Modeling the plenoptic function
Outline

• The plenoptic function
• Two-plane light fields
• Plenoptic camera
• Neural radiance fields (NeRFs)
Goal: Novel view rendering

- Given several images of the same object or scene from known viewpoints, how can we generate a rendering of the same scene from a novel viewpoint?
- Multiview stereo answer: create a textured 3D model from the images, use traditional graphics to render
Goal: Novel view rendering

- Given several images of the same object or scene from known viewpoints, how can we generate a rendering of the same scene from a novel viewpoint?
- Multiview stereo answer: create a textured 3D model from the images, use traditional graphics to render
- Alternate answer: model the light field of the scene, sample new views from it
The light field, or plenoptic function

Q: What is the set of all things that we can ever see?
A: The *plenoptic function*

Q: What is the set of all things that we can ever see?
A: The *plenoptic function*

Let’s start with a stationary person and try to parameterize everything that they can see…
Grayscale snapshot

$L(\theta, \phi)$

- Intensity of light
  - Seen from a single viewpoint
  - At a single time
  - Averaged over the wavelengths of the visible spectrum
Color snapshot

$L(\theta, \phi, \lambda)$

- Intensity of light
  - Seen from a single viewpoint
  - At a single time
  - As a function of wavelength
Modeling the light field

3D world

2D image

Point of observation
Modeling the light field

3D world

2D image

Painted backdrop
A movie

\[ L(\theta, \phi, \lambda, t) \]

- Intensity of light
  - Seen from a single view point
  - Over time
  - As a function of wavelength
Holographic movie

$L(\theta, \phi, \lambda, t, x, y, z)$

- Intensity of light
  - Seen from ANY viewpoint
  - Over time
  - As a function of wavelength
End-of-semester deadlines

• **Quiz 4** will be out **9AM Thursday, December 1st** through **9AM Monday, December 5th**
• **Assignment 5** is out, due **Tuesday, December 6**
• **Extra credit project presentations** will take place on **Monday, December 5th** and **Wednesday, December 7th**
  *please sign up by next Tuesday!*
• Final project reports will be due on **Monday, December 12th**
Light field modeling: Outline

• The plenoptic function
• Two-plane light fields
• Plenoptic camera
• Neural radiance fields (NeRFs)
The plenoptic function

$L(\theta, \phi, \lambda, t, x, y, z)$

• Can reconstruct every possible view, at every moment, from every position, at every wavelength
• Contains every photograph, every movie, everything that anyone has ever seen! it completely captures our visual reality!
• Not bad for a function…
The plenoptic function: More practical version

$L(\theta, \phi, x, y, z) = (r, g, b)$

• Other simplifications/variants are possible, as we will see
Modeling the plenoptic function

- **Capture**
  - Create a special camera setup to capture a slice of the plenoptic function
  - Combine captured rays for novel view synthesis, defocus, and other effects

- **Optimization**
  - Given a set of multi-view calibrated images, optimize a parametric representation of the plenoptic function of the scene
Outline

• The plenoptic function
• Two-plane light fields
Two-plane light fields

- Key idea: assuming light is constant along rays, we can create a 4D parameterization of the light field


Two-plane light fields

- Two-plane parameterization:
Two-plane light fields

- Two-plane parameterization:

\[ L(s,t,u,v) = (r,g,b) \]
Two-plane light fields

- Two-plane parameterization:

\[ L(s, t, u, v) = (r, g, b) \]
Two-plane light fields

- What do we get if we hold \( u, v \) constant and let \( s, t \) vary?
- An image!
Two-plane light fields

• What do we get if we hold $u, v$ constant and let $s, t$ vary?
• An image!
Two-plane light fields

- What do we get if we hold $s, t$ constant and let $u, v$ vary?
- A set of rays leaving a point in the scene in a bundle of directions towards the image plane
Light field visualization

Figure source: M. Levoy and P. Hanrahan
Light field capture

- Idea 1: move camera carefully over $u, v$ plane
Stanford multi-camera array

- 640 × 480 pixels × 30 fps × 128 cameras
- Synchronized timing
- Continuous streaming
- Flexible arrangement

http://graphics.stanford.edu/projects/array/
Light field capture

- Idea 2: move camera anywhere, use rebinning or resampling
Light field capture

- Idea 2: move camera anywhere, use rebinning or resampling

Figure 10: The capture stage

Figure source: S. Gortler et al.
Novel view synthesis

- For each output pixel, determine $s$, $t$, $u$, $v$, then either use closest discrete RGB or interpolate several nearby values
Outline

• The plenoptic function
• Two-plane light fields
• Plenoptic camera
Plenoptic camera

R. Ng et al. *Light Field Photography with a Hand-held Plenoptic Camera*. 2005
Conventional vs. light field camera
Conventional vs. light field camera
Prototype camera

Contax medium format camera

Kodak 16-megapixel sensor

Adaptive Optics microlens array

125µ square-sided microlenses

4000 × 4000 pixels / 292 × 292 lenses = 14 × 14 pixels per lens
Captured light field
Captured light field
Captured light field
Digitally stopping down (reducing the aperture)

- stopping down = summing only the central portion of each microlens
Digital refocusing

- refocusing = summing windows extracted from several microlenses
Digital refocusing
Digitally moving the observer

- moving the observer = moving the window we extract from the microlenses
Digitally moving the observer
Digitally moving the observer
Lytro (RIP)

https://en.wikipedia.org/wiki/Lytro
What happened to Lytro?
Outline

• The plenoptic function
• Two-plane light fields
• Plenoptic camera
• Neural radiance fields (NeRFs)
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
ECCV 2020 (best paper honorable mention)

https://www.matthewtancik.com/nerf
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
ECCV 2020 (best paper honorable mention)
Train a neural network to represent the plenoptic function

Inputs: sparsely sampled images of scene

Outputs: new views of same scene

tancik.com/nerf

Slide credit: Jon Barron
Neural radiance field

$$(x, y, z, \theta, \phi) \rightarrow F_\Omega \rightarrow (r, g, b, \sigma)$$

Spatial location  Viewing direction  Output color  Output density

MLP
9 layers,
256 channels

Volumetric “fog” model: Every 5D input gets mapped to **Color** and **Density**
NeRF rendering

• At every point you know color and density: \((c_i, \sigma_i)\)
• Need to integrate these values to render a pixel
• Idea: sum how much light reaches each point * visibility * color
How to render a pixel: Volume rendering

Given: a ray  \( \mathbf{r}(i) = \bar{o} + i\mathbf{d} \)  

At every point you know: \( (c_i, \sigma_i) \)

\[
C(\mathbf{r}) \approx \sum_{i}^{N} w_i c_i \quad w_i = T_i \alpha_i
\]

Alpha: How much light a ray segment contributes

\[
\alpha_i = 1 - \exp(-\sigma_i \delta_i)
\]

Transmittance: how much light reaches point \( i \)

\[
T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)
\]
Training: Optimization with reconstruction loss

\[
\min_{\Omega} \sum_i \|\text{render}^{(i)}(F_{\Omega}) - I^{(i)}_{\text{gt}}\|^2
\]
Example results
Viewpoint-dependent effects
Viewpoint-dependent effects
Rendering expected depth

\[ d(r) \approx \sum_{i=1}^{N} T_i \alpha_i z_i \]
Because it models the entire plenoptic function you can insert objects with proper occlusion effects (in contrast to lightfields)
Extract surface on high density regions
NeRF limitations

- Expensive / slow to train and render
- Sensitive to sampling strategy
- Does not generalize between scenes
- Sensitive to pose accuracy
- Assumes static scene
- Assumes static lighting and camera focus
- Not a mesh
NeRF explosion

Awesome Neural Radiance Fields

A curated list of awesome neural radiance fields papers, inspired by awesome-computer-vision.

How to Pull Request?
If you are interested in adding papers, feel free to submit a pull request following the instruction here.

Table of Contents
- Survey
- Papers
- Talks

Survey

- Neural Volume Rendering: NeRF And Beyond, Dellaert and Yen-Chen, Arxiv 2020 | blog | github

Papers

- NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, Mildenhall et al., ECCV 2020 | github | bibtex
- NeRF++: Analyzing and Improving Neural Radiance Fields, Zhang et al., Arxiv 2020 | github | bibtex
- DeRF: Decomposed Radiance Fields, Rebain et al. Arxiv 2020 | bibtex
- NeRD: Neural Reflectance Decomposition from Image Collections, Boss et al., Arxiv 2020 | github | bibtex
- NeRF::: Neural Radiance Fields Without Known Camera Parameters, Wang et al., Arxiv 2021 | github | bibtex

Faster Inference

- Neural Sparse Voxel Fields, Liu et al., NeurIPS 2020 | github | bibtex
- AutoInt: Automatic Integration for Fast Neural Volume Rendering, Lindell et al., Arxiv 2020 | bibtex

Unconstrained Images

- NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections, Martin-Brual et al., Arxiv 2020 | bibtex

Deformable

- Deformable Neural Radiance Fields, Park et al., Arxiv 2020 | github | bibtex
- D-NeRF: Neural Radiance Fields for Dynamic Scenes, Pumarola et al., Arxiv 2020 | bibtex
- Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction, Gafni et al., Arxiv 2020 | bibtex
- Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Deforming Scene from Monocular Video, Tresch et al., Arxiv 2020 | github | bibtex

Video

- Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes, Li et al., Arxiv 2020 | bibtex
- Space-time Neural Irradiance Fields for Free-Viewpoint Video, Xian et al., Arxiv 2020 | bibtex
- Neural Radiance Flow for 4D View Synthesis and Video Processing, Du et al., Arxiv 2020 | bibtex

Generalization

- GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis, Schwarz et al., NeurIPS 2020 | github | bibtex
- GIF: Learning a General Radiance Field for 3D Scene Representation and Rendering, Trudel et al.,

https://github.com/yenchenlin/awesome-NeRF
NeRF generalizations

- PixelNeRF [Yu et al. CVPR’21]
- IBRNet [Wang et al. CVPR’21]
- MVSNeRF [Yao et al. ICCV’21]
- NerfingMVS [Wei et al. ICCV’21]
- NeRFormer [Reizenstein et al. ICCV’21]
- GRF [Trevithick et al. ICCV’21]
- …
NeRF in the wild

R. Martin-Brualla et al. NeRF in the Wild. CVPR 2021
PixelNeRF

Three input views

PixelNeRF

3-view NeRF

A. Yu et al. *PixelNeRF: Neural Radiance Fields from One or Few Images*. CVPR 2021
A. Yu et al. **PixelNeRF: Neural Radiance Fields from One or Few Images**. CVPR 2021
PixelNeRF

A. Yu et al. *PixelNeRF: Neural Radiance Fields from One or Few Images*. CVPR 2021
Fast Inference

- PlenOctrees [Yu et al. ICCV’21]
- SNeRG [Hedman et al. ICCV’21]
- FastNeRF [Garbin et al. ICCV’21]
- KiloNeRF [Reiser et al. ICCV’21]
- AutoInt [Lindell et al. CVPR’21]
- ...
Plenoxels

S. Fridovich-Kiel et al. Plenoxels: Radiance Fields without Neural Networks. CVPR 2022
Figure 2. **Overview of our sparse Plenoxel model.** Given a set of images of an object or scene, we reconstruct a (a) sparse voxel (“Plenoxel”) grid with density and spherical harmonic coefficients at each voxel. To render a ray, we (b) compute the color and opacity of each sample point via trilinear interpolation of the neighboring voxel coefficients. We integrate the color and opacity of these samples using (c) differentiable volume rendering, following the recent success of NeRF [26]. The voxel coefficients can then be (d) optimized using the standard MSE reconstruction loss relative to the training images, along with a total variation regularizer.

Pose Estimation

- GNeRF [Meng et al. ICCV ‘21]
- BARF [Lin et al. ICCV ‘21]
- NeRF– [Wang et al. arXiv ‘21]
- SC-NeRF [Jeong et al. ICCV ’21]
- iNeRF [Yen-Chen et al. IROS ‘21]
Robotics / Simulation

NeRF-GTO: Using a Neural Radiance Field to Grasp Transparent Objects [Ichnowski et al. CoRL ‘21]

3D Neural Scene Representations for Visuomotor Control [Li et al. Corl ‘21]

Others: iMAP[Sucar ICCV’21]
Object Decomposition

ST-NeRF [Zhang et al.]
SISGRAPH ‘21

Yang et al.

Neural Scene Graphs [Ost et al. CVPR ‘21]

Others: OSF [Guo et al.], uORF [Yu et al.]
Neural RGB-D surface reconstruction

Figure 1. Our method obtains a high-quality 3D reconstruction from an RGB-D input sequence by training a multi-layer perceptron. The core idea is to reformulate the neural radiance field definition in NeRF [48], and replace it with a differentiable rendering formulation based on signed distance fields which is specifically tailored to geometry reconstruction.

D. Azinovic et al. Neural RGB-D Surface Reconstruction. CVPR 2022
DreamFusion

B. Poole, A. Jain, J. Barron, B. Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. arXiv 2022
DreamFusion