Neuralangelo: High-Fidelity Neural Surface Reconstruction

- Neuralangelo: High-Fidelity Neural Surface Reconstruction (arxiv.org)
- Neuralangelo: High-Fidelity Neural Surface Reconstruction (nvidia.com)
- bennyguo/instant-nsr-pl: Neural Surface reconstruction based on Instant-NGP. Efficient and customizable boilerplate for your research projects. Train NeuS in 10min! (github.com)

1 Introduction

Two important findings:

- using *numerical gradients* to compute higher-order derivatives, such as surface normals for the eikonal regularization, is critical to stabilizing the optimization
- a *progressive optimization schedule* plays an important role in recovering the structures at different levels of details

2 Related Work

- Multi-view surface reconstruction
- NeRF
- Neural surface reconstruction

3 Methods

3.1 Preliminaries

- Volume rendering of SDF

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{x}_i)) - \Phi_s(f(\mathbf{x}_{i+1}))}{\Phi_s(f(\mathbf{x}_i))}, 0\right)$$

where Φ_s is the sigmoid function

> 相当于当前距离的距离改变量

3.2 Numerical Gradient Computation

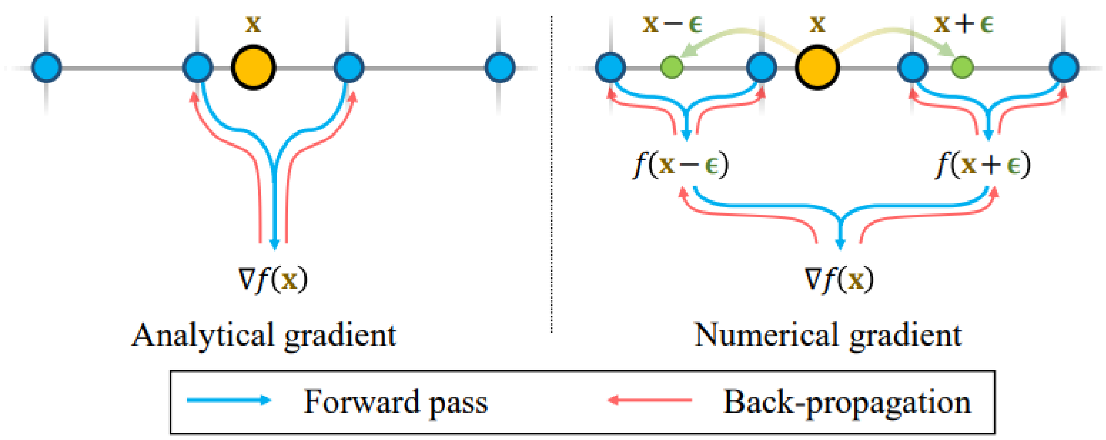

Figure 2. Using **numerical gradients** for higher-order derivatives distributes the back-propagation updates beyond the local hash grid cell, thus becoming a smoothed version of **analytical gradients**.

Problem of *de facto* (traditional) methods using *analytical gradients*:

- Analytical gradients of hash encoding w.r.t. position are **not continuous across space under trilinear interpolation**
- The derivative of hash encoding is local, i.e., when $\mathbf{x}_i$ moves across grid cell borders, the corresponding hash entries will be different
- optimization updates only propagate to **local hash grids**

To overcome using *numerical gradients*:

- If the step size of the numerical gradient ϵ is smaller than the grid size of hash encoding, the numerical gradient would be equivalent to the analytical gradient
- otherwise, hash entries of multiple grid cells would participate in the surface normal computation
- To compute the surface normals using the numerical gradient, additional SDF samples are needed. For example, where $\epsilon_x = [\epsilon, 0, 0]$ and γ is the hash encoding, the x-component of the surface normal can be found as

$$\nabla_x f(\mathbf{x}_i) = \frac{f\big(\gamma(\mathbf{x}_i + \epsilon_x)\big) - f\big(\gamma(\mathbf{x}_i - \epsilon_x)\big)}{2\epsilon}$$

> 即$\frac{\Delta y}{\Delta x}$，不使用无穷小量的求导

> Intuitively, numerical gradients with carefully chosen step sizes can be **interpreted as a smoothing operation** on the analytical gradient expression

3.3 Progressive Levels of Details

> Coarse-to-fine optimization can better shape the loss landscape to avoid falling into false local minima

- step size ϵ
  - Imposing $\mathcal{L}_{eik}$ with a larger ϵ for numerical surface normal computation ensures the surface normal is **consistent at a larger scale**, thus producing consistent and continuous surfaces
  - imposing $\mathcal{L}_{eik}$ with a smaller ϵ **affects a smaller region** and avoids smoothing details
  - initialize the step size ϵ to the coarsest hash grid size and exponentially decrease it matching different hash grid sizes throughout the optimization process
- hash grid resolution V
  - only enable an initial set of coarse hash grids and progressively activate finer hash grids throughout optimization when ϵ decreases to their spatial size.

3.4 Optimization

To further encourage the smoothness of the reconstructed surfaces, we impose a prior by regularizing the mean curvature of SDF:

$$\mathcal{L}_{curv} = \frac{1}{N}\sum_{i=1}^{N}|\nabla^2 f(\mathbf{x}_i)|$$

instead of applying $\mathcal{L}_{curv}$ from the beginning of the optimization process, we use a **short warmup period** that linearly increases the curvature loss strength

4 Evaluation

- Input
  - video
- output
  - geometry

reference

-