[NICE-SLAM: Neural Implicit Scalable Encoding for SLAM]

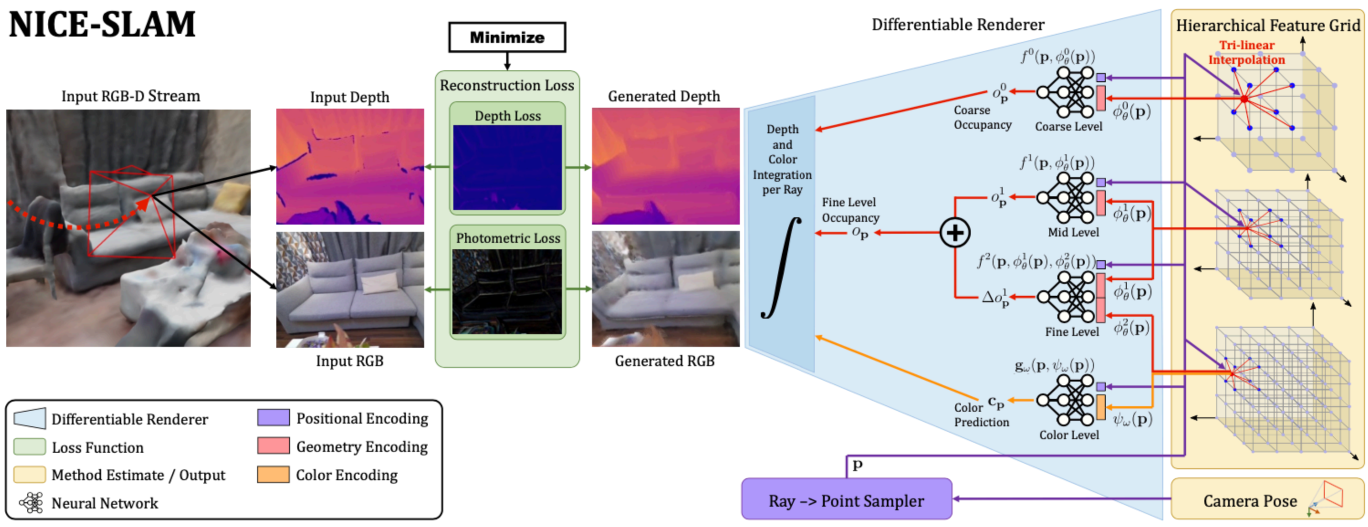
Abstract & Intro:

- SLAM: simultaneous localization and mapping
- Requirement
    - Real-time computation
    - Predictive: Can make prediction for regions without observation
    - Scalable: Can be scaled up to large scenes
    - Robust to noise
- Limitation of current methods:
    - Over smoothed scene reconstruction, difficult to scale up to large scenes
    - Does not incorporate location information in the observations
- Idea:
    - Use multi-level location information (hierarchical scene representation)
    - Incorporate inductive biases of neural implicit decoders pretrained at different spatial resolutions.
    - Minimizing re-rendering losses

Related work:

- World-centric map representation, voxel grids => more accurate geometry at lower grid resolutions
- iMAP (baseline)
- ConvONet

Method:



- Hierarchical Feature Grid: The latent vector in ConvONet
    - Coarse (lowest resolution), mid, fine (highest resolution) grids
    - Each point represents a vector (feature including geometric (θ) and color (ω) information)
    - Level geometrtic representation

$$
\begin{aligned}
o_{\mathbf{p}}^0 &= f^0(\mathbf{p}, \phi_\theta^1(\mathbf{p})) \\
o_{\mathbf{p}}^1 &= f^1(\mathbf{p}, \phi_\theta^1(\mathbf{p})) \\
\Delta o_{\mathbf{p}}^1 &= f^2(\mathbf{p}, \phi_\theta^1(\mathbf{p}), \phi_\theta^2(\mathbf{p})) \\
o_{\mathbf{p}} &= o_{\mathbf{p}}^1 + \Delta o_{\mathbf{p}}^1
\end{aligned} \tag{1}
$$

, where

- $\mathbf{p}$ represents the point location
- occupancy value $o_{\mathbf{p}}$ represents the probability of point $\mathbf{p}$ that it is contained in the surface
- ϕ represents tri-linear interpolation
- f are the neural networks (decoder)
- $o_{\mathbf{p}}^0$ is the occupancy obtained by mid-level which is used to predict the **unobserved part**. Note that for coarse-level, learnable Gaussian positional encoding is used for $\mathbf{p}$
- $o_{\mathbf{p}}^1$ is the occupancy obtained by mid-level
- $o_{\mathbf{p}}^2$ is the residual obtained by a concatenation of the mid-level and fine-level features to capture high-frequency details

- Color representation:

$$
c_{\mathbf{p}} = g_\omega(\mathbf{p}, \psi_\omega(\mathbf{p})) \tag{2}
$$

, where

  - color value $c_{\mathbf{p}}$ represents the estimated color
  - g_ω is a decoder
  - ψ is the tri-linear interpolation of another feature grid

- Depth and color rendering
  - Camera pose defines the camera position $\mathbf{o}$ and direction (unit vector) $\mathbf{r}$
  - N_{strat} samples for stratified sampling (different depths) , and N_{imp} samples near the depth value of the current ray $+(-)0.05D$ along the ray $d_i, i \in \{1, \dots, N\}$

$$
\mathbf{p}_i = \mathbf{o} + d_i \mathbf{r} \tag{3}
$$

  - Calculate the probability of the existence for each point for each level (coarse c, fine f), and the depth and colors are represented as the expectation of the samples:
    - the probability of the existence for each point is represented as the probability that the ray can reach the point

$$
\begin{gathered}
w_i^c = o_{\mathbf{p}_i}^0 \prod_{j=1}^{i-1}(1 - o_{\mathbf{p}_j}^0), w_i^f = o_{\mathbf{p}_i} \prod_{j=1}^{i-1}(1 - o_{\mathbf{p}_j}) \\
\hat{D}^c = \sum_{i=1}^{N} w_i^c d_i, \hat{D}^f = \sum_{i=1}^{N} w_i^f d_i, \hat{I} = \sum_{i=1}^{N} w_i^f c_i
\end{gathered} \tag{4}
$$

    - the variance is also calculated:

$$
\hat{D}_{var}^c = \sum_{i=1}^{N} w_i^c (\hat{D}^c - d_i)^2, \hat{D}_{var}^f = \sum_{i=1}^{N} w_i^f (\hat{D}^f - d_i)^2 \tag{5}
$$

- Optimization

- Pretrained decoder: The decoders f are trained separately as the decoder part of ConvONet. Note the difference is that f^2 is trained by a concatenated feature. This is fixed during optimization.
- Loss
  - Geometric loss

$$\mathcal{L}_g^l = \frac{1}{M}\sum_{m=1}^{M}|D_m - \hat{D}_m^l|, l \in \{c, f\}. \tag{6}$$

  - Photometric loss

$$\mathcal{L}_p = \frac{1}{M}\sum_{m=1}^{M}|I_m - \hat{I}_m|. \tag{7}$$

  - Modified geometric loss

$$\mathcal{L}_{g\_var} = \frac{1}{M_t}\sum_{m=1}^{M_t}\frac{|D_m - \hat{D}_m^c|}{\sqrt{\hat{D}_{var}^c}} + \frac{|D_m - \hat{D}_m^f|}{\sqrt{\hat{D}_{var}^f}}. \tag{8}$$

- Reconstruction
  - First: Optimize mid-level ϕ_θ^1 Using $\mathcal{L}_g^f$
  - Second: Optimize ϕ_θ^1, ϕ_θ^2 features with the same fine-level depth loss $\mathcal{L}_g^f$
  - Third: Optimize feature grids at all levels and **color decoder** using the following loss

$$\min_{\theta,\omega}(\mathcal{L}_g^c + \mathcal{L}_g^f + \lambda_p\mathcal{L}_p) \tag{9}$$

- Camera Tracking: optimize modified geometric loss $\mathcal{L}_{g\_var}$

$$\min_{\mathbf{R},\mathbf{t}}(\mathcal{L}_{g\_var} + \lambda_{pt}\mathcal{L}_p) \tag{10}$$

- Robustness to Dynamic Objects: remove pixel from optimization if the loss is larger than 10 times of the median loss of all pixels

- Keyframe selection: only incloude keyframes which have visual overlap with the current cframe when optimizing the scene geometry => only optimize necessary parameters.