Neural Radiance Fields

Jon Barron

Google Research
About me

UC Berkeley
PhD Student
2008-2013
Advisor: Jitendra Malik

Google Research: Perception
Research Scientist
2013-Now

Team:
Peter Hedman
Pratul Srinivasan
Ben Mildenhall
Research Interests

Inverse Rendering

Color Constancy

Smooth Motion / Depth Estimation

Loss Functions
Research Impact

- HDR+ / Night Sight
- Lens Blur / Portrait Mode
- Portrait Light

- Google Glass
- Jump
What is graphics?
What is graphics?

Mesh Rendering
Is this “neural rendering”?

Paradigm 1:
“The neural network is a black box that directly renders pixels”

Paradigm A:
“The neural network is a black box that models the geometry of the world, and a (non-learned) graphics engine renders it”

Scene Representation
Implicit Representations
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Ben Mildenhall*
Pratul Srinivasan*
Matt Tancik*
Jon Barron
Ravi Ramamoorthi
Ren Ng

UC Berkeley
UC Berkeley
UC Berkeley
Google Research
UC San Diego
UC Berkeley

Google
Google
Google

*Equal contribution
Problem: View Interpolation

Inputs: sparsely sampled images of scene

Outputs: new views of same scene

tancik.com/nerf
RGB-alpha volume rendering for view synthesis

**Soft 3D**  
(Penner & Zhang 2017)  
Culmination of non-deep stereo matching techniques

**Multiplane image methods**
- Stereo Magnification (Zhou et al. 2018)
- Pushing the Boundaries… (Srinivasan et al. 2019)
- Local Light Field Fusion (Mildenhall et al. 2019)
- DeepView (Flynn et al. 2019)
- Single-View… (Tucker & Snavely 2020)

Typical deep learning pipelines - images go into a 3D CNN, big RGBA 3D volume comes out

**Neural Volumes**  
(Lombardi et al. 2019)  
Direct gradient descent to optimize an RGBA volume, regularized by a 3D CNN
RGB-alpha volume rendering for view synthesis

**Soft 3D**
(Penner & Zhang 2017)
Culmination of non-deep stereo matching techniques

**Multiplane image methods**
- Stereo Magnification (Zhou et al. 2018)
- Pushing the Boundaries… (Srinivasan et al. 2019)
- Local Light Field Fusion (Mildenhall et al. 2019)
- DeepView (Flynn et al. 2019)
- Single-View… (Tucker & Snavely 2020)

Typical deep learning pipelines - images go into a 3D CNN, big RGBA 3D volume comes out

**Neural Volumes**
(Lombardi et al. 2019)
Direct gradient descent to optimize an RGBA volume, regularized by a 3D CNN
RGB-alpha volume rendering for view synthesis

Soft 3D
(Penner & Zhang 2017)
Culmination of non-deep stereo matching techniques

Multiplane image methods
Stereo Magnification (Zhou et al. 2018)
Pushing the Boundaries… (Srinivasan et al. 2019)
Local Light Field Fusion (Mildenhall et al. 2019)
DeepView (Flynn et al. 2019)
Single-View… (Tucker & Snavely 2020)

Typical deep learning pipelines - images go into a 3D CNN, big RGBA 3D volume comes out

Neural Volumes
(Lombardi et al. 2019)
Direct gradient descent to optimize an RGBA volume, regularized by a 3D CNN
RGB-alpha volume rendering for view synthesis

**Soft 3D** (Penner & Zhang 2017)
Culmination of non-deep stereo matching techniques

**Neural Volumes** (Lombardi et al. 2019)
Direct gradient descent to optimize an RGBA volume, regularized by a 3D CNN

**Multiplane image methods**
- Stereo Magnification (Zhou et al. 2018)
- Pushing the Boundaries... (Srinivasan et al. 2019)
- Local Light Field Fusion (Mildenhall et al. 2019)
- DeepView (Flynn et al. 2019)
- Single-View... (Tucker & Snavely 2020)

**Typical deep learning pipelines** - images go into a 3D CNN, big RGBA 3D volume comes out

+ Great rendering model: good for optimization
- Horrible storage requirements (1-10 GB)
Neural networks as a continuous shape representation

\[ f_\theta(p) = \tau \]

Occupancy Networks, Mescheder et al. CVPR 2019
Neural networks as a continuous shape representation

**Occupancy Networks**  
(Mescheder et al. 2019)  
\((x, y, z) \rightarrow\) occupancy

\[ f_\theta(p) = \tau \]

**DeepSDF**  
(Park et al. 2019)  
\((x, y, z) \rightarrow\) distance

**Scene Representation Networks**  
(Sitzmann et al. 2019)  
\((x, y, z) \rightarrow\) latent vec. (color, dist.)

**Differentiable Volumetric Rendering**  
(Niemeyer et al. 2020)  
\((x, y, z) \rightarrow\) color, occ.
Neural networks as a continuous shape representation

Occupancy Networks
(Mescheder et al. 2019)
\((x, y, z) \rightarrow \text{occupancy}\)

DeepSDF
(Park et al. 2019)
\((x, y, z) \rightarrow \text{distance}\)

- Limited rendering model: difficult to optimize
+ Highly compressible (1-10 MB)

Scene Representation Networks
(Sitzmann et al. 2019)
\((x, y, z) \rightarrow \text{latent vec. (color, dist.)}\)

Differentiable Volumetric Rendering
(Niemeyer et al. 2020)
\((x, y, z) \rightarrow \text{color, occ.}\)
NeRF (neural radiance fields)

\[(x, y, z, \theta, \phi) \rightarrow F_\theta \rightarrow (r, g, b, \sigma)\]

- Spatial location
- Viewing direction
- Fully-connected neural network
  - 9 layers,
  - 256 channels
- Output color
- Output density
Generate views with traditional volume rendering
Volume rendering is trivially differentiable

Rendering model for ray \( r(t) = o + td \):

\[
C \approx \sum_{i=1}^{N} T_i \alpha_i c_i
\]

\( T_i \) \( \alpha_i \) \( c_i \)

weights colors
Volume rendering is trivially differentiable

Rendering model for ray \( r(t) = o + td \):

\[
C \approx \sum_{i=1}^{N} T_i \alpha_i c_i
\]

How much light is blocked earlier along ray:

\[
T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)
\]
Volume rendering is trivially differentiable

Rendering model for ray \( r(t) = o + td \):

\[
C \approx \sum_{i=1}^{N} T_i \alpha_i c_i
\]

How much light is blocked earlier along ray:

\[
T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)
\]

How much light is contributed by ray segment \( i \):

\[
\alpha_i = 1 - e^{-\sigma_i \delta t_i}
\]
Optimize with gradient descent on rendering loss

\[
\min_{\theta} \sum_{i} \left\| \text{render}_{i}(F_{\theta}) - I_{i} \right\|^2
\]
Training network to reproduce all input views of the scene
Viewing directions as input

(a) View 1

(b) View 2

(c) Radiance Distributions
Results
vs. Prior Work (Implicit / MLP)

SRN [Sitzmann et al. 2019] vs. NeRF

Nearest Input
vs. Prior Work (Implicit / MLP)

SRN [Sitzmann et al. 2019]

NeRF

Nearest Input
View-Dependent Effects
Detailed Geometry & Occlusion
Detailed Geometry & Occlusion
Meshable
Baking Neural Radiance Fields for Real-Time View Synthesis

arXiv 2021

Peter Hedman  Pratul P. Srinivasan  Ben Mildenhall  Jonathan T. Barron  Paul Debevec

Google Research

Paper  Video  Demos

http://nerf.live/
Naive implementation produces blurry results

NeRF (Naive)
Naive implementation produces blurry results

Naive implementation: blurry

NeRF (Naive)

NeRF (with positional encoding)
Toy problem: memorizing a 2D image

$$(x, y) \rightarrow \text{ memory block } \rightarrow (r, g, b)$$
Toy problem: memorizing a 2D image

- Ground truth image
- Standard fully-connected net
\[
\begin{pmatrix}
\sin(v), \cos(v) \\
\sin(2v), \cos(2v) \\
\sin(4v), \cos(4v) \\
\vdots \\
\sin(2^{L-1}v), \cos(2^{L-1}v)
\end{pmatrix}
\]
Ground truth image

Standard fully-connected net

With Positional Encoding
Positional encoding also directly improves our scene representation!
Why?
Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

Matthew Tancik*, Pratul Srinivasan*, Ben Mildenhall*, Sara Fridovich-Keil, Nithin Ragahavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, Ren Ng
Positional Encoding [1]: \( \gamma(v) = [\cos(2^0v), \sin(2^0v), \ldots, \cos(2^{L-1}v), \sin(2^{L-1}v)] \)

Random Fourier Features [2]: \( \gamma(v) = [\cos(Bv), \sin(Bv)] \quad B \sim \mathcal{N}(0, \sigma^2) \)

Neural Tangent Kernel

\[ f(x; \theta) \approx \sum_i (K^{-1}y)_i k(x_i, x) \]

Under certain conditions, neural networks are kernel regression(!)

\[ k(x_i, x_j) = h_{NTK}(\langle x_i, x_j \rangle) \]

ReLU MLPs correspond to a “dot product” kernel

\[ h_{NTK} : \mathbb{R} \to \mathbb{R} \]

Jacot et al., NeurIPS, 2018, Arora, et al., ICML, 2019, Basri et al., 2020., Du et al., ICLR, 2019., Lee et al., NeurIPS, 2019
Dot Product of Fourier Features

\[
\langle \gamma(v_1), \gamma(v_2) \rangle = \sum_j \left( \cos(b_j^T v_1) \cos(b_j^T v_2) + \sin(b_j^T v_1) \sin(b_j^T v_2) \right)
\]

\[
= \sum_j \cos \left( b_j^T (v_1 - v_2) \right) \quad \text{(cosine difference trig identity)}
\]

\[
\triangleq h_\gamma(v_1 - v_2)
\]

Fourier Features → \textit{stationary} kernel
Resulting composed NTK is stationary

\[ h_{\text{NTK}} \left( \langle \gamma(v)_i, \gamma(v)_j \rangle \right) = h_{\text{NTK}}(h_\gamma(v_i - v_j)) \]

Resulting network regression function is a convolution

\[ \hat{f} = (h_{\text{NTK}} \circ h_\gamma) \ast \sum_{i=1}^{n} w_i \delta v_i \]
Mapping bandwidth controls underfitting / overfitting

\[ \gamma(v) = [\cos(Bv), \sin(Bv)] \quad B \sim \mathcal{N}(0, \sigma^2) \]
Mapping bandwidth controls underfitting / overfitting

$$\gamma(v) = [\cos(Bv), \sin(Bv)] \quad B \sim \mathcal{N}(0, \sigma^2)$$
Mapping bandwidth controls underfitting / overfitting

\[
\gamma(v) = [\cos(Bv), \sin(Bv)] \quad B \sim \mathcal{N}(0, \sigma^2)
\]
With Fourier features
\[ \gamma(v) = FF(v) \]

No Fourier features
\[ \gamma(v) = v \]

(b) Image regression
\( (x,y) \rightarrow \text{RGB} \)

(c) 3D shape regression
\( (x,y,z) \rightarrow \text{occupancy} \)

(d) MRI reconstruction
\( (x,y,z) \rightarrow \text{density} \)

(e) Inverse rendering
\( (x,y,z) \rightarrow \text{RGB, density} \)
B = SCALE * np.random.normal(shape=(input_dims, NUM_FEATURES))
x = np.concatenate([np.sin(x @ B), np.cos(x @ B)], axis=-1)
x = nn.Dense(x, features=256)
Coordinate based neural representation
≠
a magic black box that learns things and generalizes
Coordinate based neural representation
≠
a magic black box that learns things and generalizes

Coordinate based neural representation
=
a tiny n-dimensional lookup table with extremely high resolution
Learned Initializations for Optimizing Coordinate-Based Neural Representations

Matthew Tancik*¹  Ben Mildenhall*¹  Terrance Wang¹  Divi Schmidt¹
Pratul P. Srinivasan²  Jonathan T. Barron²  Ren Ng¹

Target

Init.  Step 1  Step 2

Standard Initialization

Meta-learned Initialization (MAML)
NeRF in the Wild: Neural Radiance Fields for Uncontrolled Photo Collections

CVPR 2021

Ricardo Martin-Brualla*, Noha Radwan*, Mehdi Sajjadi*, Jonathan T. Barron, Alexey Dosovitskiy, Daniel Duckworth

Google Brain Berlin & Google Research

https://nerf-w.github.io/
Unconstrained photo collection

Novel views + Novel appearance
Transient Viewpoint Reconstruction

Inputs

- Viewpoint
- Appearance Embedding
- Transient Embedding

Static

Transient

Uncertainty

Reconstruction

Target

Photos by Flickr users: vasnic64 / CC BY
Viewpoint

Appearance Embedding
“NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, Mildenhall, Srinivasan, Tancik et. al., ECCV 2020
Trevi flythrough
Neural Rendering in the Wild

Ours

Neural Rendering in the Wild

“Neural Rerendering In the Wild”, Meshry et. al., CVPR 2019
Thanks!

http://jonbarron.info

https://twitter.com/jon_barron