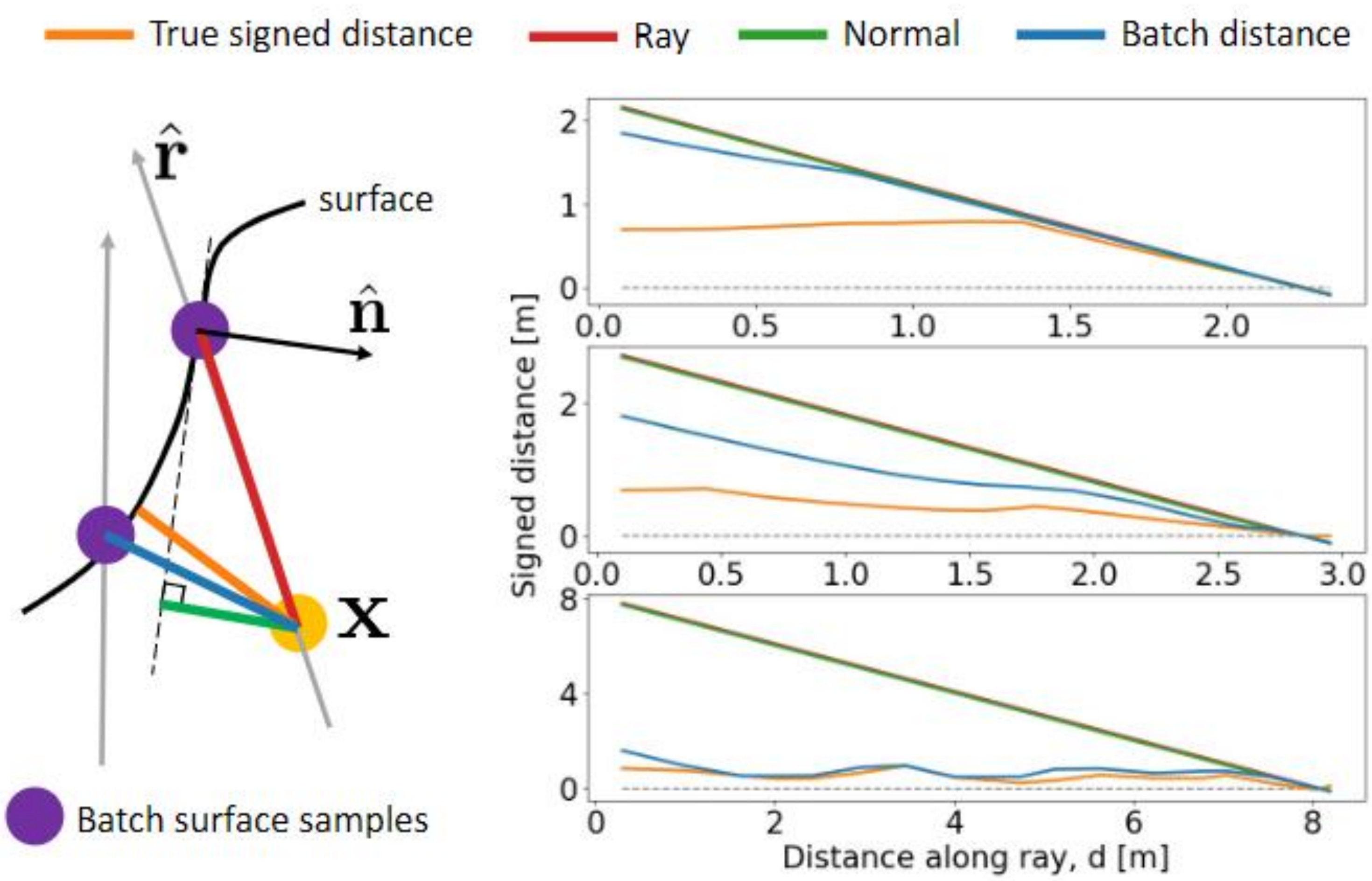
- Frame Selection:
  - Latest frames
  - Select some history keyframes

- In each frame:
  - Randomly sample some pixels
  - Select some points on the ray between the camera and the pixel: Stratified sampling + Gaussian sampling + Surface point



Distance Loss

$$Loss = E_x[L_{distance}(b(x), f(x)) + L_{orientation}(g(x), \nabla f(x)) + L_{property}(f, x)]$$



$b(x) = |x - p|$, p is the estimated closet surface point

1. Use the surface point of the same ray
2. If x is near to the surface point, apply normal correction
3. Choose p among all surface points in this training batch

Distance Loss

$$Loss = E_x[L_{distance}(b(x), f(x)) + L_{orientation}(g(x), \nabla f(x)) + L_{property}(f, x)]$$

$b(x)$ upper bounds $f(x)$, and it should be tight near the surface.

$$\mathcal{L}_{\text{sdf}}(f(\mathbf{x};\theta), b) = \begin{cases} \lambda_{\text{surf}} \mathcal{L}_{\text{near\_surf}} & \text{if } |D[u,v] - d| \leq t \\ \mathcal{L}_{\text{free\_space}} & \text{otherwise.} \end{cases} \quad (8)$$

$$\mathcal{L}_{\text{near\_surf}}(f(\mathbf{x};\theta), b) = |f(\mathbf{x}_i;\theta) - b| \, .$$

$$\mathcal{L}_{\text{free\_space}}(f(\mathbf{x};\theta), b) = \max\left(0,\ e^{-\beta f(\mathbf{x}_i;\theta)} - 1,\ f(\mathbf{x}_i;\theta) - b\right) \, . \quad (6)$$

Orientation Loss

$$Loss = E_x[L_{distance}(b(x), f(x)) + L_{orientation}(g(x), \nabla f(x)) + L_{property}(f, x)]$$

$$\mathbf{g}(\mathbf{x}, \mathcal{P}) = \mathrm{sgn}(D[u, v] - d) \,.\, (\mathbf{x} - \arg\min_{\mathbf{p}\in\mathcal{P}} |\mathbf{x} - \mathbf{p}|)$$

p is the same point chosen previously to estimate $b(x)$

When x is near to a surface, g is replaced by the surface normal

$$\mathcal{L}_{\text{grad}}(\nabla_{\mathbf{x}} f(\mathbf{x}; \theta), \mathbf{g}) = 1 - \frac{\nabla_{\mathbf{x}} f(\mathbf{x}; \theta) \cdot \mathbf{g}}{\|\nabla_{\mathbf{x}} f(\mathbf{x}; \theta)\| \|\mathbf{g}\|}$$

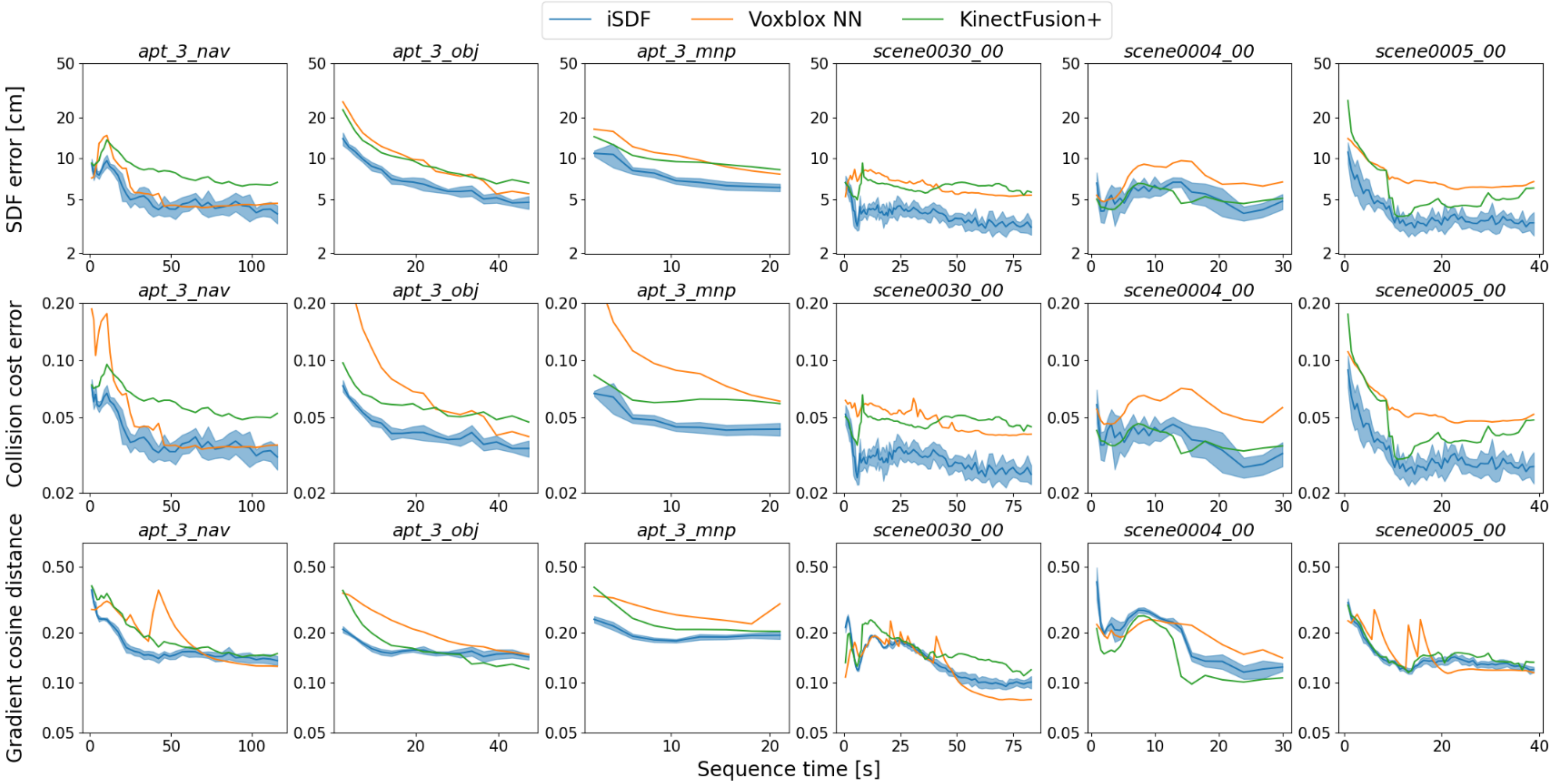
Orientation Loss

$$Loss = E_x[L_{distance}(b(x), f(x)) + L_{orientation}(g(x), \nabla f(x)) + L_{property}(f, x)]$$

$$\mathcal{L}_{\text{eik}}(f(\mathbf{x};\theta)) = \begin{cases} |\, \|\nabla_{\mathbf{x}} f(\mathbf{x};\theta)\| - 1| & \text{if } |D[u,v] - d| \geq a \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

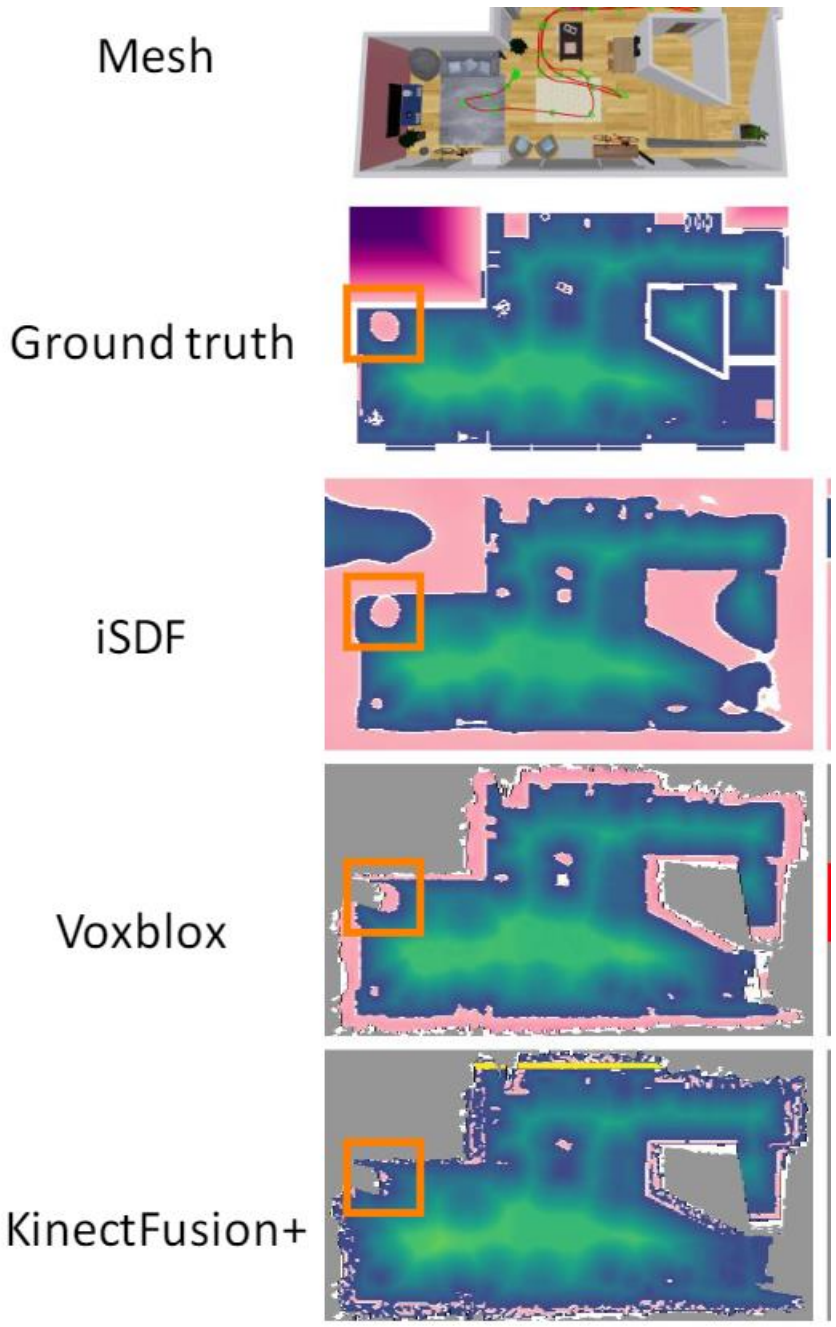
For points that are far from the surface, we regularize $\nabla f(x)$. This can be seen as a BFS propagation.

Numerical Results: Lower Error

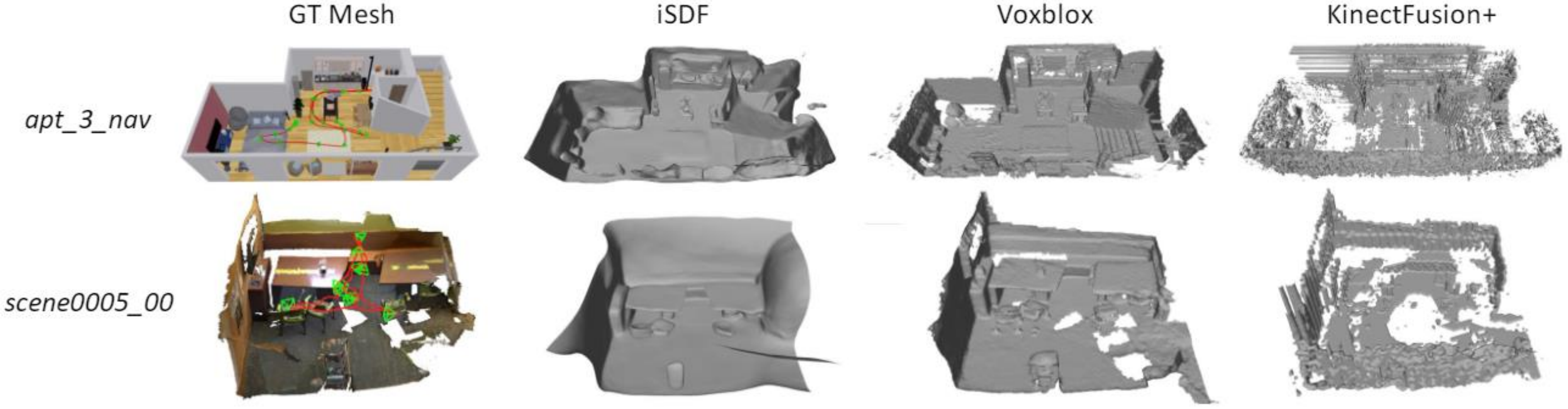


Generalization

- Because iSDF uses NN, it can generalize to unseen regions.



Smoothness



Flexible Resolution

- Voxel-based methods have fixed resolution, while neural representation can attend to both detailed salt shaker and coarse fridge