NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Juhyun Lee
2021. 08.
OEQE Lab
Seoul national university

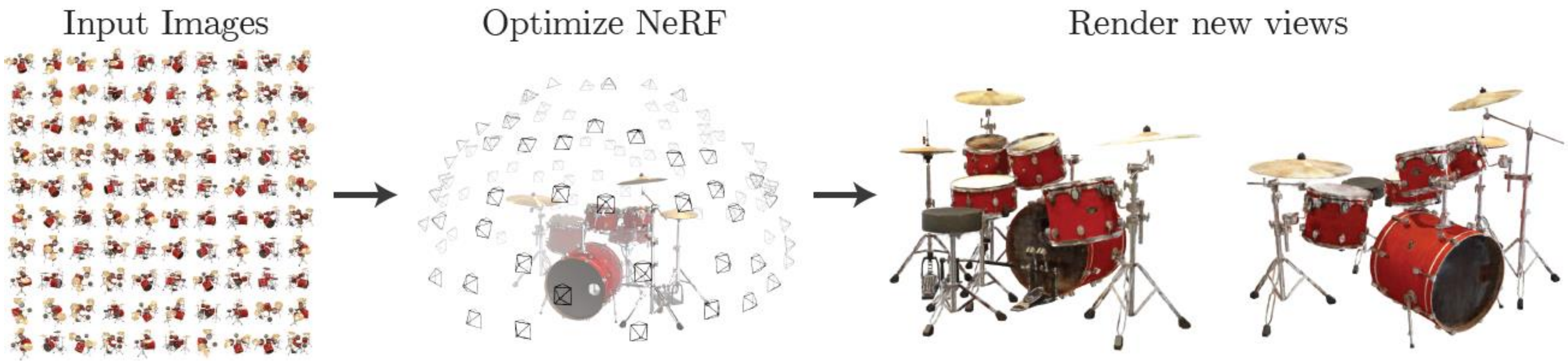
Optical Engineering and Quantum Electronics Lab.,
Electrical and Computer Engineering,
Seoul National University, Republic of Korea

Contents

- Abstract
- Main Idea
- Volume rendering
- Implementation - positional encoding and network architecture
- Hierarchical volume sampling
- Results

NeRF

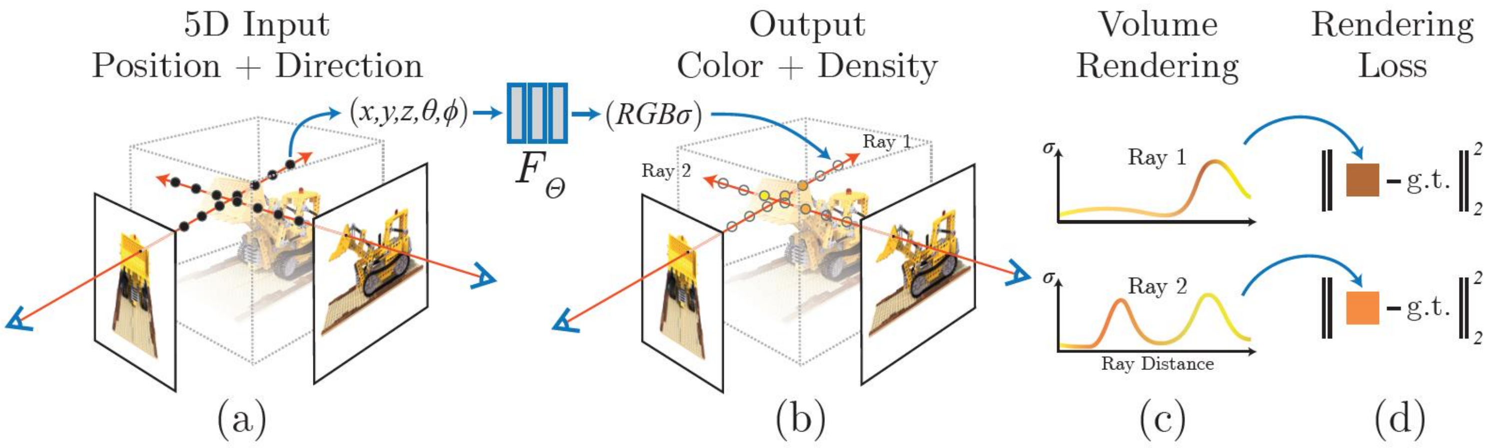
B. Mildenhall, P. Srinivasan, and M. Tancik, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” in Proceedings of European Conference on Computer Vision, (Springer, 2020).



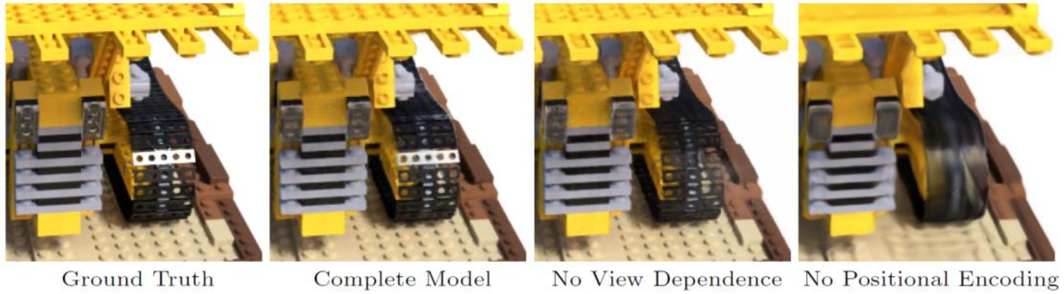
- Neural radiance fields (NeRF): scene representation using neural network
- Input images rendered from discretized viewpoint
  → Rendering new views which is not included in training dataset

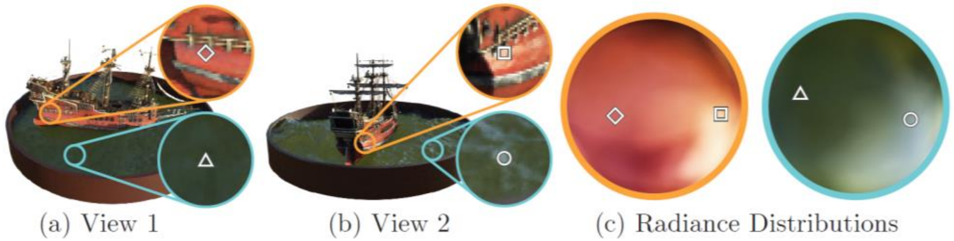
NeRF

B. Mildenhall, P. Srinivasan, and M. Tancik, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” in Proceedings of European Conference on Computer Vision, (Springer, 2020).



- N of 2D image → Synthesizing random view image
- $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$
  Replacing with deep neural network
- **c** : with respect to location **x** and viewing direction **d**
  $\boldsymbol{\sigma}$: with respect to location **x**
- Continuous radiance field is synthesized for the changing in viewing direction.





NeRF

B. Mildenhall, P. Srinivasan, and M. Tancik, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in Proceedings of European Conference on Computer Vision, (Springer, 2020).
N. Max, "Optical models for direct volume rendering," IEEE Trans. Vis. Comp. Graph. 1, 99–108 (1995).

Volume rendering

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\underbrace{\sigma(\mathbf{r}(t))}_{\text{density}}\underbrace{\mathbf{c}(\mathbf{r}(t),\mathbf{d})}_{\text{RGB}}dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right).$$

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\; t_n + \frac{i}{N}(t_f - t_n)\right].$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right), \delta_i = t_{i+1} - t_i$$

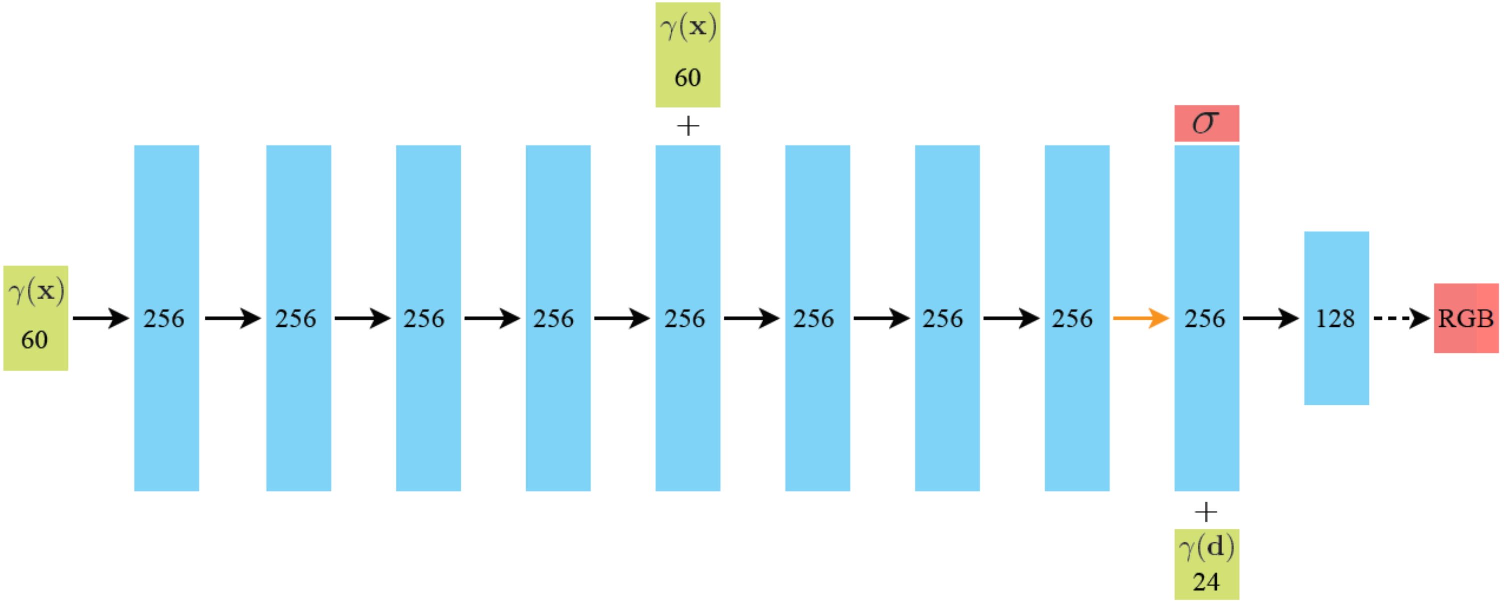
- $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ → ray
- $\sigma(\mathbf{r}(t))$ → the differential probability of a ray terminating at an infinitesimal particle at location $\mathbf{x}$
- T(t) → accumulated transmittance, the probability that the ray travels from t_n to t without hitting any other particle.
- Deterministic quadrature limit resolution of NeRF → stratified sampling

NeRF

B. Mildenhall, P. Srinivasan, and M. Tancik, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in Proceedings of European Conference on Computer Vision, (Springer, 2020).

Implementation – positional encoding and network architecture

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right).$$



- Mapping the inputs to a higher dimensional space using high frequency functions before passing them to the network → better fitting of data that contains high frequency variation
- This paper set $L = 10$ for $\gamma(\mathbf{x})$ and $L = 4$ for $\gamma(\mathbf{d})$.

NeRF

B. Mildenhall, P. Srinivasan, and M. Tancik, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in Proceedings of European Conference on Computer Vision, (Springer, 2020).

Hierarchical volume sampling

- Sample N_c locations using stratified sampling → evaluate **"coarse"** network
- Given "coarse" network, producing a more informed sampling of points N_f → evaluate **"fine"** network
- Final rendered color of the ray $\hat{C}_f(\mathbf{r})$ is computed using $N_c + N_f$ samples.

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i \,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i)) \,, \quad \hat{w}_i = w_i / \sum_{j=1}^{N_c} w_j$$

Implementation details

- At each optimization iteration,
  randomly sampling camera rays → hierarchical sampling $N_c + N_f$ → volume rendering → computing loss
- $\mathcal{L} = \sum_{r \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$
- 4096 rays, $N_c = 64, N_f = 128$.
- The optimization for a single scene typically takes around 100-300k iterations to converge on a single Nvidia V100 GPU (about 1-2 days)

NeRF

B. Mildenhall, P. Srinivasan, and M. Tancik, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” in Proceedings of European Conference on Computer Vision, (Springer, 2020).

Results