Neural Rendering in the Wild

조민지

Vision & Display Systems Lab.

Dept. of Electronic Engineering, Sogang University

Outline

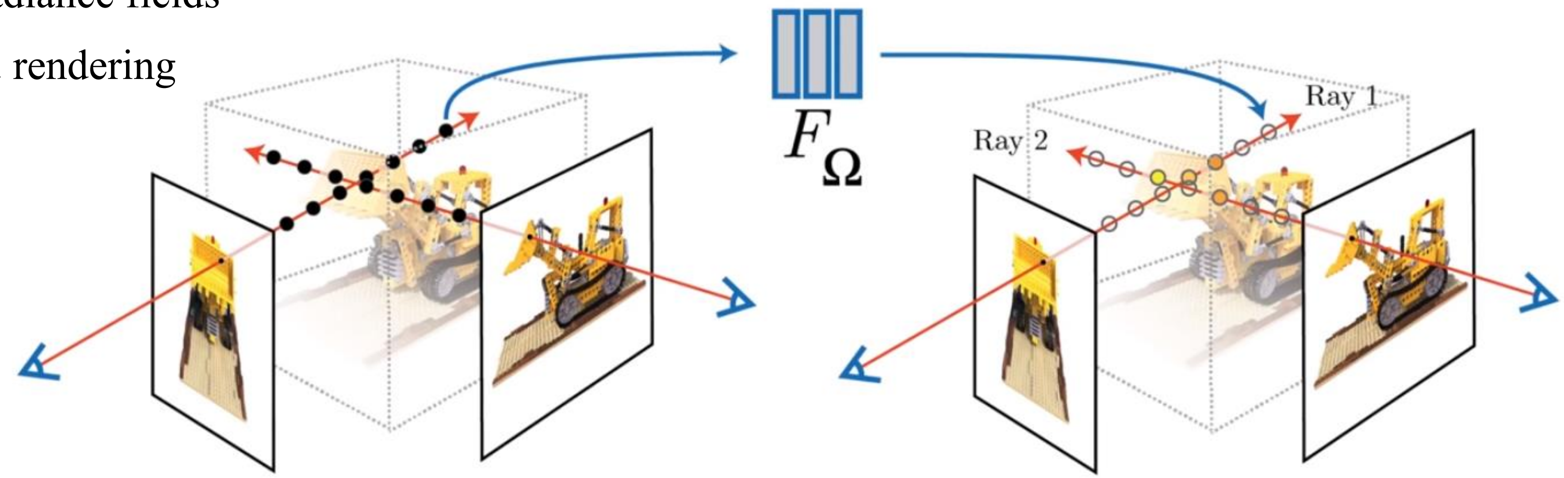
- Intro
- Prior Work
  - NeRF : Representing Scenes as Neural Radiance Fields for View Synthesis (2020 ECCV)
  - Limitations for photos in the wild
- NeRS: Neural Reflectance Surfaces for Sparse-view 3D Reconstruction in the Wild (2021 NIPS)
- NeRF in the Wild : Neural Radiance Fields for Unconstrained Photo Collections (2021 CVPR)

Intro

- 3D Rendering
  and View synthesis

Prior Work

- NeRF : Neural Radiance Field Scene Representation
  - Neural Network as scene representation
    - Volume rendering with neural radiance fields
    - View synthesis and image-based rendering

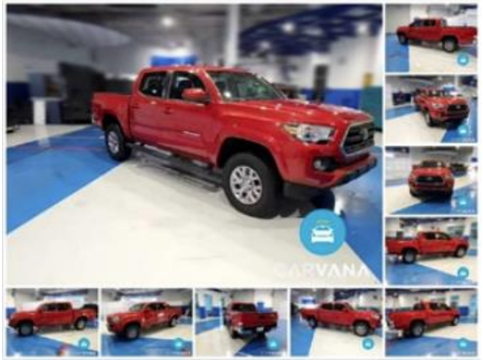


$$\min_{\Omega} \sum_i \|\text{render}^{(i)}(F_\Omega) - I_{\text{gt}}^{(i)}\|^2$$

Prior Work

- NeRF : Neural Radiance Field Scene Representation
  - Limitations for photos in the wild
    - NeRF struggles to generalize when trained with sparse views
      - ∵ Geometry and appearance is **arbitrary**

Few training images with sparse views



Reconstructed by NeRF

Prior Work

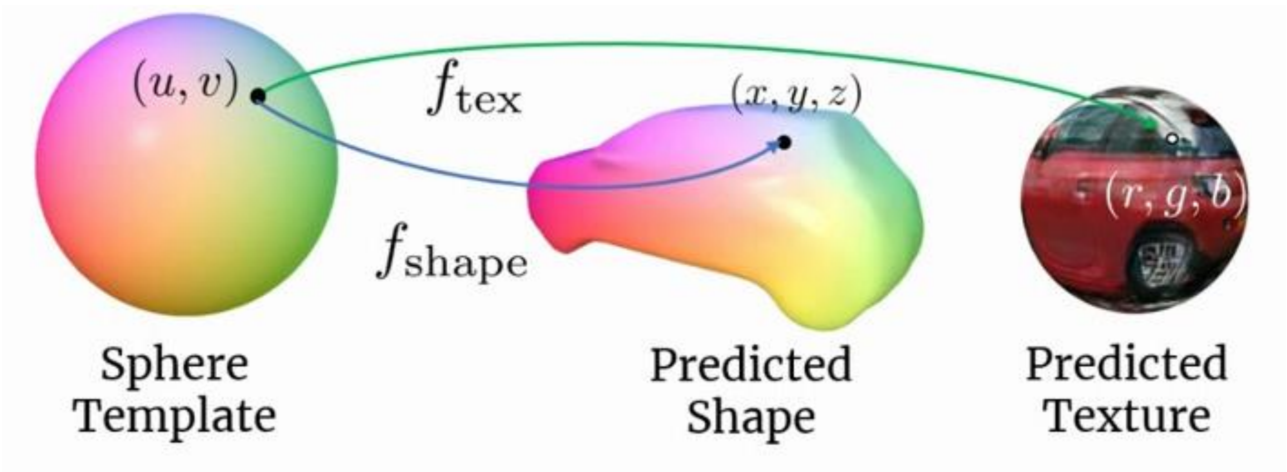
- NeRF : Neural Radiance Field Scene Representation
  - Limitations for photos in the wild
    - NeRF results in errors when applied to photos with inaccurate poses
      - ∵ **Strict** consistency assumptions of NeRF
      - COLMAP fails to recover meaningful poses & reconstructions,
        - ✓ when given wide-baseline input images
        - ✓ when objects have less texture information

Prior Work

- NeRF : Neural Radiance Field Scene Representation
  - Limitations for photos in the wild
    - NeRF results in inaccuracies when applied to photos in the wild
      - ∵ NeRF assumes the scene to be **static**, but this does note hold in the wild data

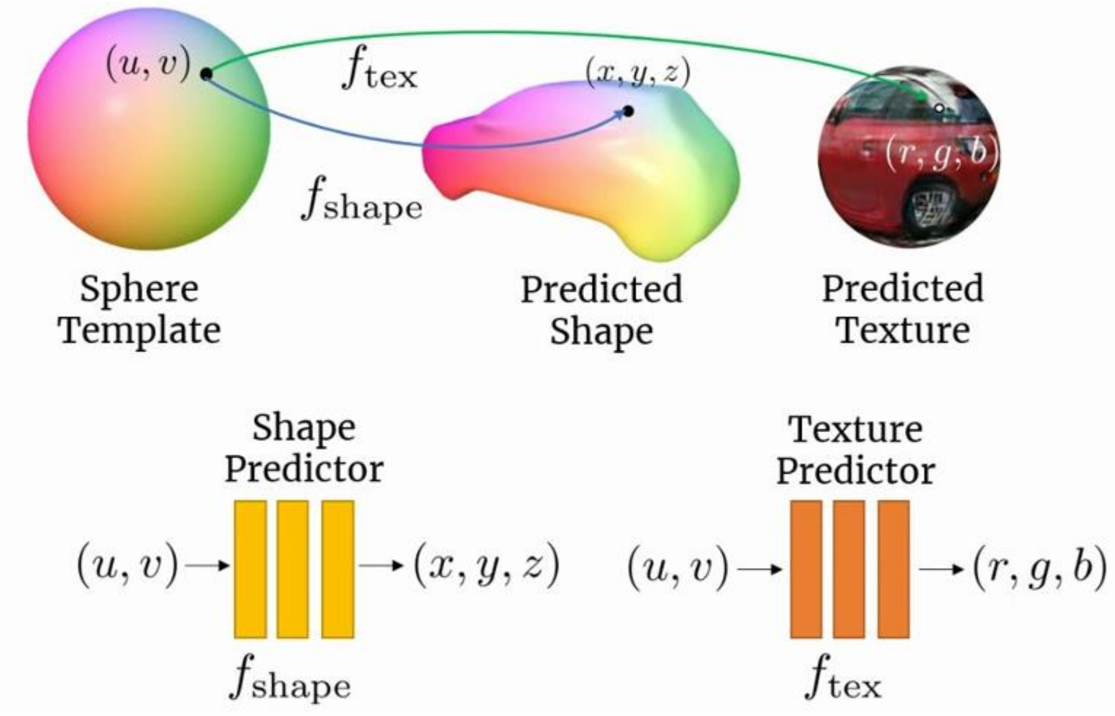
NeRS: Neural Reflectance Surface Representation

- Goal: 3D reconstruction from sparse views
  - NeRS enforces watertight and closed manifolds (=surfaces)
    - $\Longrightarrow$ Geometry and appearance are constrained to surface



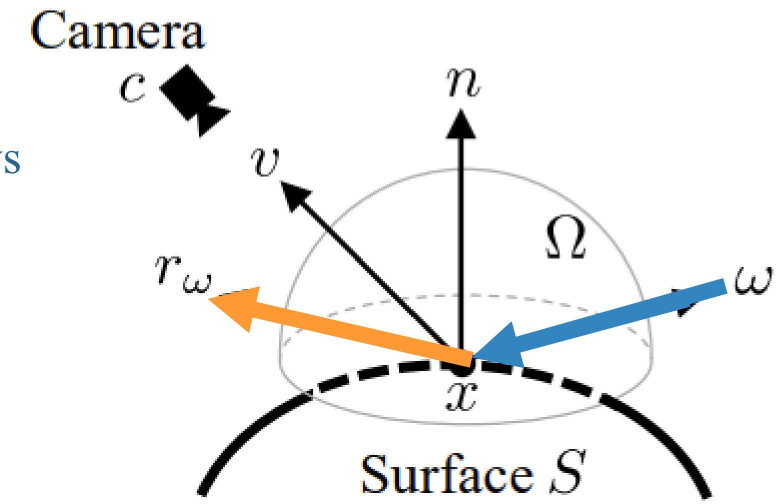
NeRS: Neural Reflectance Surface Representation

- Representing neural surfaces
  - Continuous representation of shape and texture
    - Shape is deformed from a unit sphere via f_{shape}
      - $f_{shape}(u, v) = (x, y, z)$
    - Texture follows implicit per-uv color value via f_{tex}
      - $f_{tex}(u, v) = (r, g, b)$



NeRS: Neural Reflectance Surface Representation

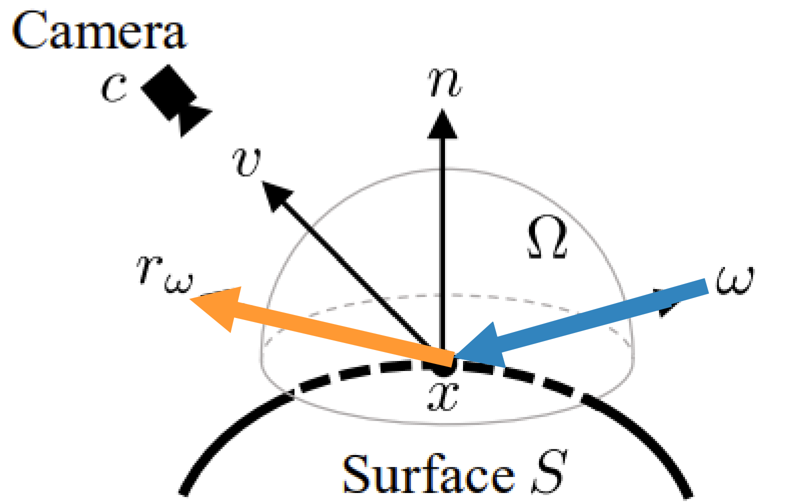
- Rendering view-dependent effects
  - Same point can have different appearance (=illumination) by views
    - Due to reflection caused by surface material properties
  - Radiance L_0 described by rendering equation
    - Diffuse components & reflection components



$$L_o(x, v) = \int_\Omega \boxed{\underbrace{f_r(x, v, \omega}_{\text{BRDF}}} \boxed{\underbrace{L_i(x, \omega)}_{\text{Incoming Radiance}} \underbrace{(\omega \cdot n)}_{\text{Cosine Reduction}}} d\omega$$

NeRS: Neural Reflectance Surface Representation

- Rendering view-dependent effects
  - Rendering using Phong Reflection
    - Decompose illumination into diffuse & specular components
  - View-dependent specular components
    - α means "mirror-ness", k_s means intensity of the specular highlight



$$L_o(x,v) = \int_\Omega \boxed{f_r(x,v,\omega)} \boxed{L_i(x,\omega)(\omega\cdot n)} d\omega$$

$$\approx \boxed{T(x) I_{\text{diffuse}}(x)} + \boxed{k_s I_{\text{specular}}(x,v)}$$

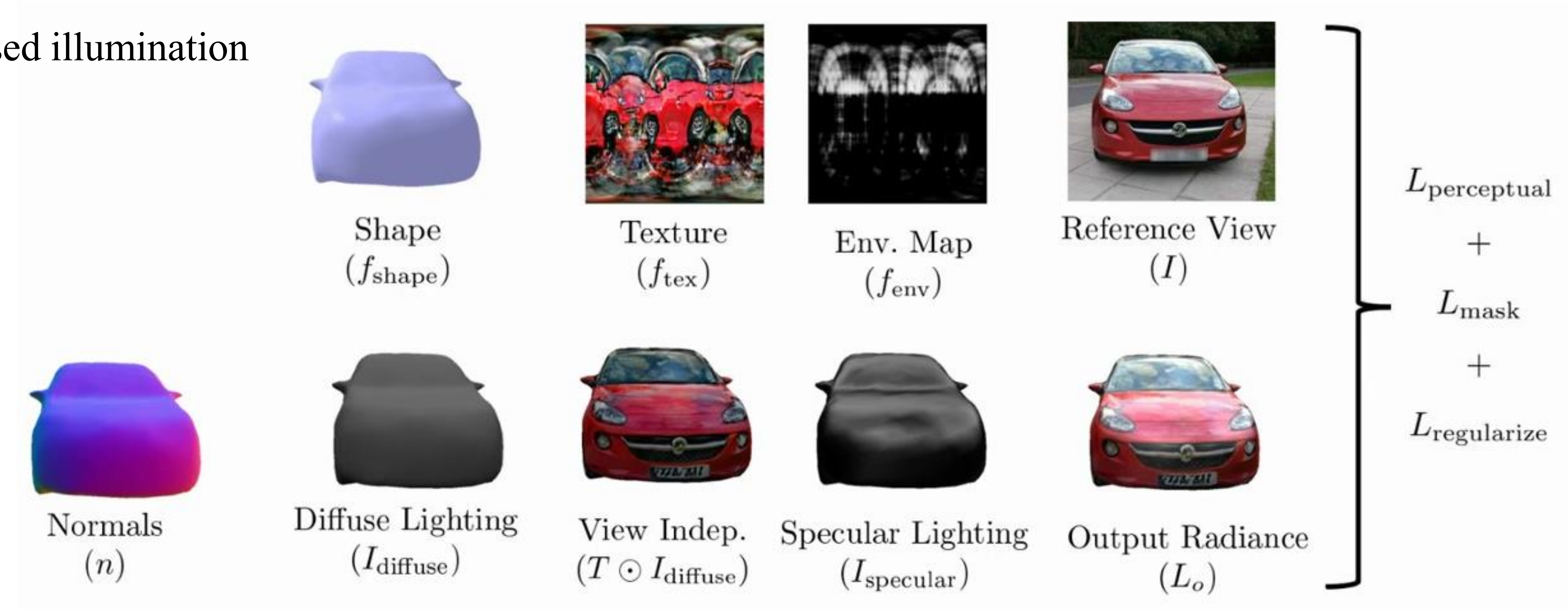
$$I_{\text{diffuse}}(x) = \sum_{\omega\in\Omega} (\omega\cdot n) L_i(\omega)$$

$$I_{\text{specular}}(x,v) = \sum_{\omega\in\Omega} (r_{\omega,n}\cdot v)^\alpha L_i(\omega)$$

$$T(x) = f_{tex}$$

NeRS: Neural Reflectance Surface Representation

- Surface-based illumination



NeRS: Neural Reflectance Surface Representation

- Evaluation

Training views

NeRS

IDR

NeRF

NeRS: Neural Reflectance Surface Representation

- Evaluation

Training views

NeRS
w/o view-dependence

NeRS

Illumination of
mean texture

NeRS: Neural Reflectance Surface Representation

- Conclusion
  - Enforced watertight and closed manifolds
    - Allows to model surface-based appearance affects like View-dependent specularities
  - Enabling reconstruction of objects with diverse material properties & learning from sparse in-the-wild multi-view data
  - But… there still exist brightness ambiguity btw texture and lighting
    - Ex. Gray car under dark env. is predicted to be dark gray color under bright env.

≠

NeRF for Unconstrained Photo Collections

- Baseline : NeRF
  - Representing a scene as a continuous 5D (or 6D) function

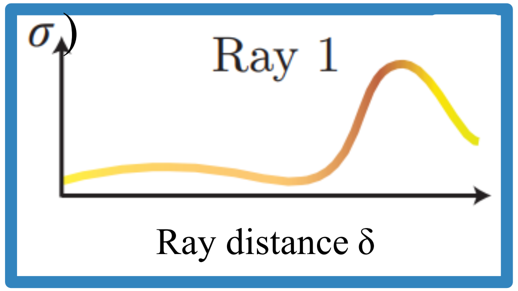
$$(\overbrace{x, y, z}^{\text{Spatial location } \vec{x}}, \underbrace{\theta, \emptyset}_{\substack{\text{Viewing direction } \vec{d} \\ \text{(2D or 3D)}}}) \rightarrow F_{\Omega} \rightarrow (\underbrace{r, g, b}_{\text{Output color } \vec{c}}, \overbrace{\sigma}^{\text{Output density}})$$

NeRF for Unconstrained Photo Collections

- Baseline : NeRF
  - Generate views with traditional volume rendering
    - Optimize every ray with gradient descent (l2 loss)

(c, σ)

Color (RGB), Volume density (Opacity)



: $C(i)$

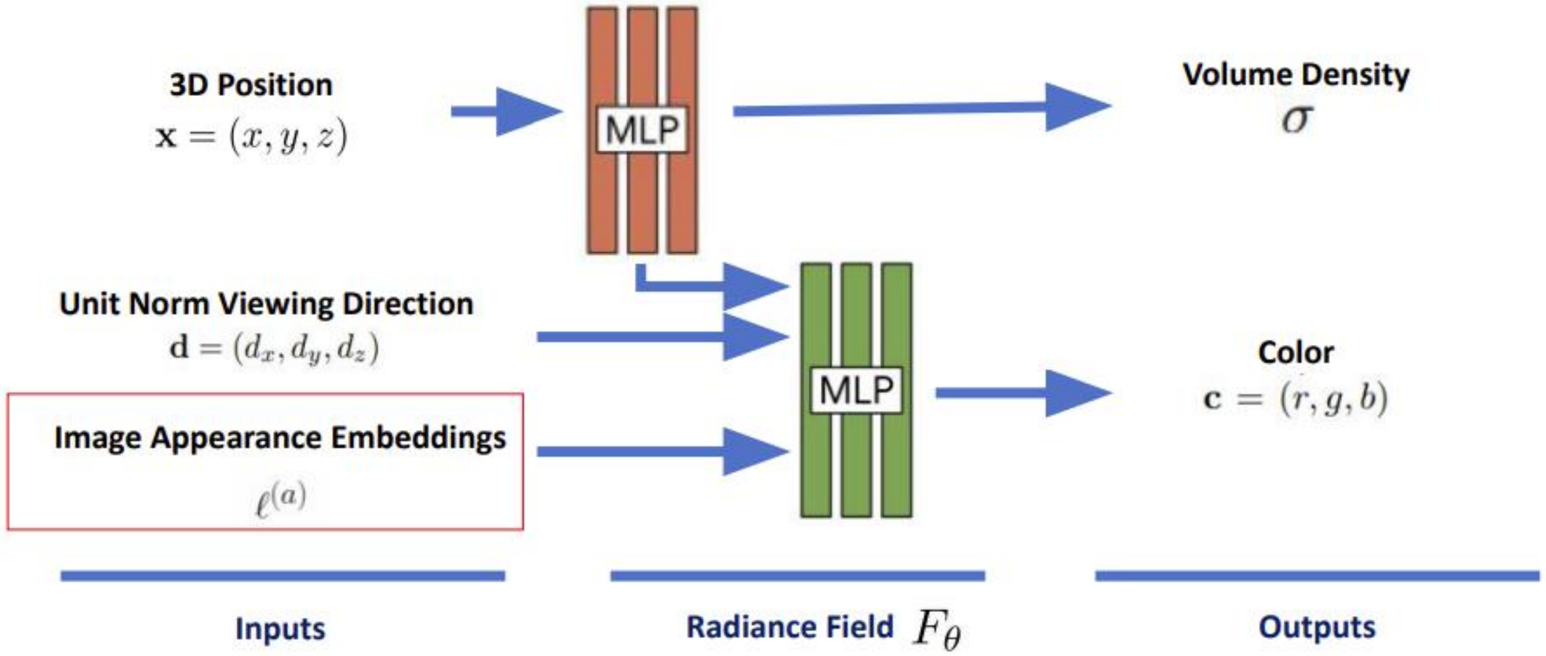
$$\min_{\Omega} \sum_i \| \mathrm{render}^{(i)}(F_\Omega) - I_{\mathrm{gt}}^{(i)} \|^2$$

: $\hat{C}(i) = \sum_{j=1}^{N} T_j \left(1 - \exp(-\sigma_j \delta_j)\right) c_j$

$T_j = \exp(-\sum_{k=1}^{j-1} \sigma_j \delta_j)$

NeRF for Unconstrained Photo Collections

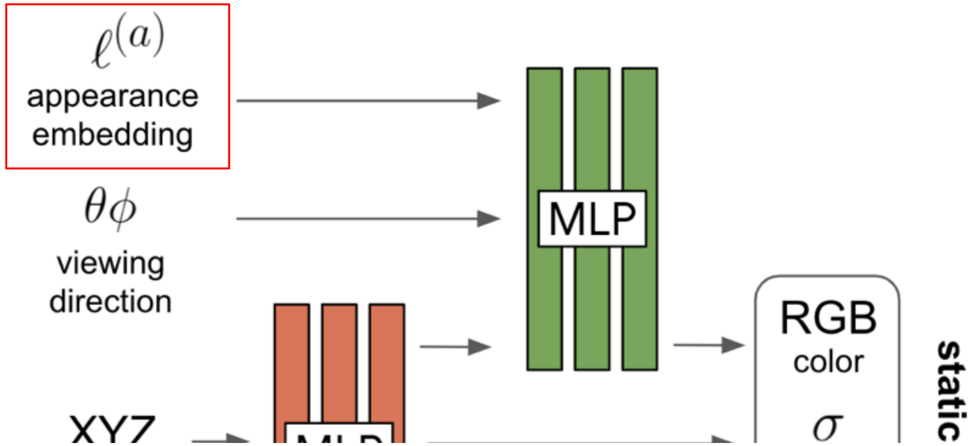
- Latent appearance modeling
  - Adapts NeRF to variable lightning and photometric changes
    - Introdu



)

NeRF for Unconstrained Photo Collections

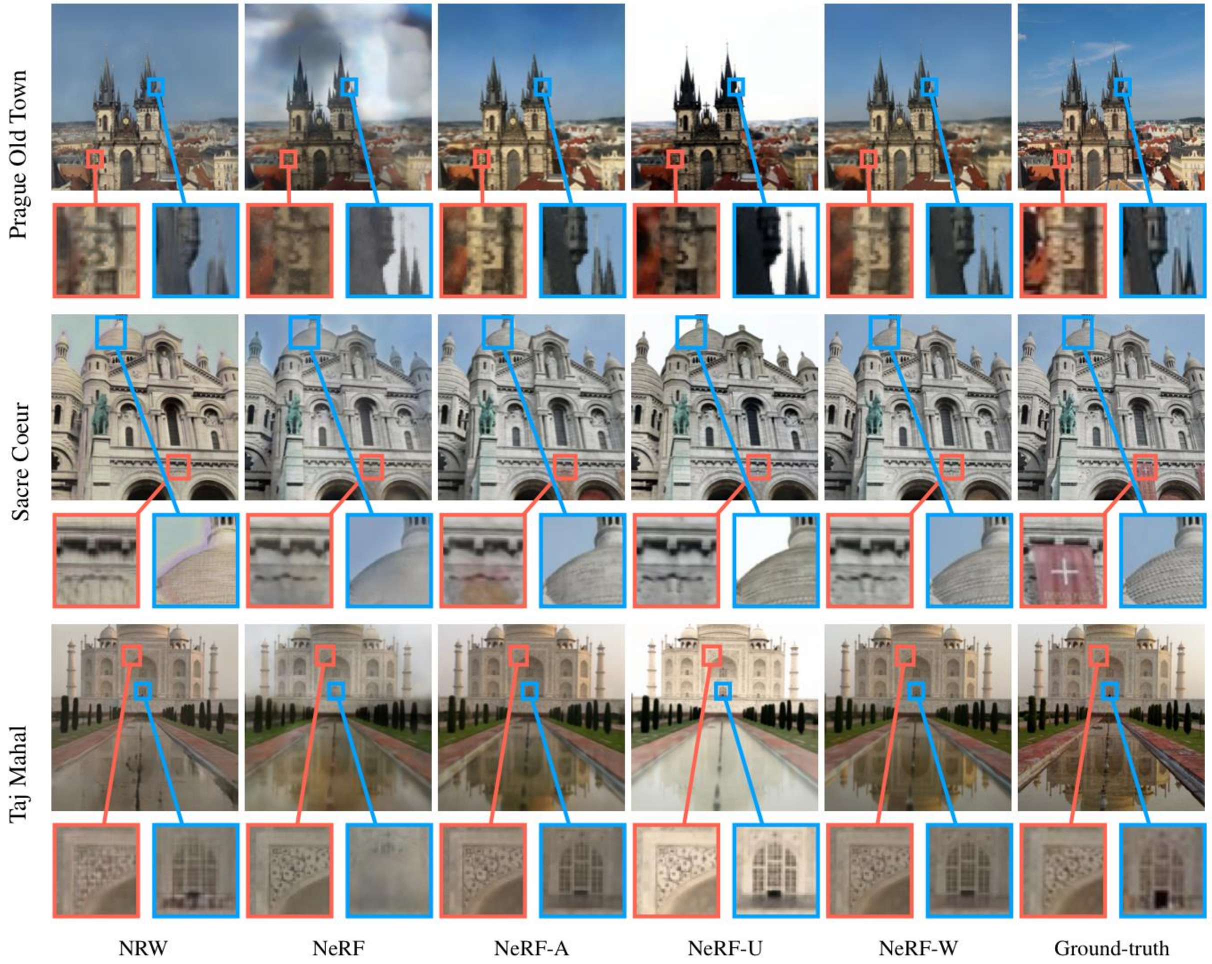
- Transient objects
  - Use ‘static’ and ‘transient’ heads of NeRF baseline
    - Two models disentangle static and transient phenomena without explicit supervision



(a) Static (b) Transient (c) Composite (d) Image (e) Uncertainty

NeRF for Unconstrained Photo Collections

- Evaluation



감사합니다