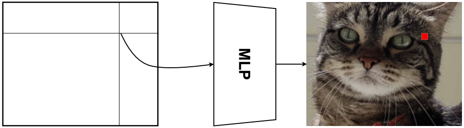
Neural Radiance Fields

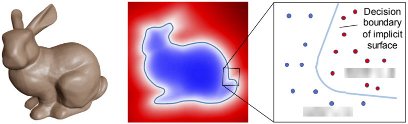
Loic Landrieu

Sep 2021

Neural Fields

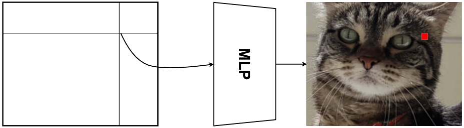
- **Implicit Representation:** F continuous function mapping spatial coordinates to colour / occupancy:
- **Image:** $F : \mathrm{R}^2 \mapsto \mathrm{R}^3$: pixel to RGB.
- **3D :** $F : \mathrm{R}^2 \mapsto \mathrm{R}^4$: RGB + occupancy.

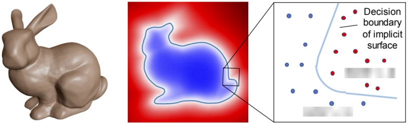




Neural Fields

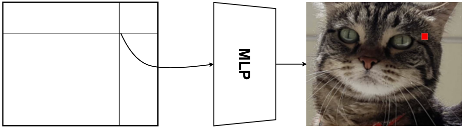
- **Implicit Representation:** F continuous function mapping spatial coordinates to colour / occupancy:
  - **Image:** $F : R^2 \mapsto R^3$: pixel to RGB.
  - **3D :** $F : R^2 \mapsto R^4$: RGB + occupancy.
- **Benefits:**
  - Infinite resolution
  - Adaptive to complexity
  - Can be learned
  - Generalization

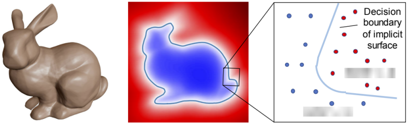




Neural Fields

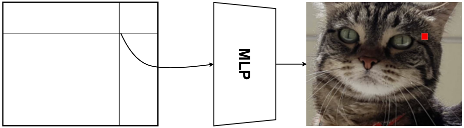
- **Implicit Representation:** F continuous function mapping spatial coordinates to colour / occupancy:
  - **Image:** $F : R^2 \mapsto R^3$: pixel to RGB.
  - **3D :** $F : R^2 \mapsto R^4$: RGB + occupancy.
- **Benefits:**
  - Infinite resolution
  - Adaptive to complexity
  - Can be learned
  - Generalization
- **Converting to other Format:**

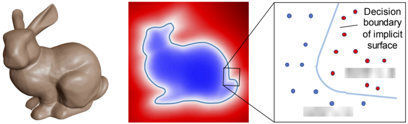




Neural Fields

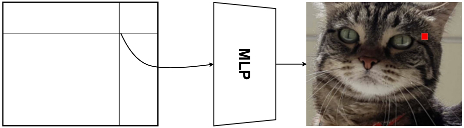
- **Implicit Representation:** F continuous function mapping spatial coordinates to colour / occupancy:
  - **Image:** $F : R^2 \mapsto R^3$: pixel to RGB.
  - **3D :** $F : R^2 \mapsto R^4$: RGB + occupancy.
- **Benefits:**
  - Infinite resolution
  - Adaptive to complexity
  - Can be learned
  - Generalization
- **Converting to other Format:**
  - **Image:** sample pixels

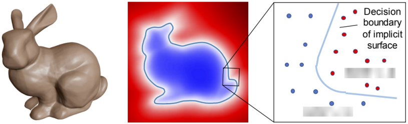




Neural Fields

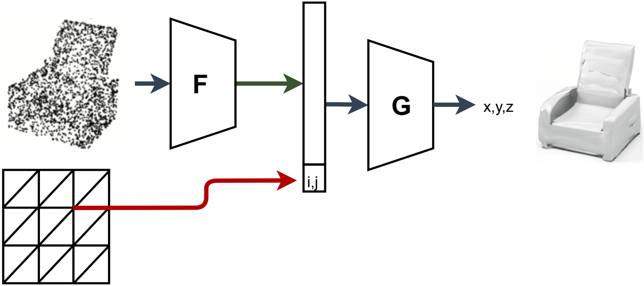
- **Implicit Representation:** F continuous function mapping spatial coordinates to colour / occupancy:
  - **Image:** $F : \mathrm{R}^2 \mapsto \mathrm{R}^3$: pixel to RGB.
  - **3D :** $F : \mathrm{R}^2 \mapsto \mathrm{R}^4$: RGB + occupancy.
- **Benefits:**
  - Infinite resolution
  - Adaptive to complexity
  - Can be learned
  - Generalization
- **Converting to other Format:**
  - **Image:** sample pixels
  - **3D:** Marching cubes.





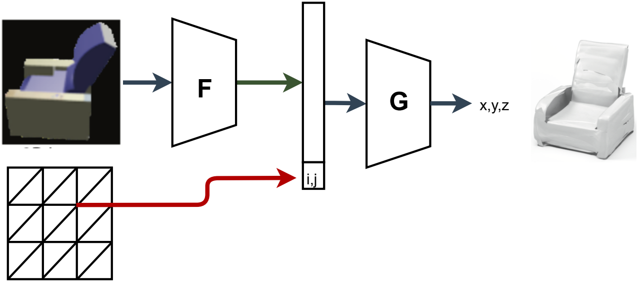
- **AtlasNet** $F : \mathrm{R}^{3\times N} \mapsto \mathrm{R}^m$ shape embedder, and $G : \mathrm{R}^{m+2} \mapsto \mathrm{R}^3$ function deforming a tessellated patch (papier-maché)

Groueix2018, **Saito2019**



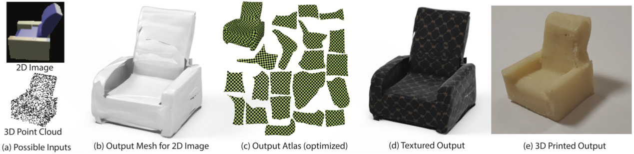
- **AtlasNet** $F : \mathbb{R}^{3\times N} \mapsto \mathbb{R}^m$ shape embedder, and $G : \mathbb{R}^{m+2} \mapsto \mathbb{R}^3$ function deforming a tessellated patch (papier-maché)
- Works with single images!

Groueix2018, **Saito2019**



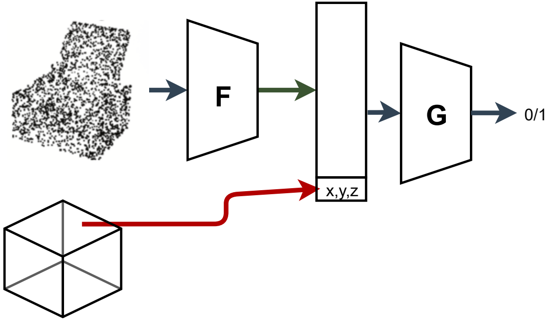
- **AtlasNet** $F : \mathrm{R}^{3\times N} \mapsto \mathrm{R}^m$ shape embedder, and $G : \mathrm{R}^{m+2} \mapsto \mathrm{R}^3$ function deforming a tessellated patch (papier-maché)
- Works with single images!

Groueix2018, **Saito2019**



- **AtlasNet** $F : \mathrm{R}^{3\times N} \mapsto \mathrm{R}^m$ shape embedder, and $G : \mathrm{R}^{m+2} \mapsto \mathrm{R}^3$ function deforming a tessellated patch (papier-maché)
- Works with single images!
- **PiFU** implicit functions : $G : \mathrm{R}^{m+3} \mapsto [0, 1]$: inside/outside

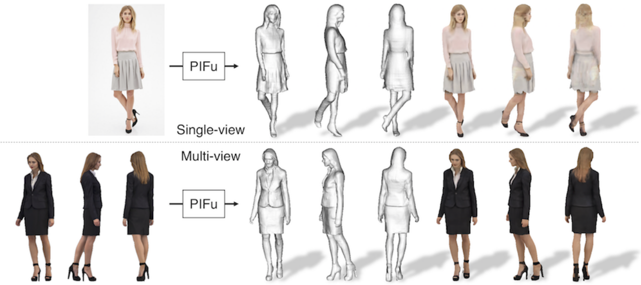
Groueix2018, **Saito2019**



Background on Deep 3D Reconstruction

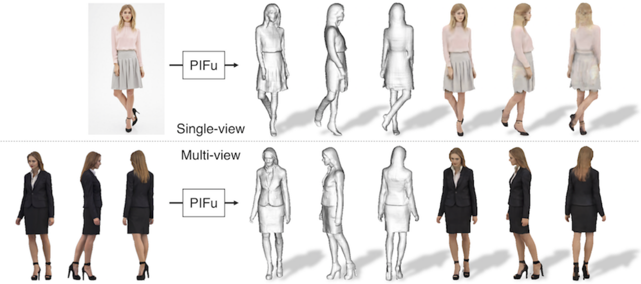
- **AtlasNet** $F : \mathbb{R}^{3\times N} \mapsto \mathbb{R}^m$ shape embedder, and $G : \mathbb{R}^{m+2} \mapsto \mathbb{R}^3$ function deforming a tessellated patch (papier-maché)
- Works with single images!
- **PiFU** implicit functions : $G : \mathbb{R}^{m+3} \mapsto [0,1]$: inside/outside
- Infinite resolution, no explicit mesh.

Groueix2018, **Saito2019**

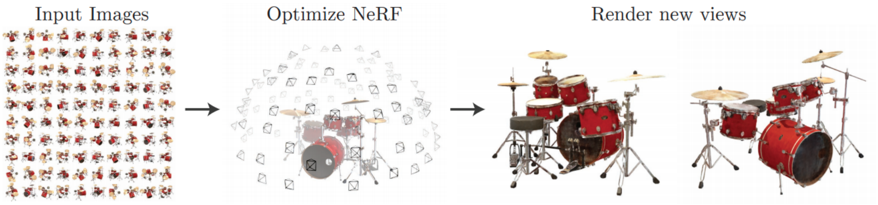


- **AtlasNet** $F : \mathrm{R}^{3\times N} \mapsto \mathrm{R}^m$ shape embedder, and $G : \mathrm{R}^{m+2} \mapsto \mathrm{R}^3$ function deforming a tessellated patch (papier-maché)
- Works with single images!
- **PiFU** implicit functions : $G : \mathrm{R}^{m+3} \mapsto [0, 1]$: inside/outside
- Infinite resolution, no explicit mesh.
- **Learns and generalize from a set of shapes.**

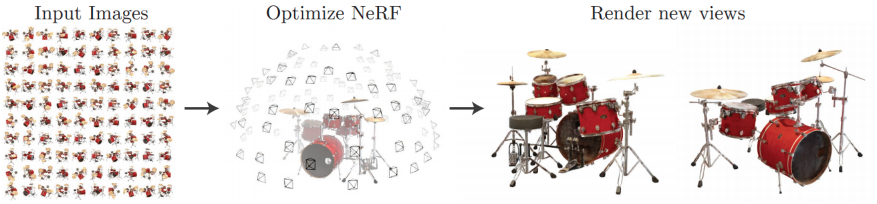
Groueix2018, **Saito2019**



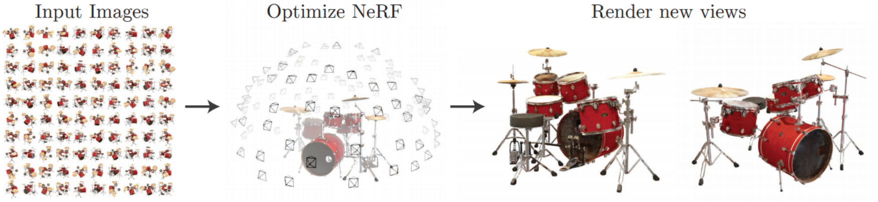
- **Problem:** new view synthesis



- **Problem:** new view synthesis
- We have (many) views of the same scene

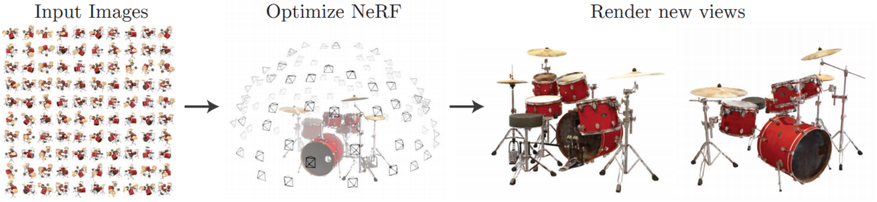


- **Problem:** new view synthesis
- We have (many) views of the same scene
- We train **a single neural network** to fit the scene



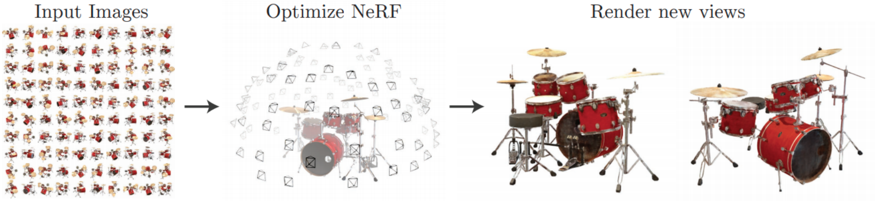
Neural Radiance Fields

- **Problem:** new view synthesis
- We have (many) views of the same scene
- We train **a single neural network** to fit the scene
- We can now generate new views from unseen camera positions



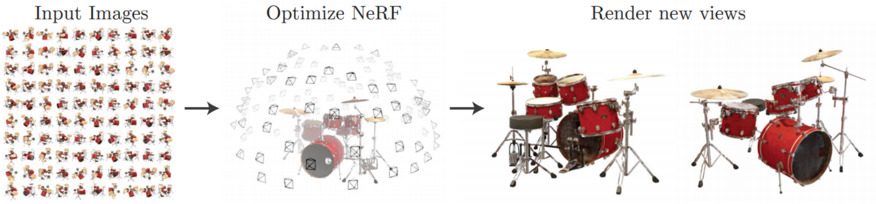
Neural Radiance Fields

- **Problem:** new view synthesis
- We have (many) views of the same scene
- We train **a single neural network** to fit the scene
- We can now generate new views from unseen camera positions
- Rely heavily on the recent **differentiable renderers**

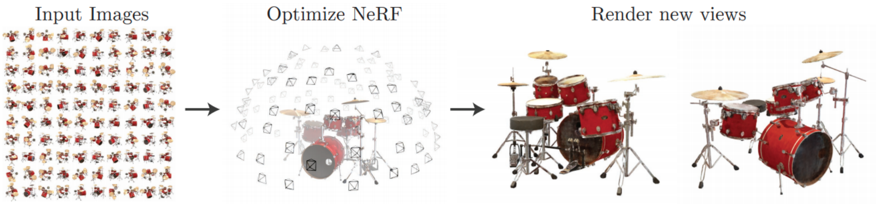


Neural Radiance Fields

- **Problem:** new view synthesis
- We have (many) views of the same scene
- We train **a single neural network** to fit the scene
- We can now generate new views from unseen camera positions
- Rely heavily on the recent **differentiable renderers**
- Does not generalize at all to unseen parts/scenes.



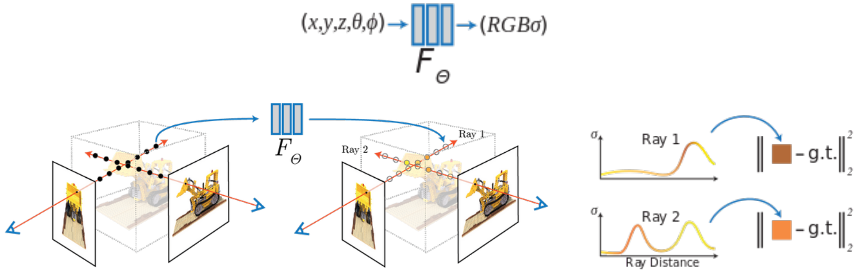
- **Problem:** new view synthesis
- We have (many) views of the same scene
- We train **a single neural network** to fit the scene
- We can now generate new views from unseen camera positions
- Rely heavily on the recent **differentiable renderers**
- Does not generalize at all to unseen parts/scenes.
- `https://www.matthewtancik.com/nerf`



Examples

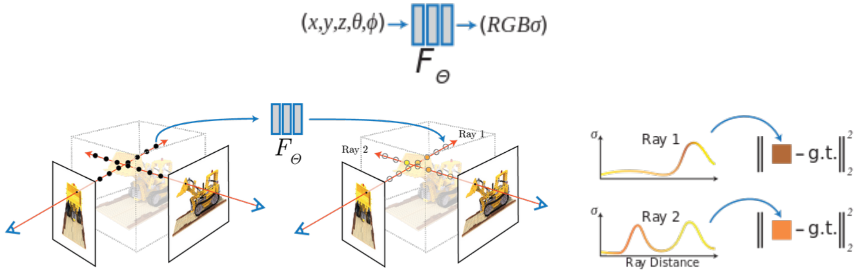
Exploit the (not yet fully understood) generalization of ANNs.

- We consider a ray associated with a pixel in an image



Differentiable Rendering

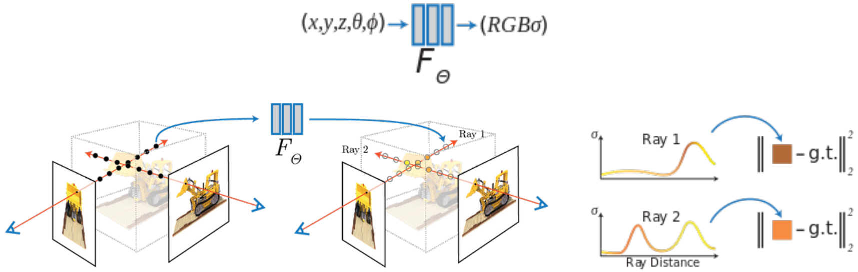
- We consider a ray associated with a pixel in an image
- We query a function $XYZ\theta\phi \mapsto RGB\sigma$ along the ray (σ : volume density)



Differentiable Rendering

- We consider a ray associated with a pixel in an image
- We query a function $XYZ\theta\phi \mapsto RGB\sigma$ along the ray (σ : volume density)
- We "integrate" the values along the ray.Rendering equation:

$$C(r) = \int_0^\infty \underbrace{\sigma(r_t)}_{\text{part of r stops.}} \quad \underbrace{c(r_t, d)}_{\text{color of pt in dir. d}} \quad T(t)dt \text{ with } \underbrace{T(t) = \exp\left(-\int_0^t \sigma(r_s)ds\right)}_{\text{what's left of r}}$$

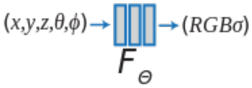


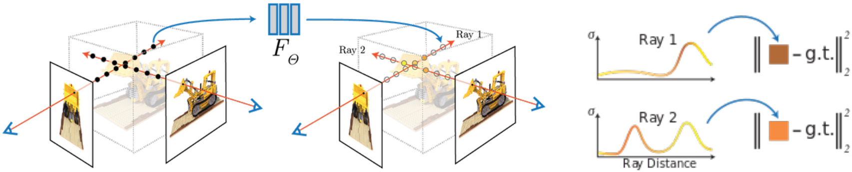
Differentiable Rendering

- We consider a ray associated with a pixel in an image
- We query a function $XYZ\theta\phi \mapsto RGB\sigma$ along the ray (σ : volume density)
- We "integrate" the values along the ray.Rendering equation:

$$C(r) = \int_0^\infty \underbrace{\sigma(r_t)}_{\text{part of r stops.}} \underbrace{c(r_t, d)}_{\text{color of pt in dir. d}} T(t)dt \text{ with } \underbrace{T(t) = \exp\left(-\int_0^t \sigma(r_s)ds\right)}_{\text{what's left of r}}$$

- We compare the resulting RGB with the true color
- For each image and pixel, many times.





Can generate photorealistic new view!

View-Dependant Appearance

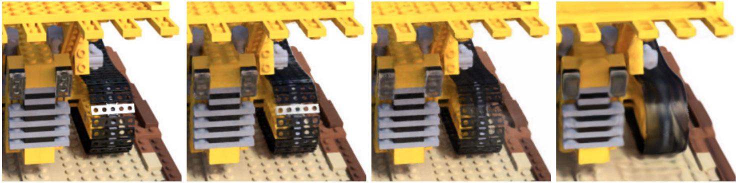
- Query at XYZ depends on the angle of view
- **Can naturally deal with non Lambertian + transparent materials!**

By-product: geometry. Can be meshified with Marching Cube.

Positional Encoding

- Very hard for ANN to learn high frequency band-pass filters
- XYZ represented with Positional Encodings / Discrete Fourrier Transforms:

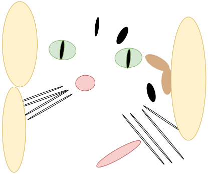
  each direction encoded into a D-sized vector of different frequencies
- $XYZ \mapsto \mathbb{R}^{3\times D}$
- Necessary to help the network to learn high frequencies / small details



Ground Truth | Complete Model | No View Dependence | No Positional Encoding

Gaussian Splatting

- Learns position and size of Gaussians
- Backpropagable through diff. renderer
- Blazing fast and super versatile!
- Very recent paper, more to come



Pros and Cons

PROS

- Looks really cool
- Does not need explicit 3D supervision
- A new way to encode 3D scenes
- The first visibility-driven implicit function method
- A new era for photogrammetry?

- Going further:

- `https://www.matthewtancik.com/nerf`

CONS

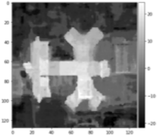
- Slow and compute-intensive to train
- Does not generalize to unseen views/ objects
- Implicit representation - hard to understand and manipulate
- Can it scale?
- Already obsolete?

Usage in Earth Observation

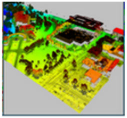
Multi-view set of images

Render new views

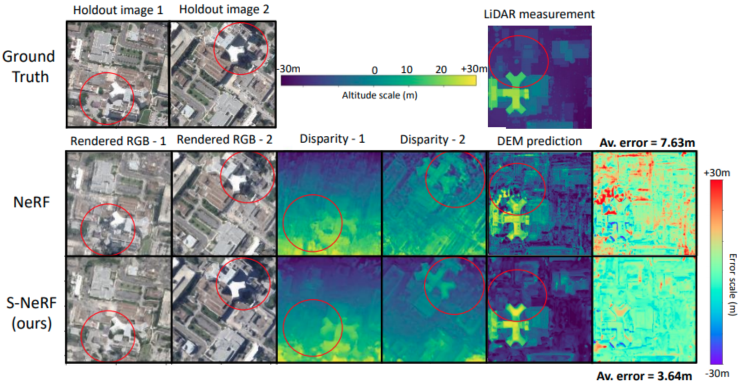
Extract depth (topography)



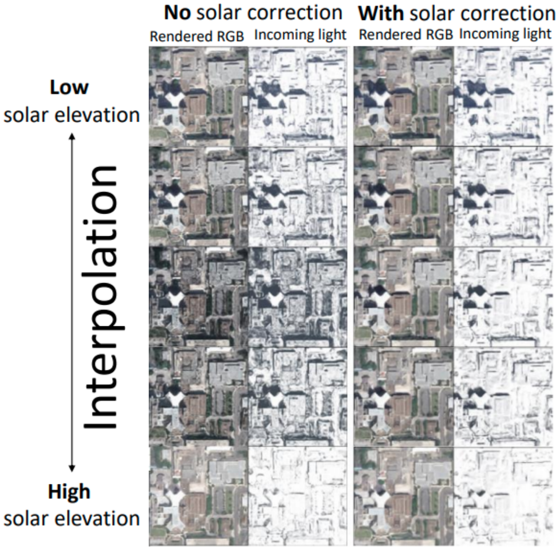
Generate polygonal
3d surface model



Shadow Neural Radiance Fields for Multi-view Satellite Photogrammetry, Derksen & Izzo, EarthVision21



Shadow Neural Radiance Fields for Multi-view Satellite Photogrammetry, Derksen & Izzo, EarthVision21



Shadow Neural Radiance Fields for Multi-view Satellite Photogrammetry, Derksen & Izzo, EarthVision21