Deep Learning for Computer Vision

Fall 2022

https://cool.ntu.edu.tw/courses/189345 (NTU COOL)
http://vllab.ee.ntu.edu.tw/dlcv.html (Public website)

Yu-Chiang Frank Wang 王鈺強, Professor

Dept. Electrical Engineering, National Taiwan University

2022/11/22

What to Cover Today?

- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
    - Extensions of NeRF
    - Advanced Topics of NeRF

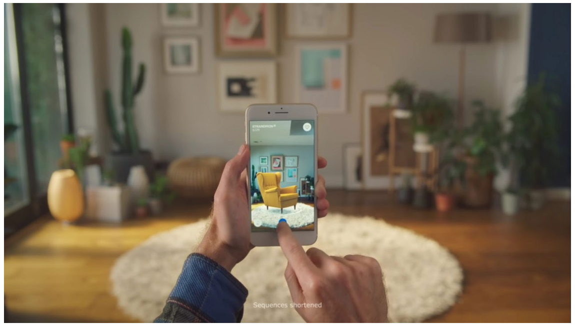
What is 3D Vision?

- Enable machine to perceive and reconstruct the 3D world which we live in.

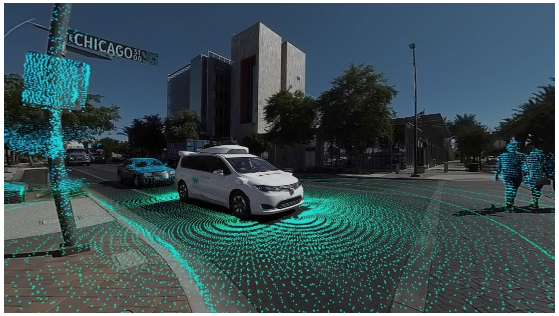
Applications of 3D Vision

- Robotics

- Augmented Reality



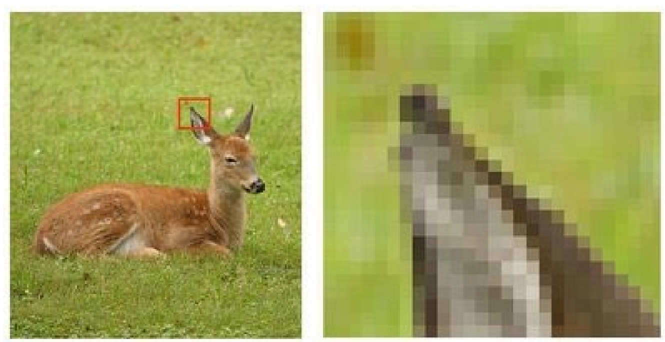
- Autonomous driving



References:
Boston Dynamics: https://www.youtube.com/watch?v=fn3KWM1kuAw
Ikea: https://www.youtube.com/watch?v=UudV1VdFtuQ
Waymo: https://www.youtube.com/watch?v=B8R148hFxPw

How to Represent the 3D World?

- Recap: 2D representations
  - RGB pixels
  - Images/videos
  - Why 2D vision not good enough?
    - Lack of depth, scene geometry, etc. information



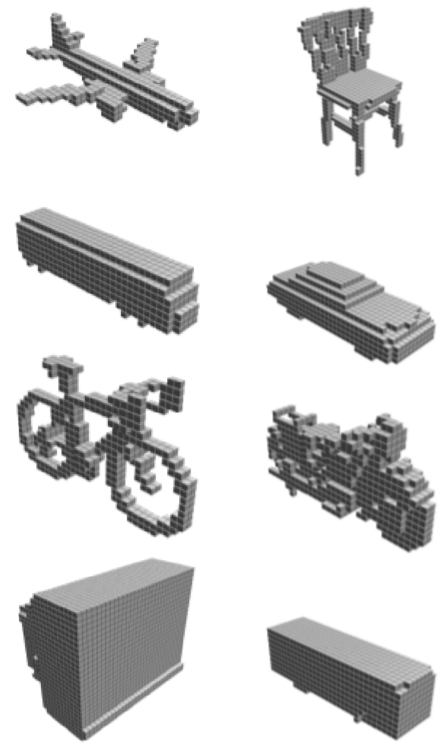
- What about 3D representations?

How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images

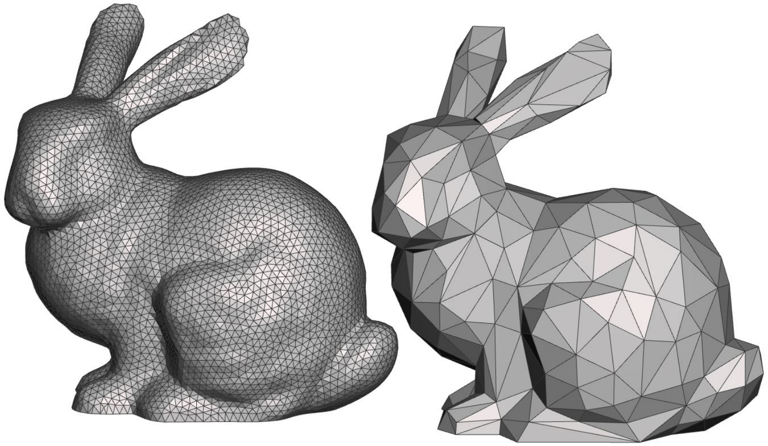
How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images
- Voxels



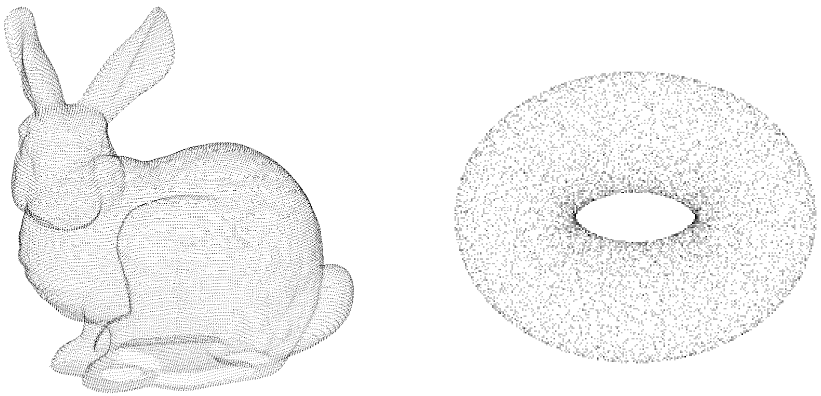
How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images
- Voxels
- Polygon Mesh



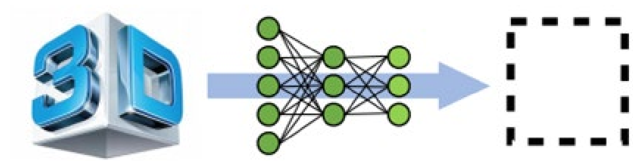
How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images
- Voxels
- Polygon Mesh
- Point Cloud
- …

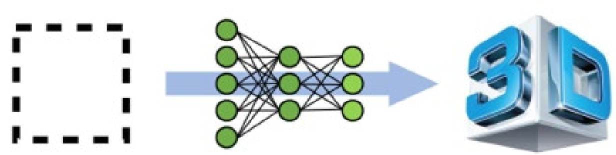


Deep Learning for 3D Vision

- Perception: extract information from 3D shapes (Part 1)



- Reconstruction: synthesis 3D shapes (Part 2)

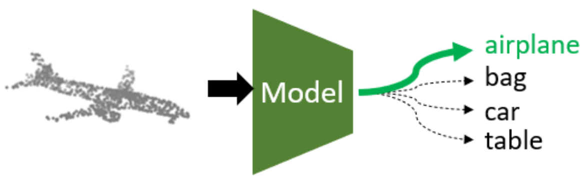


What to Cover Today?

- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
  - Extension of NeRF
  - Advanced Topics of NeRF

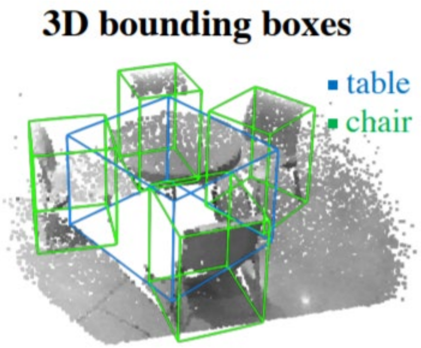
3D Perception

- Extract information from 3D shapes for downstream tasks
  - Classification
  - Object/scene segmentation
  - Pose estimation
  - Object detection



Classification

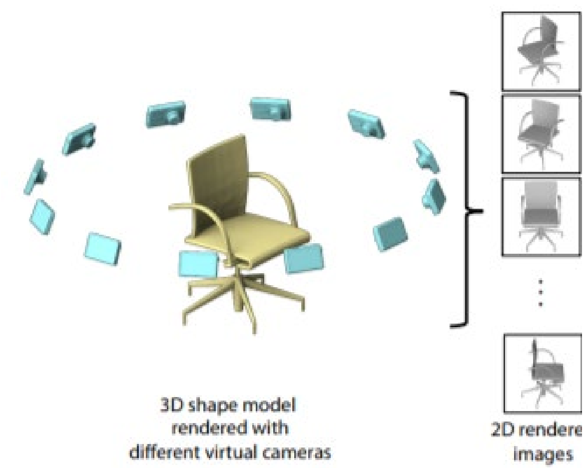
Segmentation

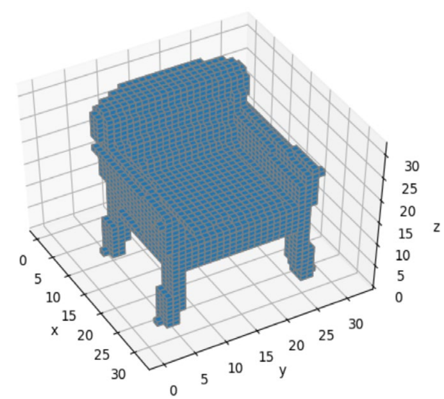


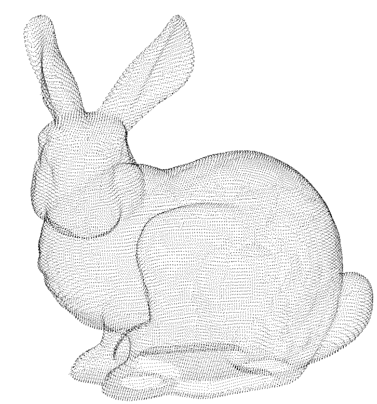
Detection

3D Perception

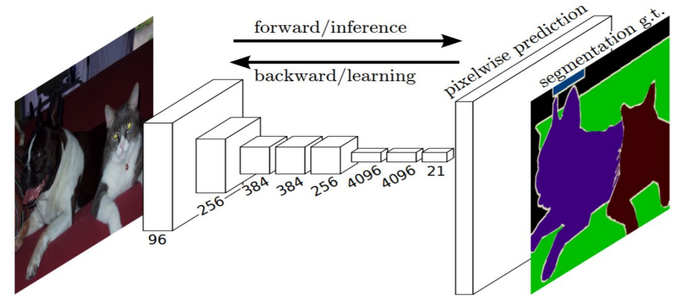
- In this part, we will talk about feature extraction from:
  - Multi-view images
  - Voxel
  - Point cloud





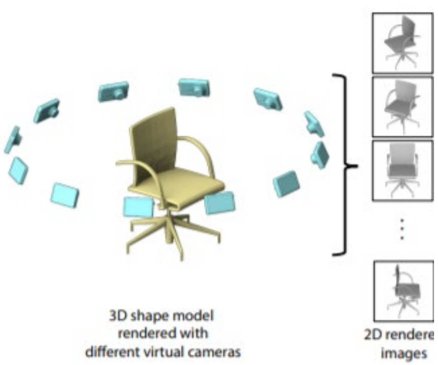


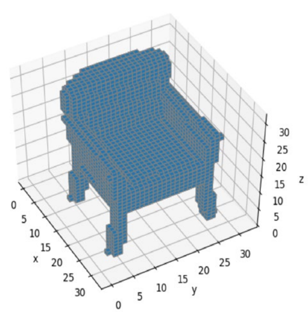
Limitation of CNN

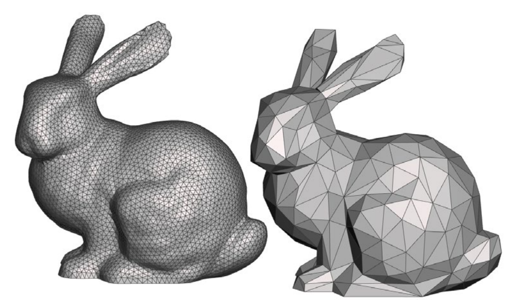


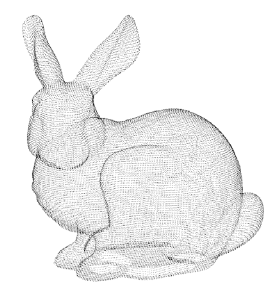
- Can we directly apply CNN on 3D data?
  - Well, it depends…

3D Representation	CNN applicable?
Multi-view images	
Voxel	
Mesh	
Point Cloud	



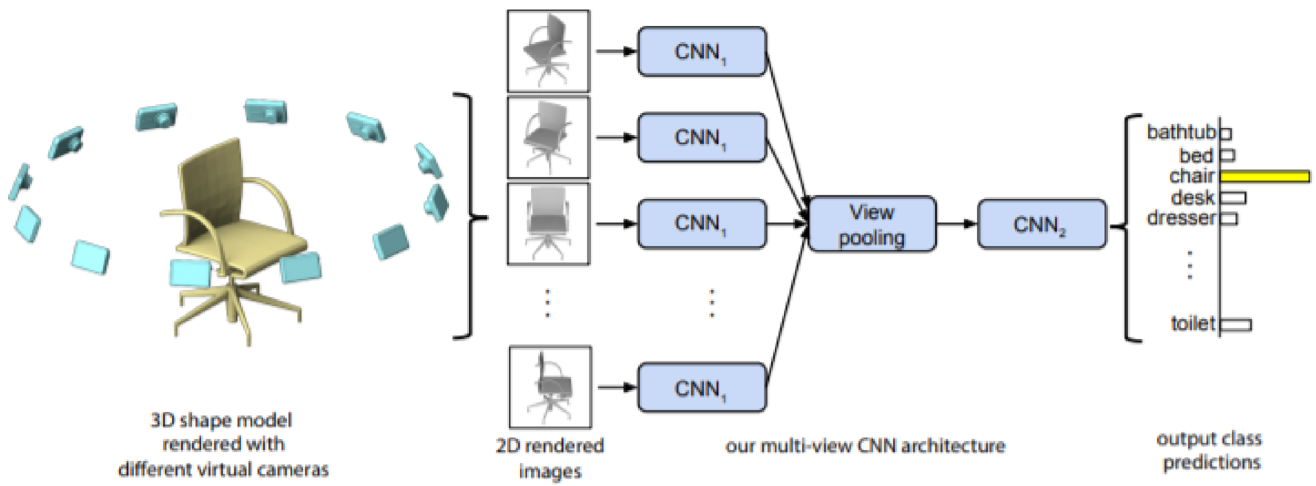






Multi-View Images

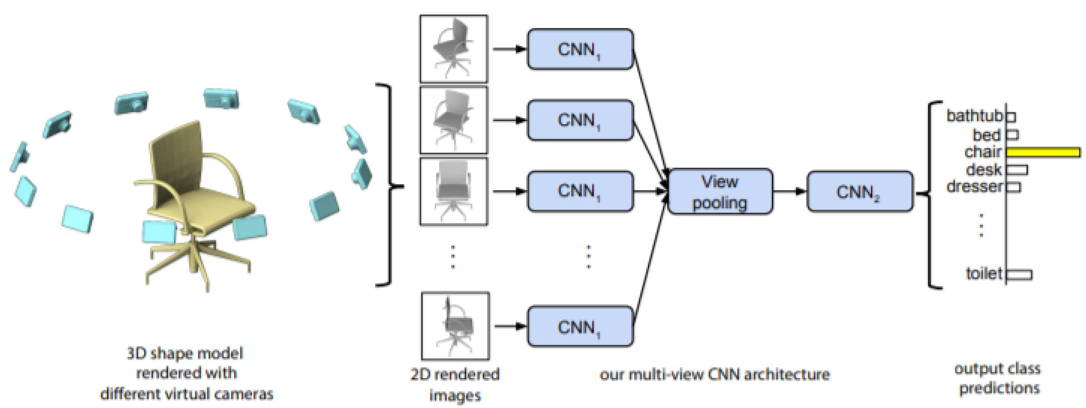
- Represent a 3D object with images captured from multiple views
- MVCNN for object recognition
  - Extract image features with shared CNN
  - Aggregate features from all views with view pooling



Multi-view convolutional neural networks for 3d shape recognition, ICCV 2015

MVCNN (cont'd)

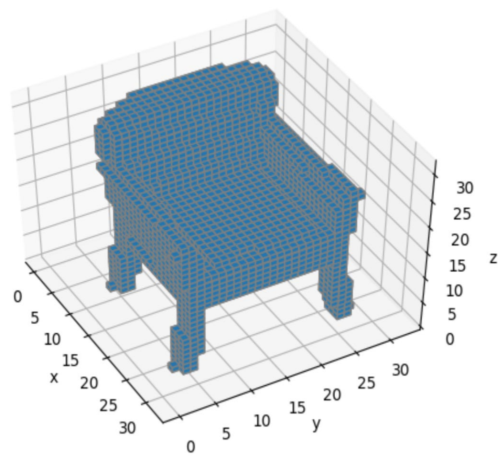
- Pros
  - Can leverage SOTA or pre-trained CNNs for excellent performance
- Cons
  - Setting not necessarily practical
  - Sensitive to (1) viewpoint selection, (2) invisible viewpoint, (3) geometry
  - Vulnerable to occlusion or
  - No information on

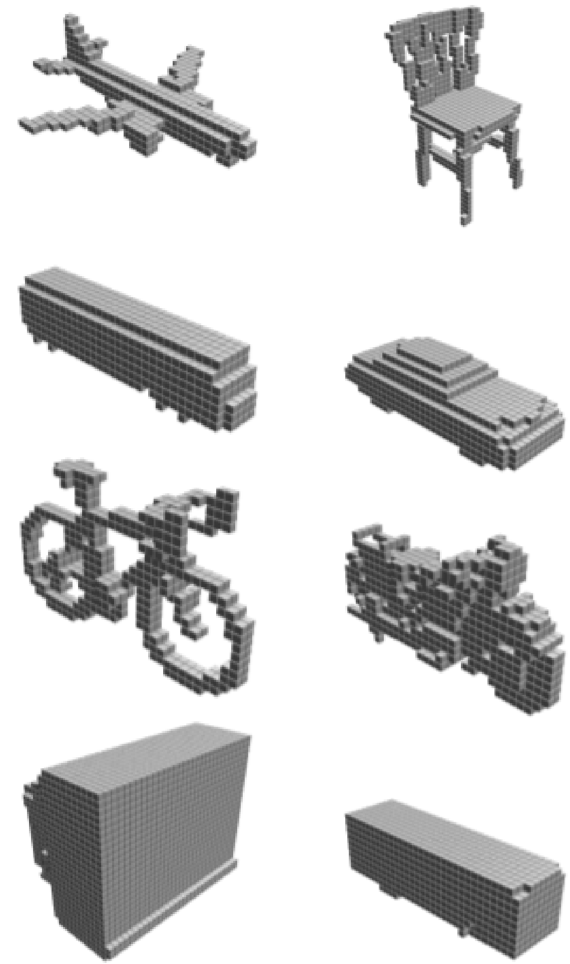


Multi-view convolutional neural networks for 3d shape recognition, ICCV 2015

Voxels

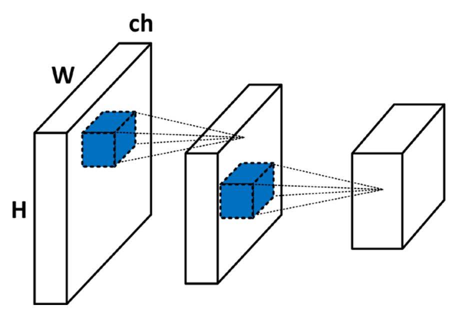
- Grids in fixed resolution $x \times y \times z$
- Each grid contains 0/1: occupancy



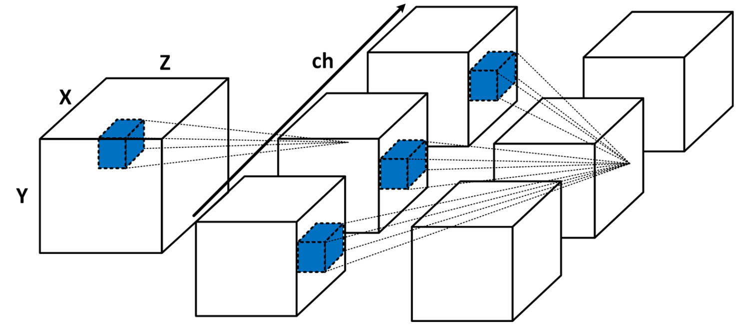


3D Convolution for Voxels

- Convolution for 2D images $y(i,j) = \sum_m \sum_n x(m,n) \cdot h(i-m, j-n)$
- Convolution for 3D voxels $y(i,j,k) = \sum_m \sum_n \sum_p x(m,n,p) \cdot h(i-m, j-n, k-p)$



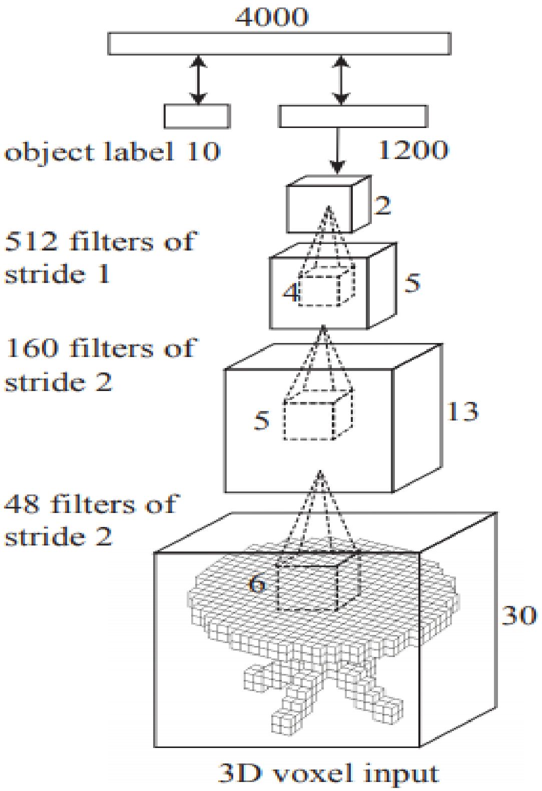
2D CNN



3D CNN

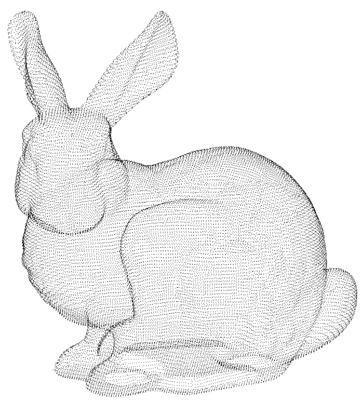
3D ShapeNets

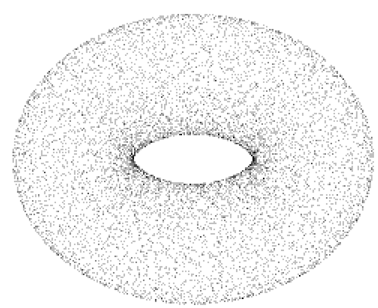
- 3D object classification with voxels via 3D CNNs
- Accuracy only ~77%, not comparable to MVCNN
  - Any explanation?
- Remarks
  - Pros:
    - Represent shape geometry
    - Easy to operate with 3D CNN
  - Cons:
    - Memory consuming...why?

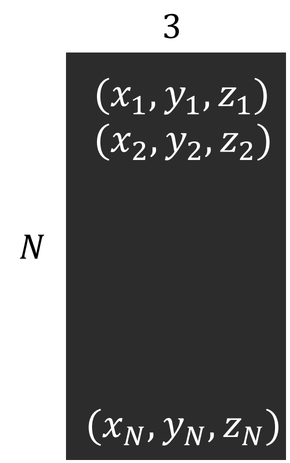


Point Cloud

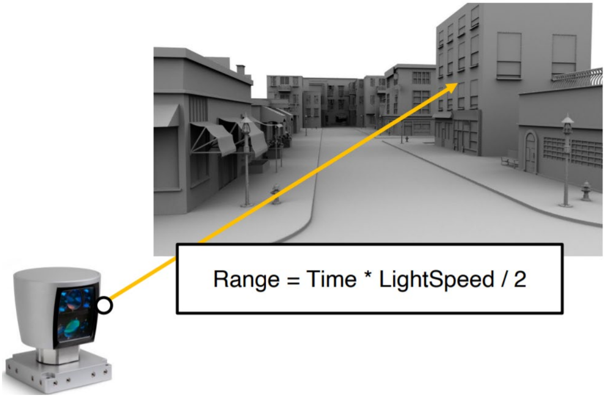
- Point cloud is a point set, representing 3D shapes
- Each point is represented by coordinates (x, y, z)
- Point cloud is stored as a $N \times 3$ matrix
  (N: point number, 3: coordinates)



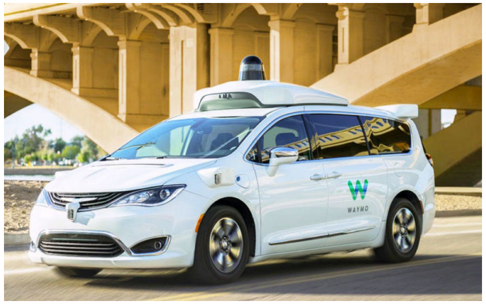




Point Cloud (cont’d)

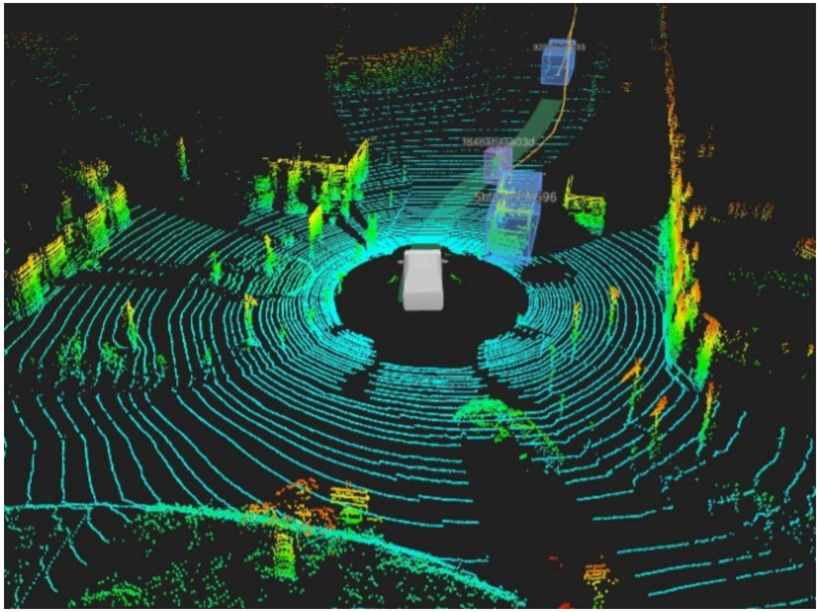


- Point cloud can be obtained from LiDAR sensors
- Can capture scene geometry



Autonomous driving

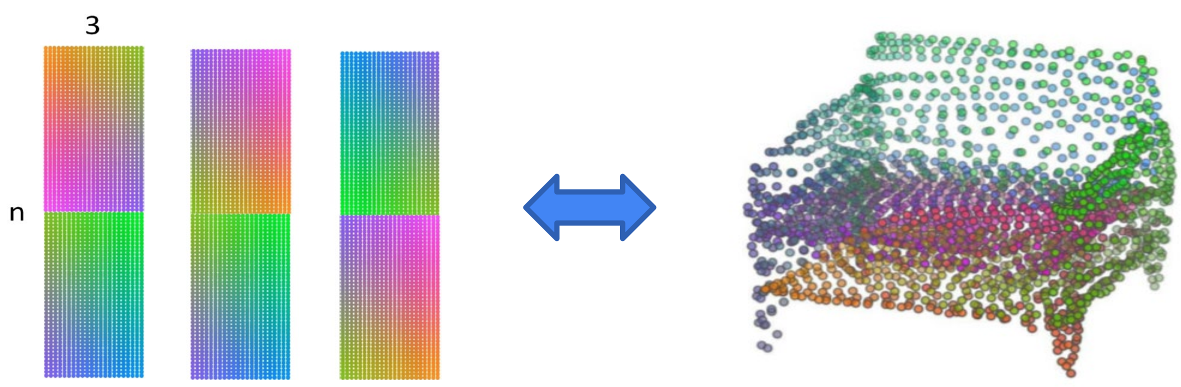
Augmented Reality (AR)



Reference: Robot Perception, taught by Prof. Shenlong Wang, UIUC
https://shenlong.web.illinois.edu/teaching/cs598fall21/assets/slides/lecture3_sensors.pdf

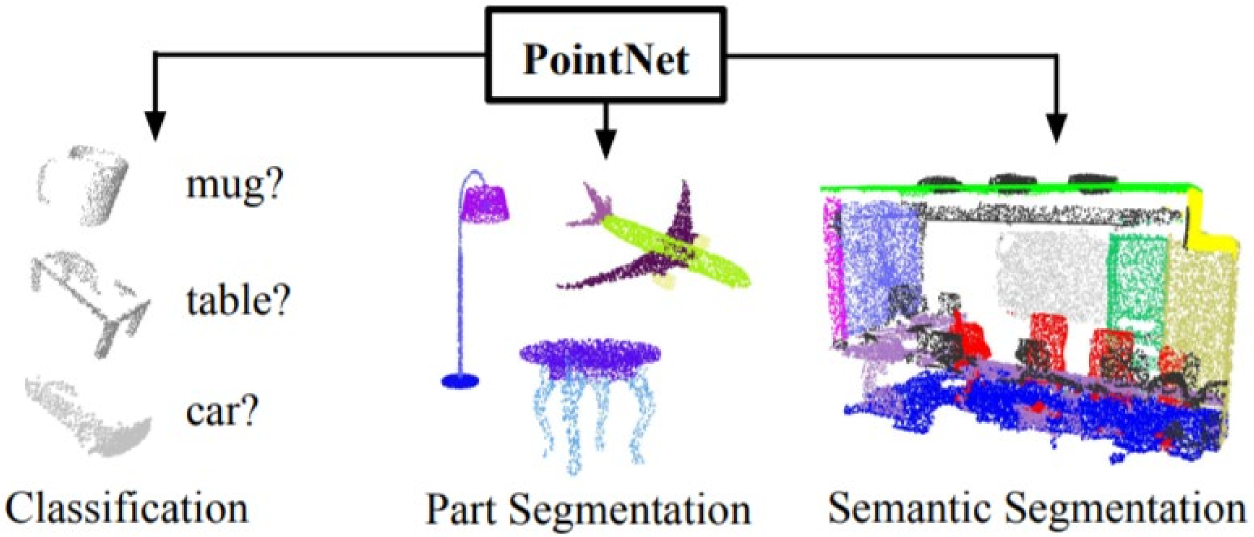
Challenges in Point Cloud

- Can we directly apply CNN on point cloud?
    - No, because point cloud is not grid-structured.
- The shape object can be represented in different orders
- Shape transformation not described (e.g., translation, rotation…)



PointNet

- Goal: Point cloud classification & segmentation

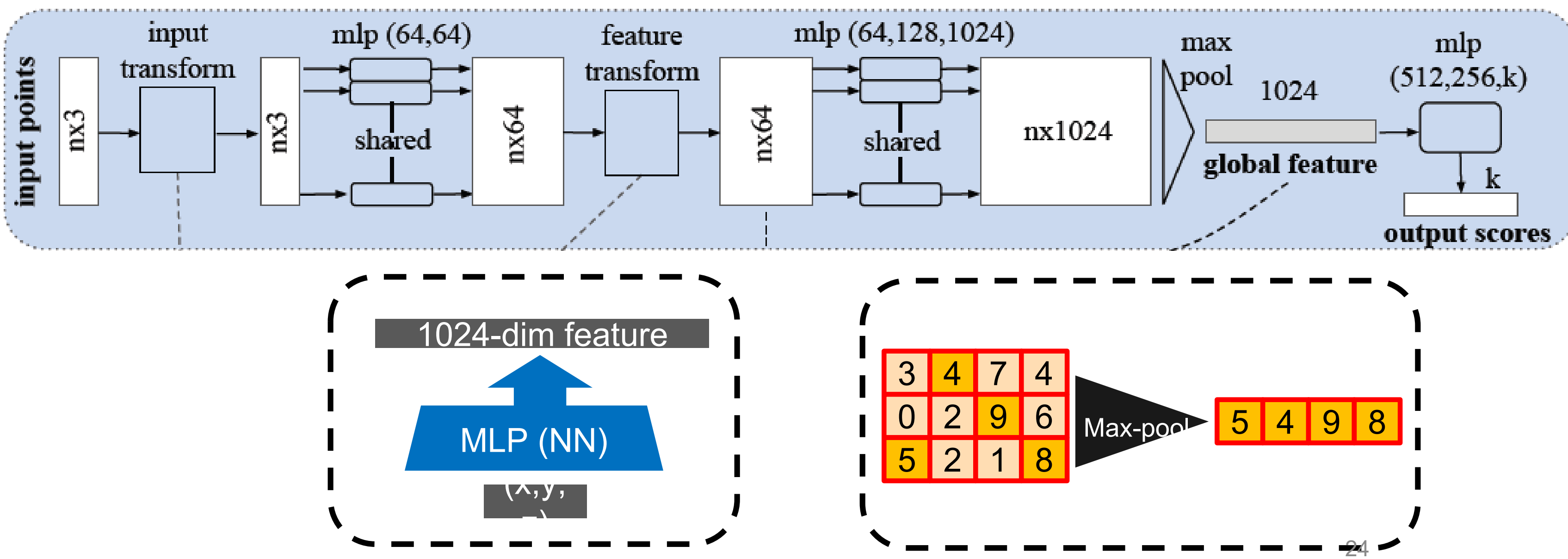


PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

PointNet

- Goal: Point cloud classification & segmentation
- Classification

Classification Network



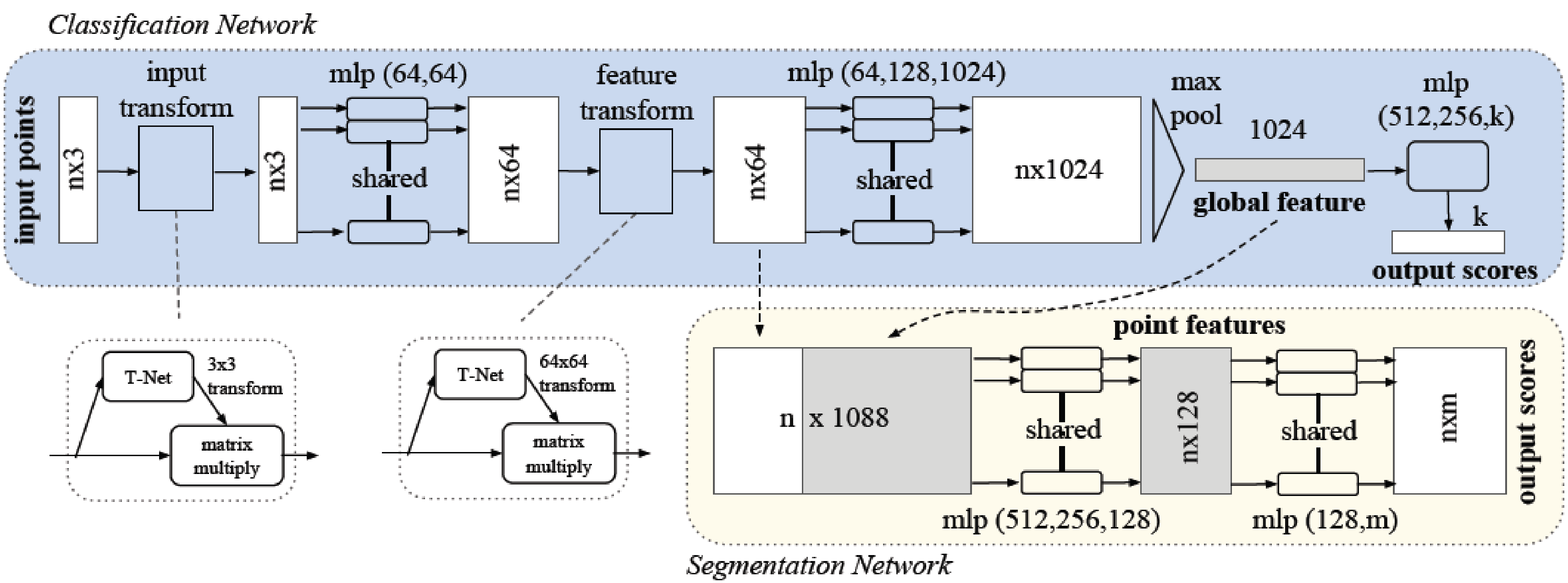
Multi-Layer Perceptron

Channel-wise max-pooling

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

PointNet

- Goal: Point cloud classification & segmentation
- Classification
- Segmentation



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

PointNet

- Goal: Point cloud classification & segmentation
- Classification & segmentation
- Qualitative results

Point: (xyz, rgb)

Part segmentation

Scene segmentation

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

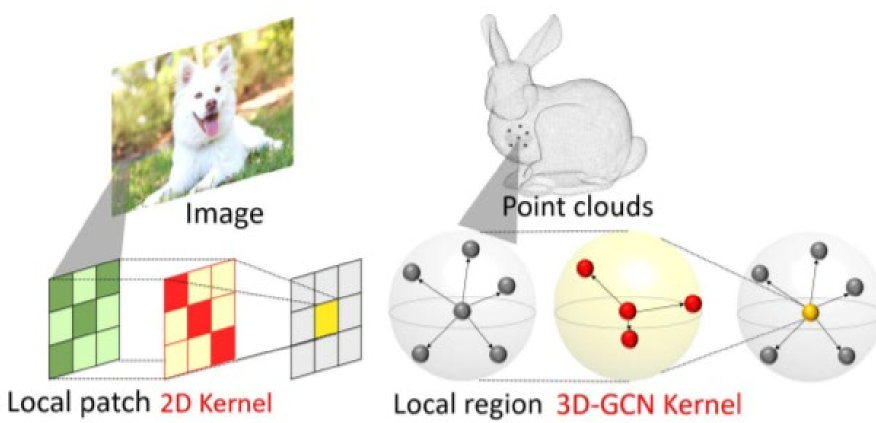
PointNet

- Goal: Point cloud classification & segmentation
- Classification & segmentation
- Qualitative results
- Remarks
  - Pros: extract features from unordered points
  - Cons:
    - Outlier/noisy point cloud data
    - Cannot capture…
    - Might not robust to transformation like

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

Extensions of PointNet

- **PointNet++**: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, NIPS 2017
- **Dynamic Graph CNN** for Learning on Point Clouds, TOG 2019
- **KPconv**: Flexible and deformable convolution for point clouds, ICCV 2019
- **Convolution in the cloud**: Learning deformable kernels in 3D graph convolution networks for point cloud analysis, CVPR 2020 (VLLab @ NTU)
- **Variational Transformer** for Dense Point Cloud Semantic Completion, NeurIPS 2022 (VLLab @ NTU)

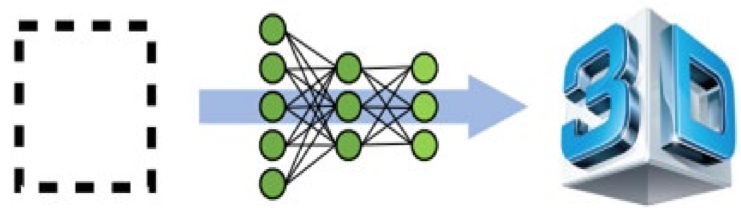


What to Cover Today?

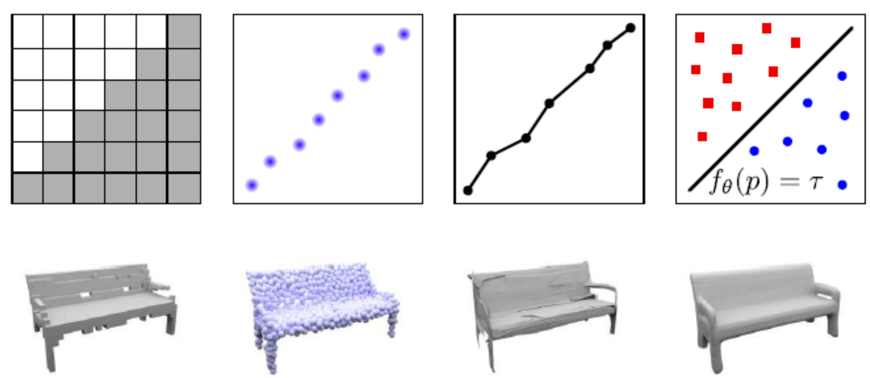
- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
  - Extension of NeRF
  - Advanced Topics of NeRF

3D Reconstruction

- Reconstruct 3D shapes/scenes from partial observations
  - Single/multi-view images
  - Videos
  - Incomplete point cloud

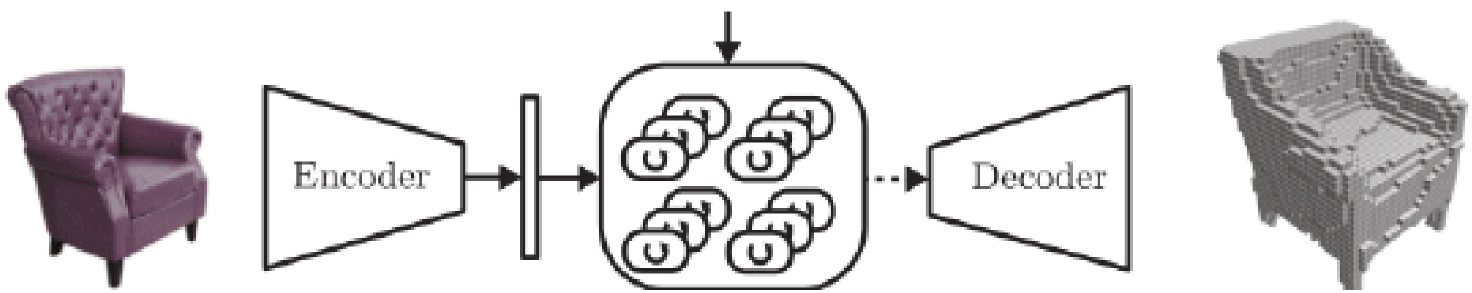


- In this part, we will talk about how to reconstruct
  - ~~Depth~~
  - Voxels
  - Point cloud
  - ~~Mesh~~
  - Function



3D R2N2 for Voxel Reconstruction

- 3D Recurrent Reconstruction Neural Network (3D-R2N2)
  - Input: one or multiple images of an object
  - Output: voxel representation

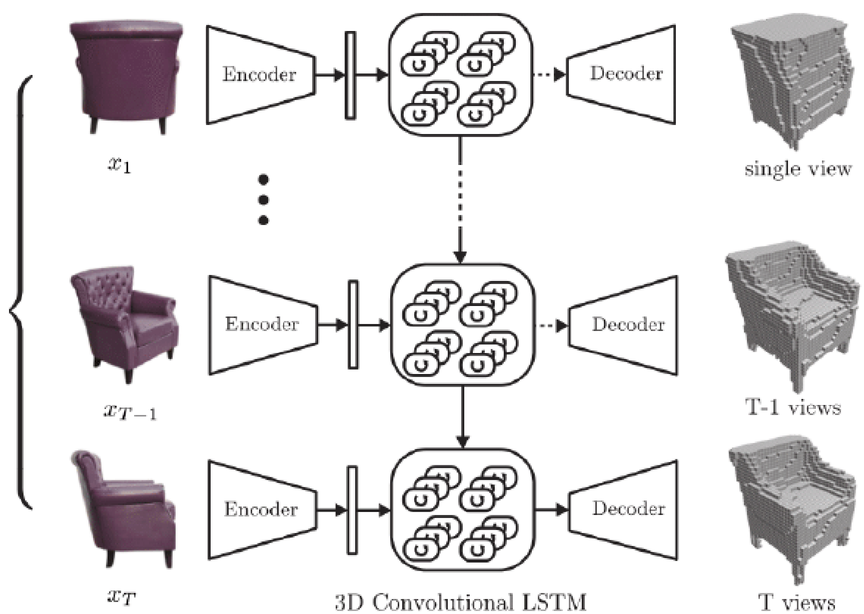


3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, ECCV 2016

3D R2N2 for Voxel Reconstruction

- 3D Recurrent Reconstruction Neural Network (3D-R2N2)
  - Input: one or multiple images of an object
  - Output: voxel representation
  - A recurrent 3D CNN with voxel-wise BCE loss

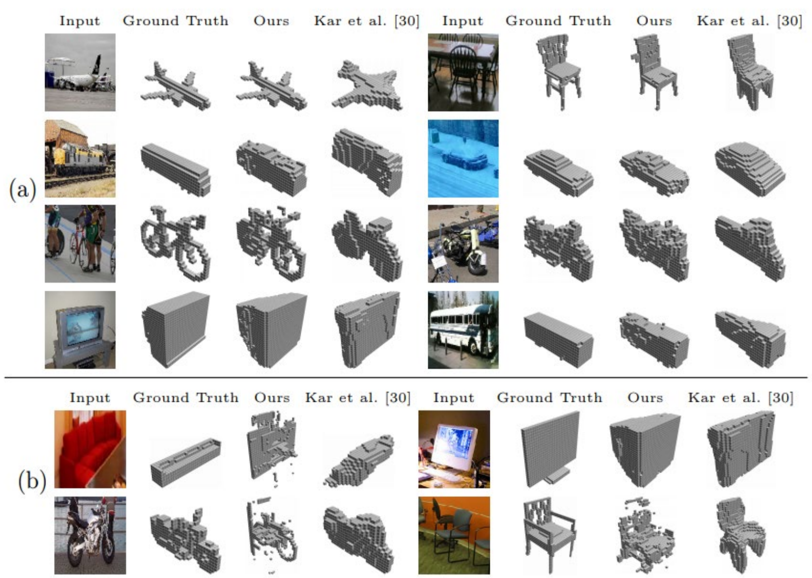
$$L(\mathcal{X}, y) = \sum_{i,j,k} y_{(i,j,k)} \log(p_{(i,j,k)}) + (1 - y_{(i,j,k)}) \log(1 - p_{(i,j,k)})$$

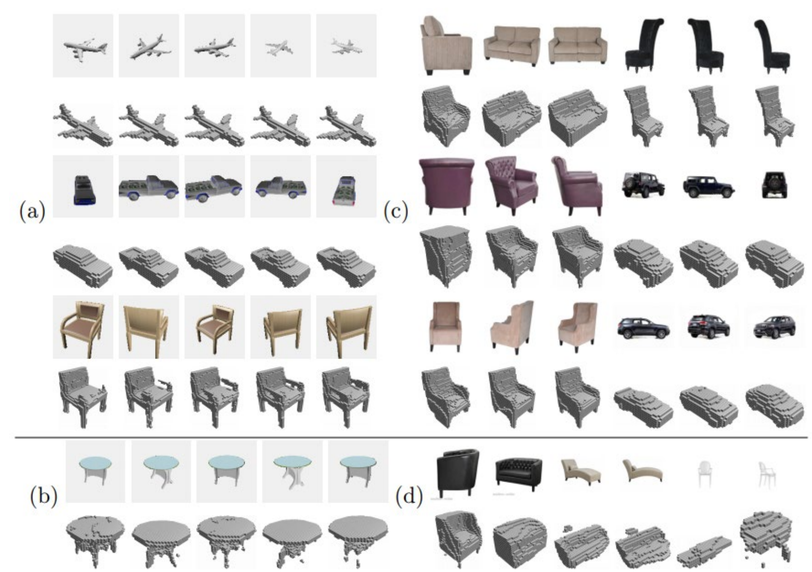


3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, ECCV 2016

3D R2N2 for Voxel Reconstruction

- 3D Recurrent Reconstruction Neural Network (3D-R2N2)
  - Input: one or multiple images of an object
  - Output: voxel representation
  - A recurrent 3D CNN with voxel-wise BCE loss
  - Examples (left: single image input, right: multiple image inputs)

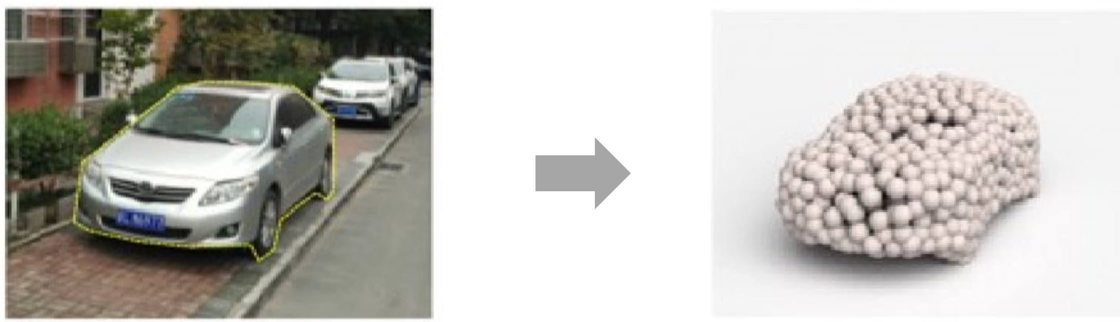




3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, ECCV 2016

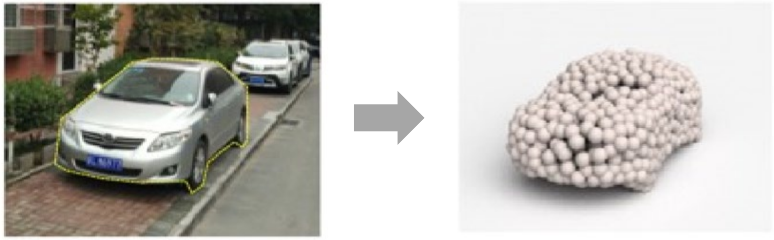
Point Set Generation

- 3D reconstruction via point cloud
- Input: single or multiple images
- Output: object point cloud

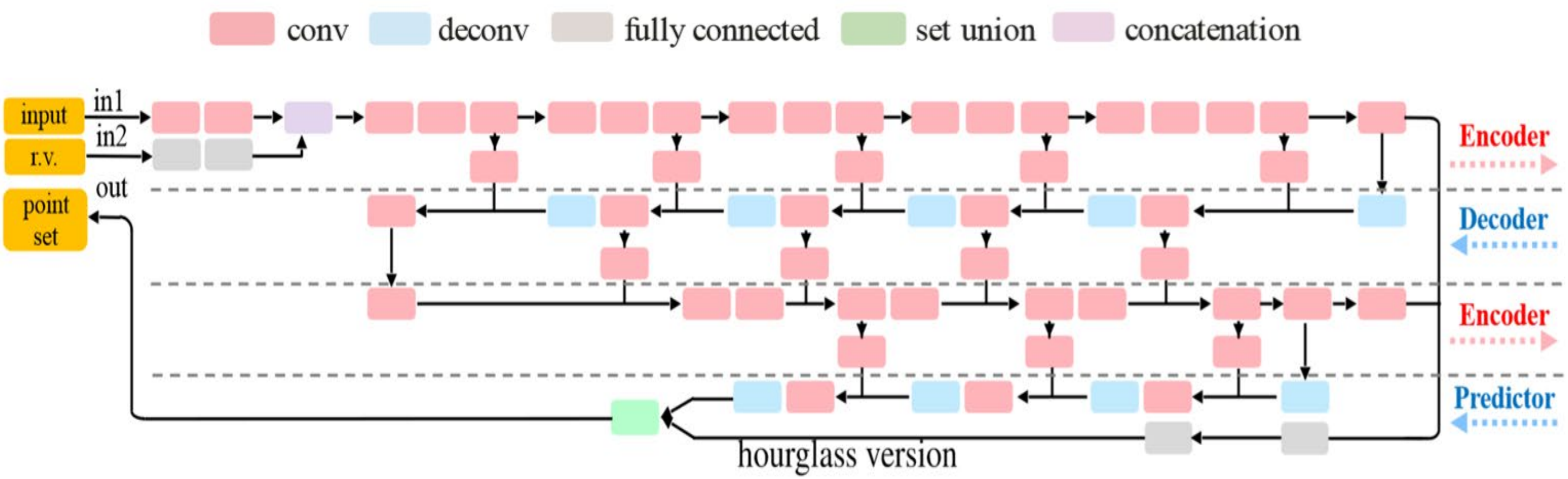


A Point Set Generation Net for 3D Object Reconstruction from a Single Image, CVPR 2017

Point Set Generation

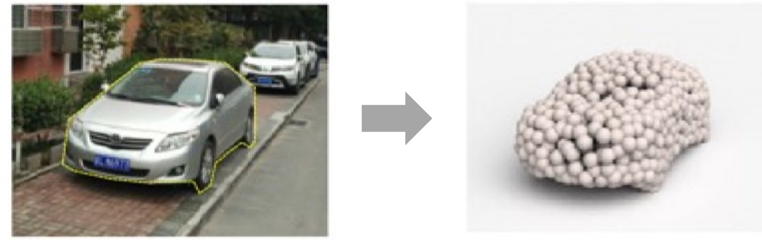


- 3D reconstruction via point cloud
- Input: single or multiple images
- Output: object point cloud (unordered)
- Two-branch prediction: fully connected for intrinsic structure + deconvolution for smooth surfaces



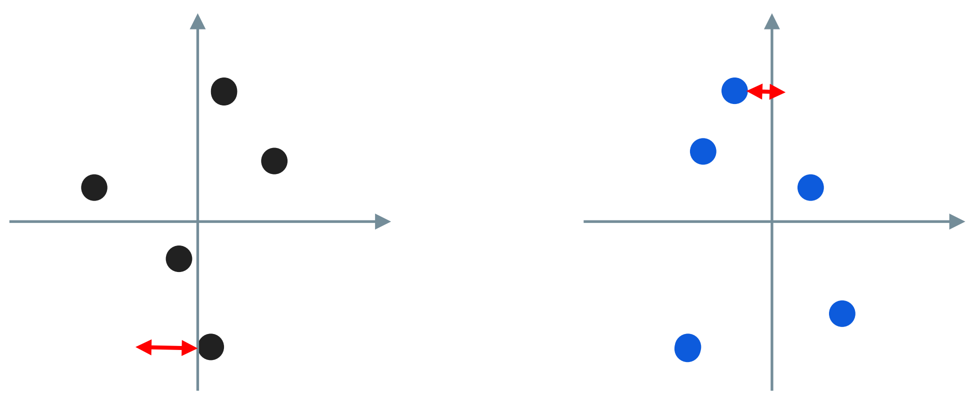
A Point Set Generation Net for 3D Object Reconstruction from a Single Image, CVPR 2017

Point Set Generation



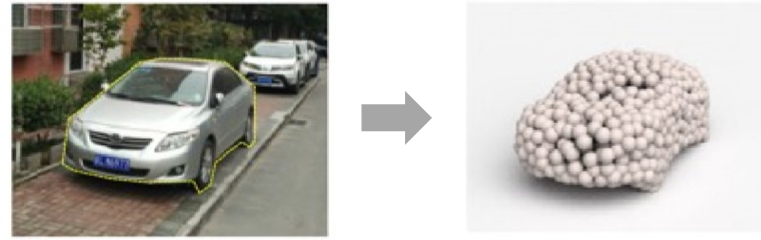
- 3D reconstruction via point cloud
- Input: single or multiple images
- Output: object point cloud (unordered)
- Two-branch prediction
- Loss function: **Chamfer distance**

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$



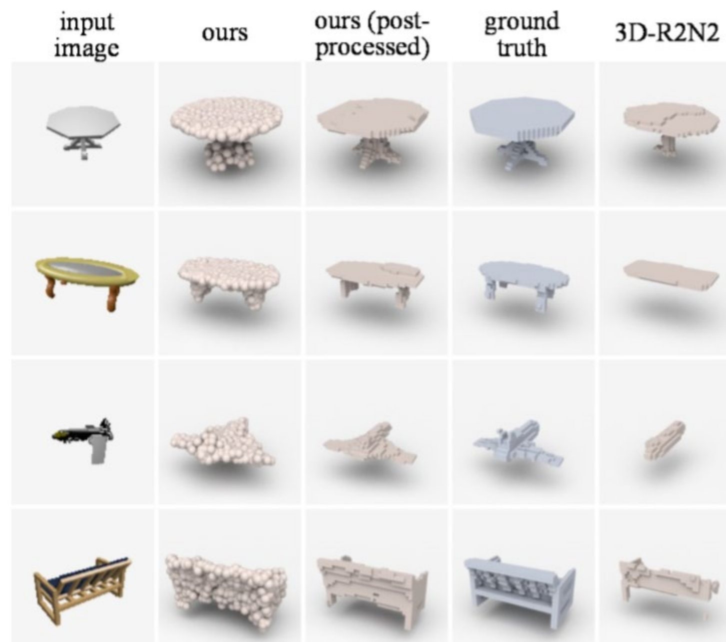
A Point Set Generation Net for 3D Object Reconstruction from a Single Image, CVPR 2017

Point Set Generation



- 3D reconstruction via point cloud
- Input: single or multiple images
- Output: object point cloud (unordered)
- Two-branch prediction & loss function
- Example results

Figure 12. Visualization of points predicted by the deconvolution branch (blue) versus the fully connected branch (red).

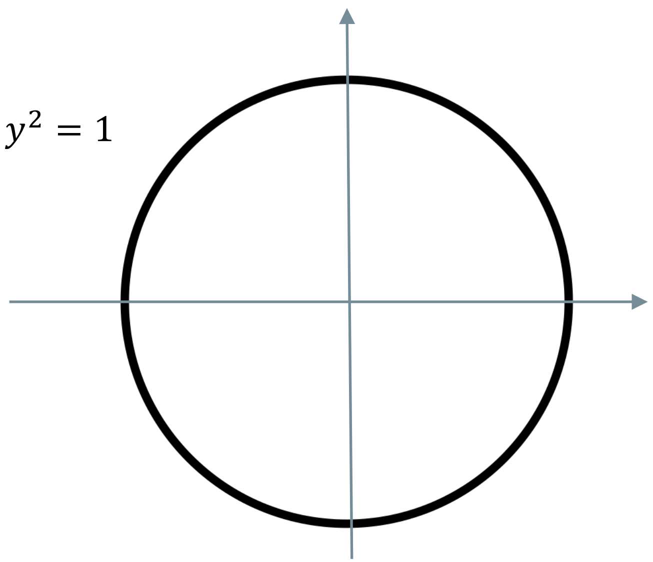


A Point Set Generation Net for 3D Object Reconstruction from a Single Image, CVPR 2017

Implicit Representation

- Represent shapes as “function”
- Tell us whether a point is on the surface

$x^2 + y^2 = 1$

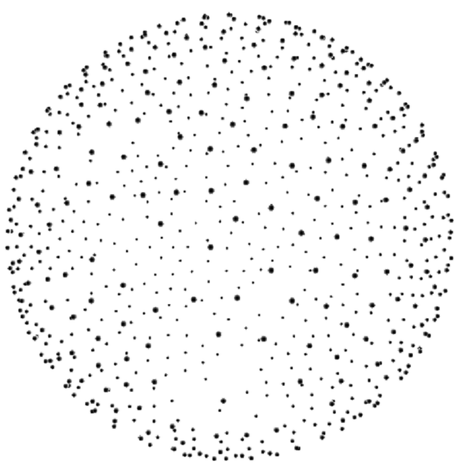


2D circle

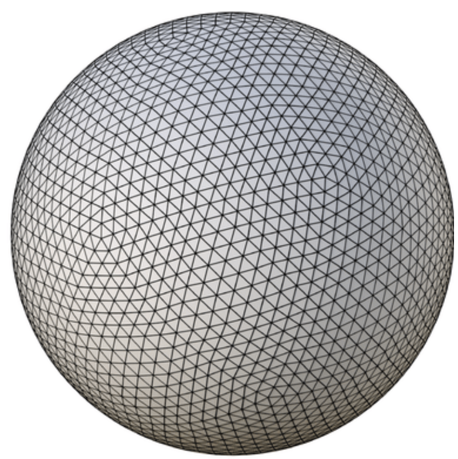
Q: Are these points on the circle?
(0, 1)
(1, 0)
(1, 1)
(0, 0)

Implicit Representation

- Represent shapes as “function”
- Unit sphere: $f(x, y, z) = x^2 + y^2 + z^2 - 1 = 0$
  - Surface is the zero level set of $f(.)$

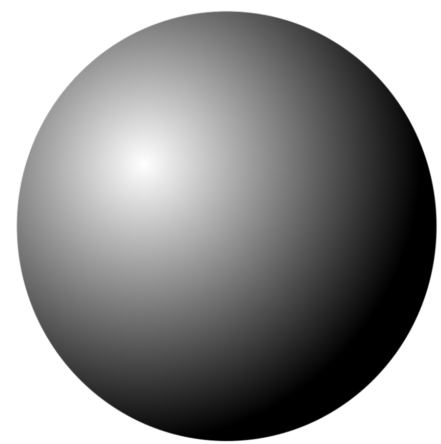


Point cloud



Mesh

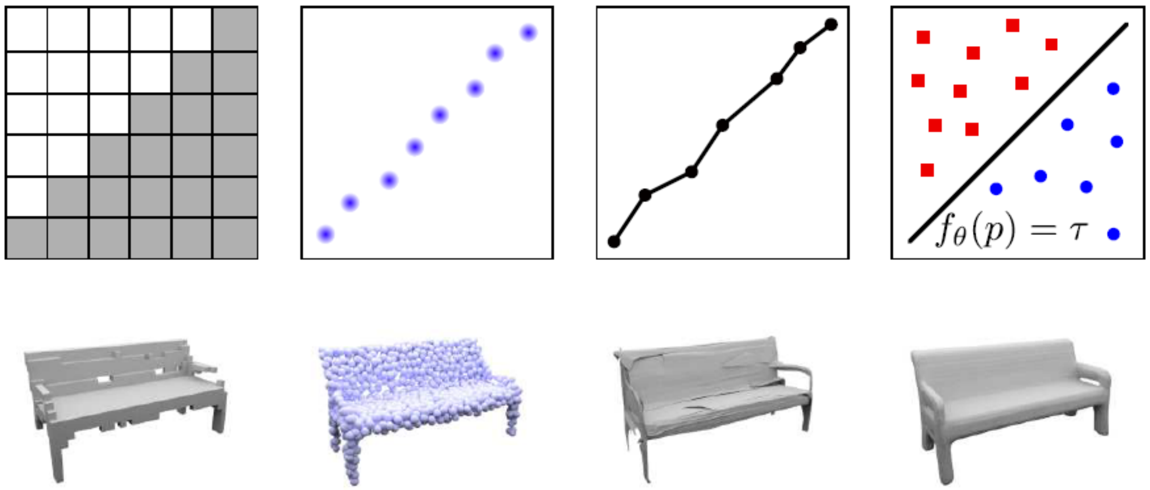
$x^2 + y^2 + z^2 = 1$



Implicit function

Occupancy Network

- Shape is a function that determines a point is inside/outside of it



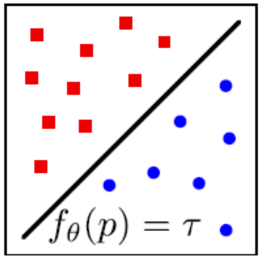
(a) Voxel (b) Point (c) Mesh (d) Ours

Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

Occupancy Network

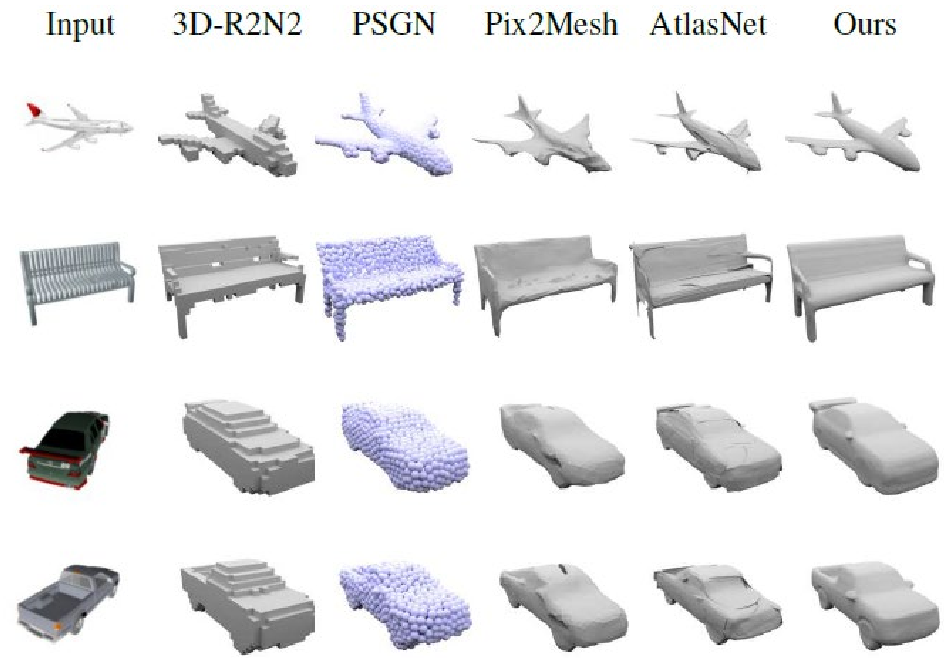
- Make model learn to predict occupancy at every possible 3D point $p \in R^3$
- Think of occupancy function as a “classifier”
- Condition on object feature X

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$



Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

Occupancy Network



Strength

- Flexible shape topology
- Arbitrary resolution
- Few model parameters

Weakness

- No info on…
- Require post-processing to get mesh
- Cannot handle complex scene

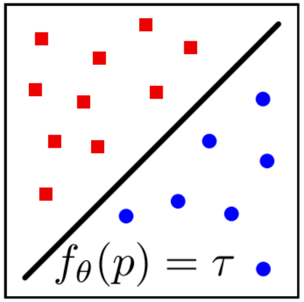
Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

What to Cover Today?

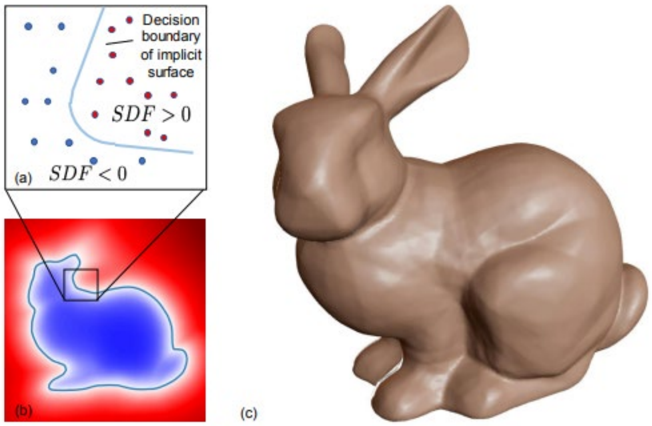
- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
  - Extension of NeRF
  - Advanced Topics of NeRF

Recap:
Neural Networks as a Continuous Shape Representation

Occupancy Networks
(Mescheder et al. 2019)
(x,y,z) -> occupancy



Deep SDF
(Park et al. 2019)
(x,y,z) -> distance



Pros: Compact and expressive parameterization

Cons: Limited rendering, difficult to optimize

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Many slides from Jon Barron and cs598dwh (UIUC)

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

Slide credit: cs598dwh

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Problem: Novel view synthesis (NVS)

Inputs: sparsely sampled images of scene

Outputs: *new* views of same scene

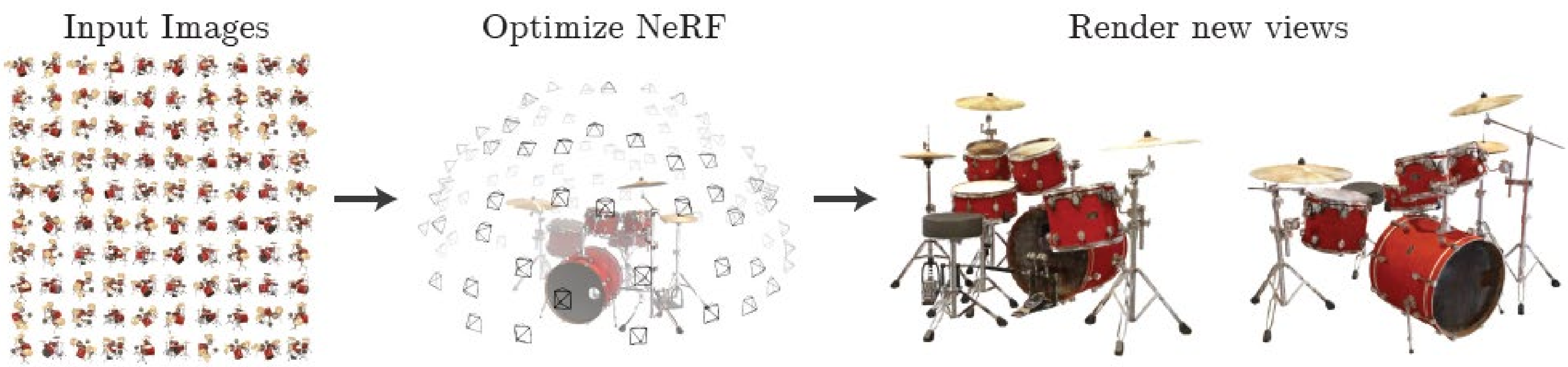
tancik.com/nerf

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

NeRF (Neural randiance field)

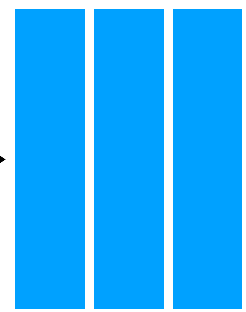
- Goal: learn 3D representation, and perform **novel view synthesis**
- Input: multi-view images + camera poses
- Output: 3D representation (neural radiance field)



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

NeRF (Neural randiance field)

$$\underbrace{(x, y, z}_{\text{Spatial location}}, \underbrace{\theta, \phi)}_{\text{Viewing direction}} \rightarrow F_\theta \rightarrow (\underbrace{r, g, b}_{\text{Output color}}, \underbrace{\sigma}_{\text{Output density}})$$



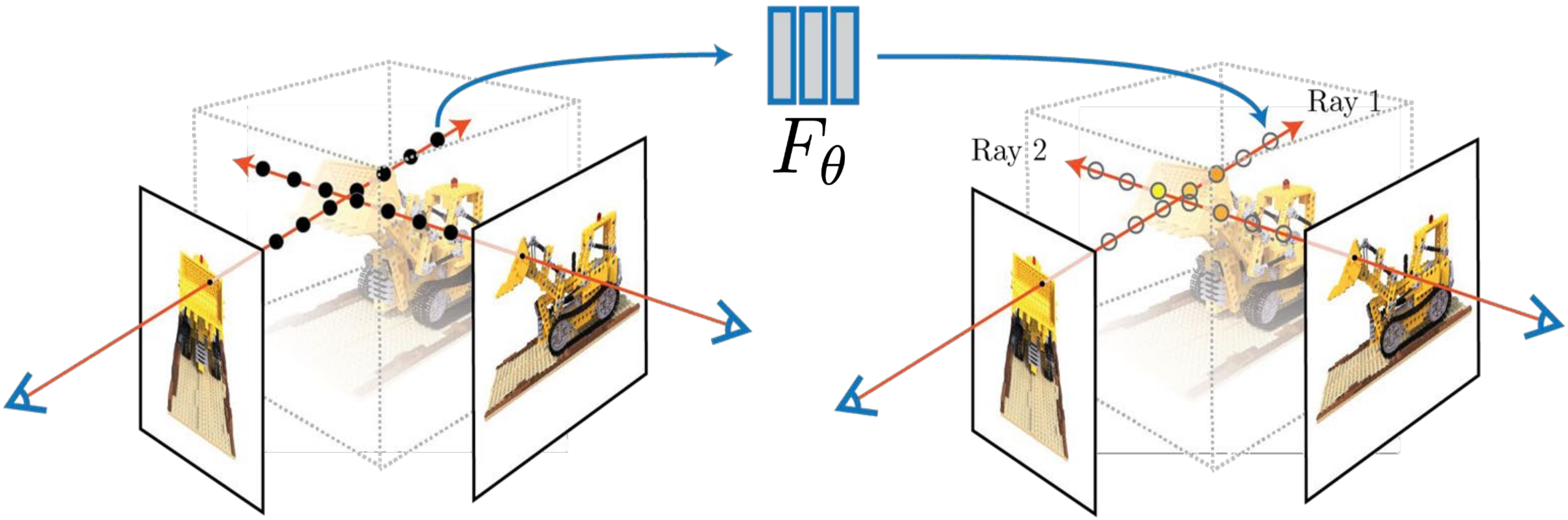
F_θ

Fully-connected neural network
9 layers,
256 channels

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Generate views with traditional volume rendering



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Generate views with traditional volume rendering

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

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

weights

colors

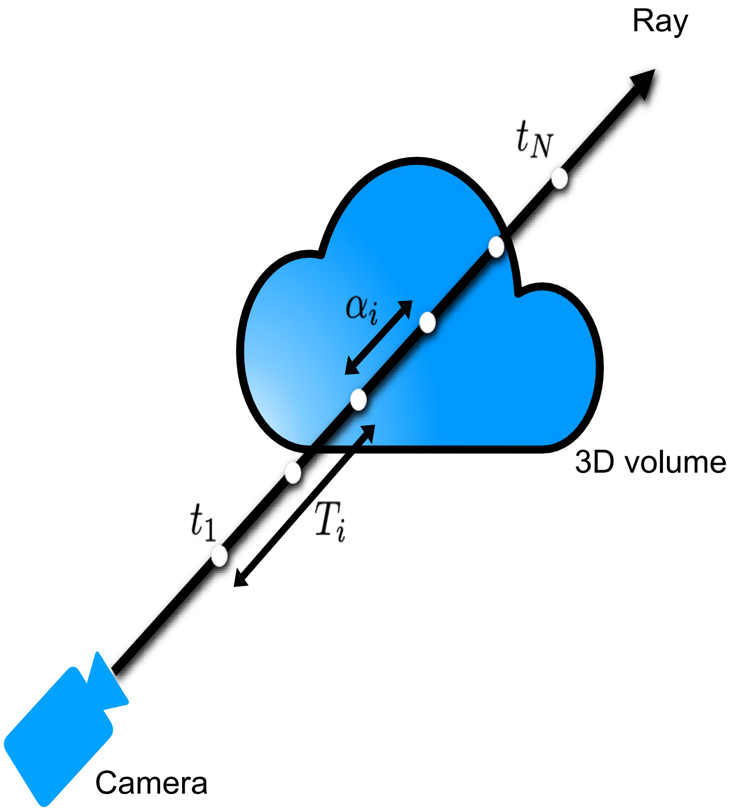
- How much light is blocked earlier along ray:

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

- How much light is contributed by ray segment *i*:

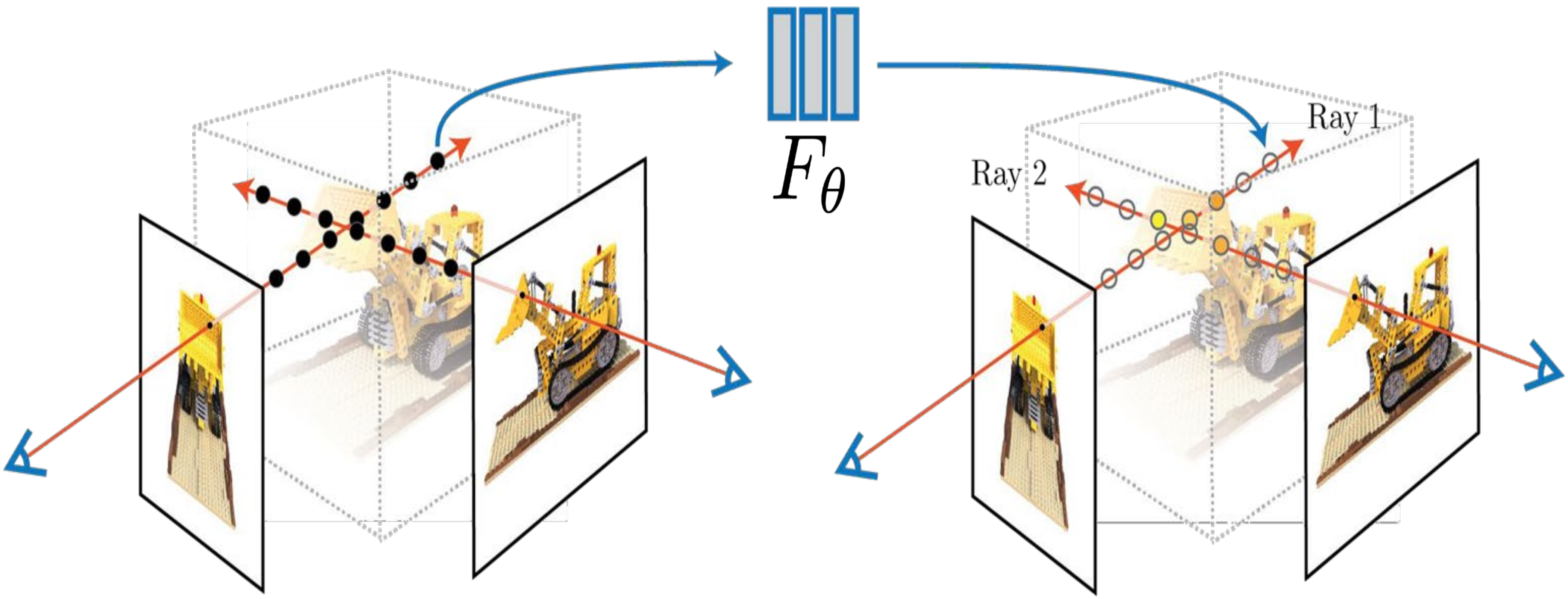
$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$

Density * Distance Between Points



Slide credit: Jon Barron

Optimize with gradient descent on rendering loss

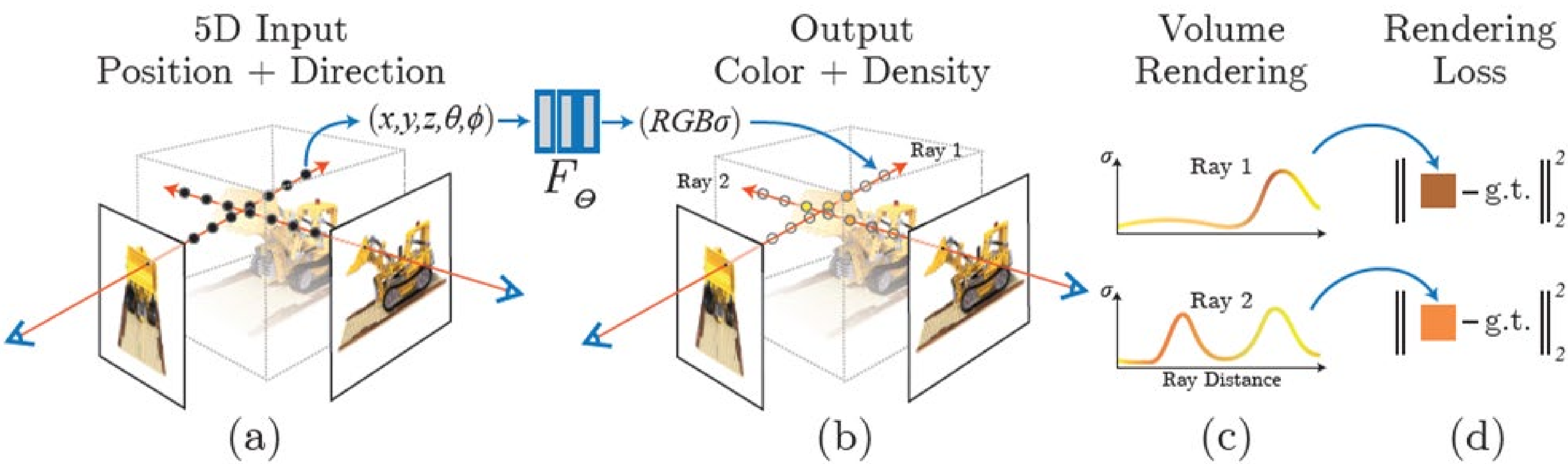


$$\min_{\theta} \sum_i ||\,\mathrm{render}_i(F_\theta) - I_i||^2$$

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

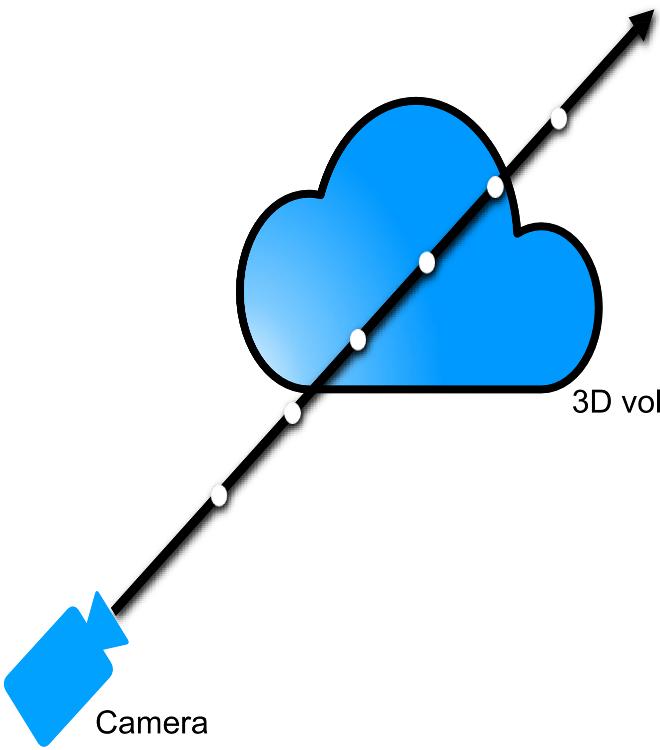
Training network to reproduce all input views of the scene



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Can we allocate samples more efficiently? --Two pass rendering



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

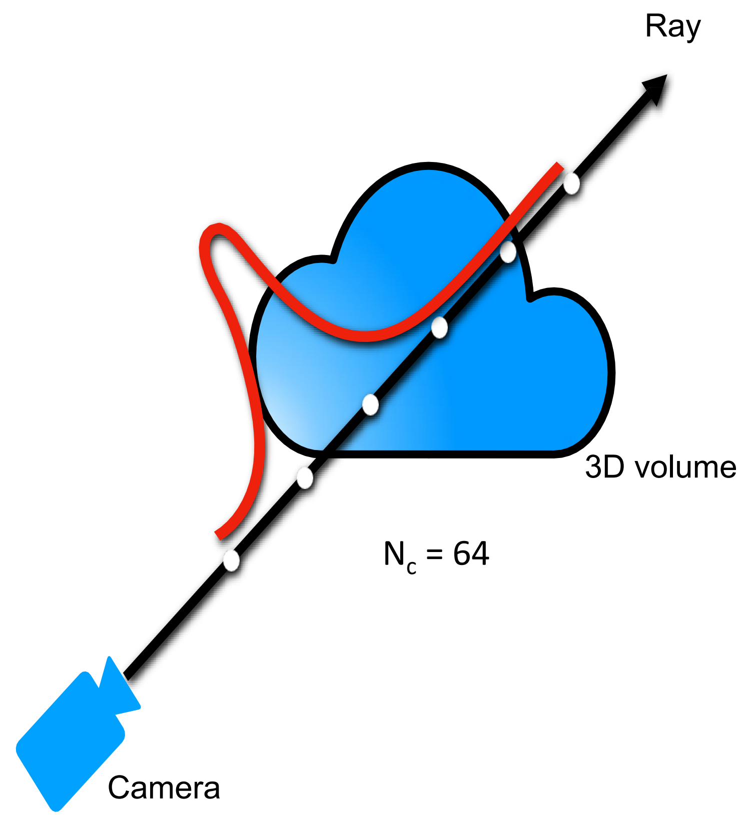
Slide credit: Jon Barron

Two pass rendering:coarse network

- Sparsely sample points along ray
- Serve as a coarse guidance

$$C \approx \sum_{i=1}^{N} \boxed{T_i \alpha_i} c_i$$

treat weights as probability
distribution for new samples



NeRF: Representing Scenes as Neural Radiance Fields for
View Synthesis, ECCV 2020

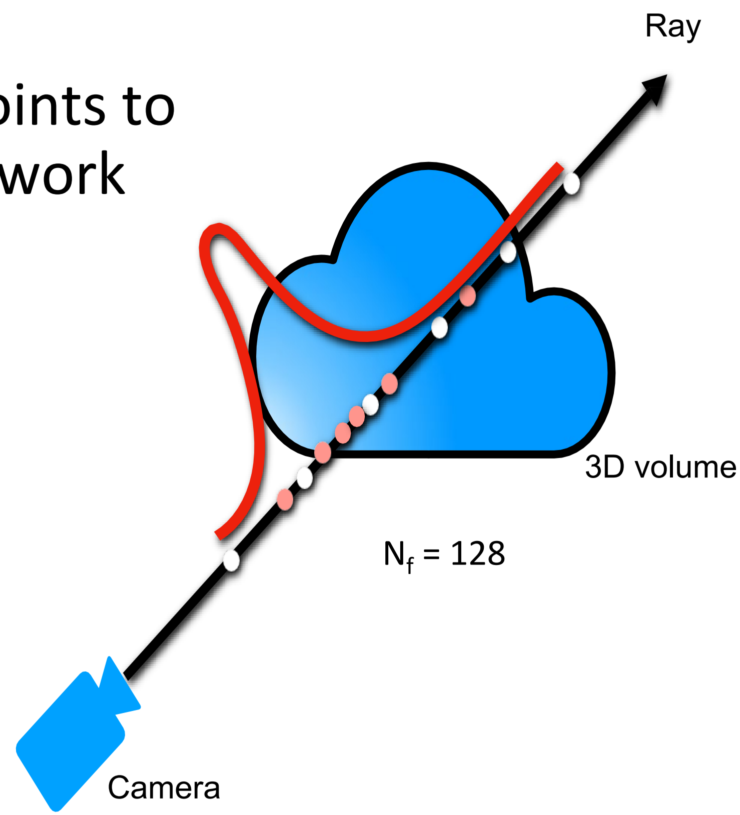
Slide credit: Jon Barron

Two pass rendering:fine network

- Use the coarse predicted density to resample new points along ray
- Together compute all $N_c + N_f$ points to calculate final color for fine network

$$C \approx \sum_{i=1}^{N} \boxed{T_i \alpha_i} c_i$$

treat weights as probability distribution for new samples



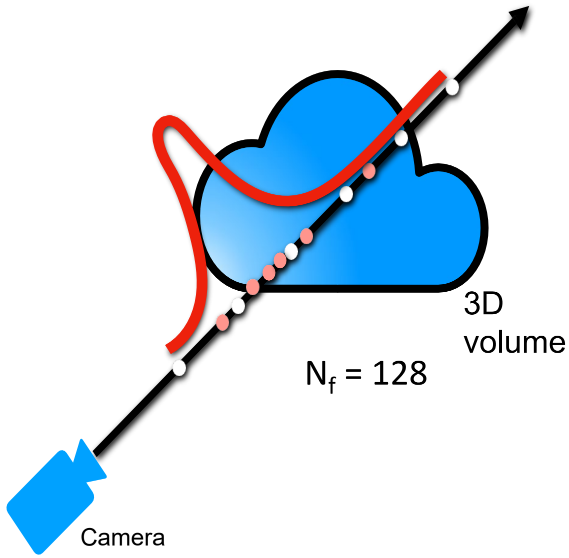
$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$ (coarse + fine)

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Two pass rendering: optimization

- Optimize coarse network and fine network together
- Only use the prediction of fine network when rendering a new scene



$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \boxed{\hat{C}_c(\mathbf{r})} - C(\mathbf{r}) \right\|_2^2 + \left\| \boxed{\hat{C}_f(\mathbf{r})} - C(\mathbf{r}) \right\|_2^2 \right]$$ (coarse + fine)

predicted color from coarse network

predicted color from fine network

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Positional encoding

NeRF (Naive)

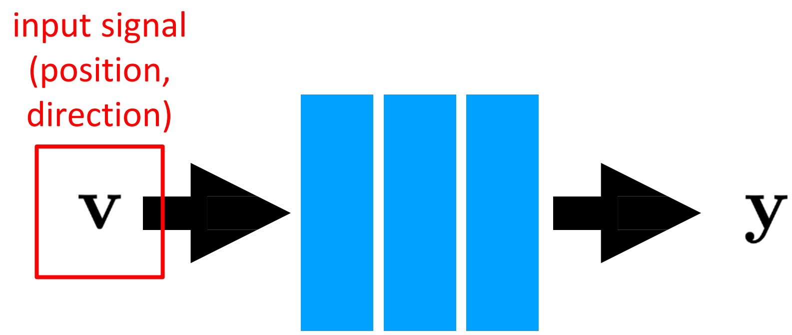
NeRF (with positional encoding)

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

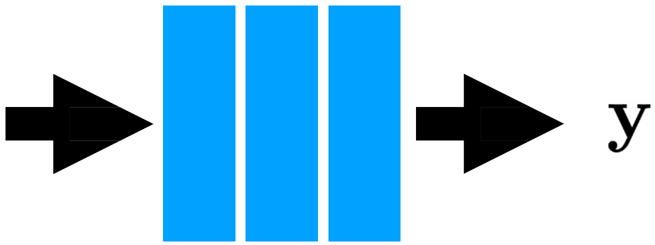
Slide credit: Jon Barron

Positional encoding

Naive



$$\begin{pmatrix} \sin(\mathbf{v}), \cos(\mathbf{v}) \\ \sin(2\mathbf{v}), \cos(2\mathbf{v}) \\ \sin(4\mathbf{v}), \cos(4\mathbf{v}) \\ \cdots \\ \sin(2^{L-1}\mathbf{v}), \cos(2^{L-1}\mathbf{v}) \end{pmatrix}$$

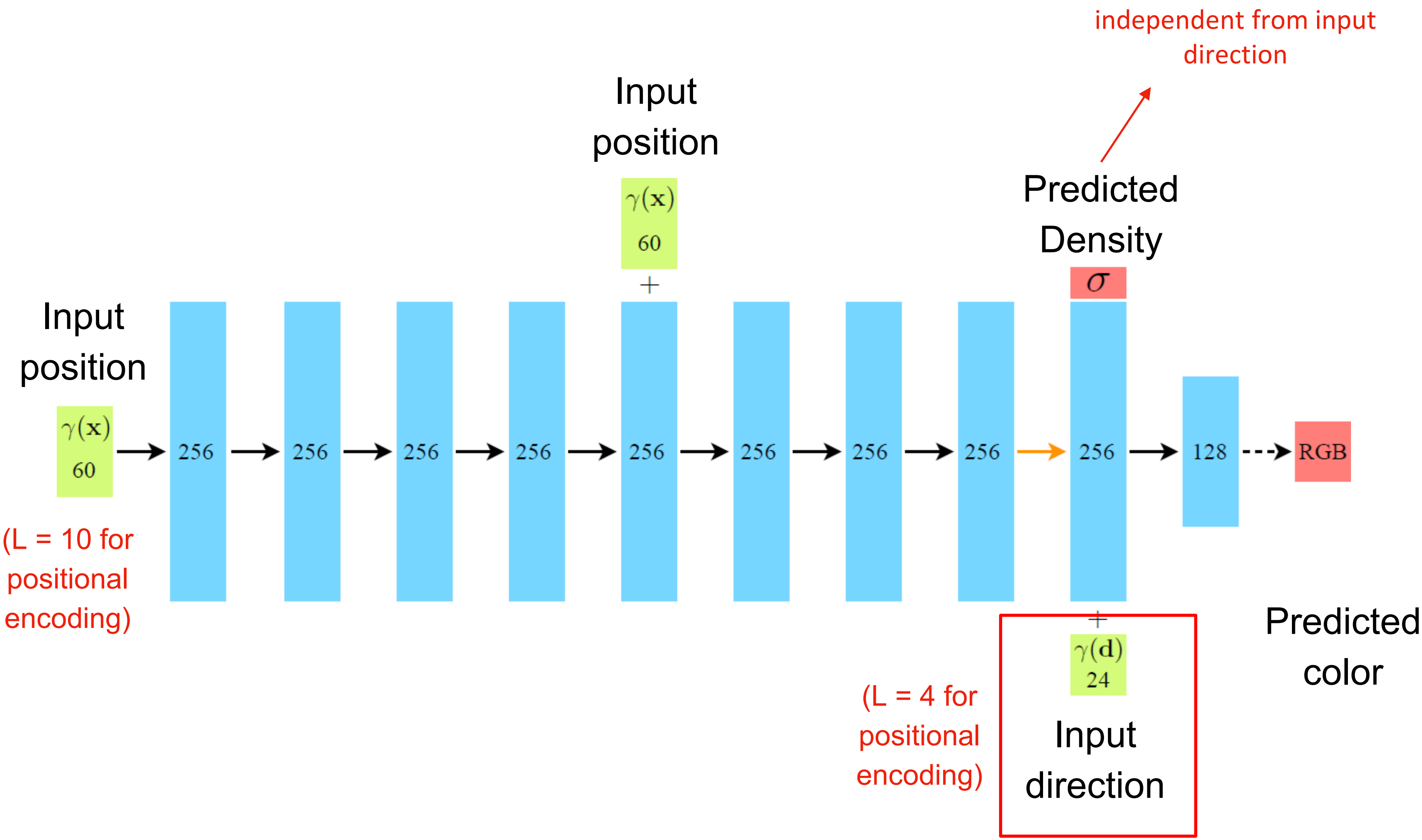


Positional encoding

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

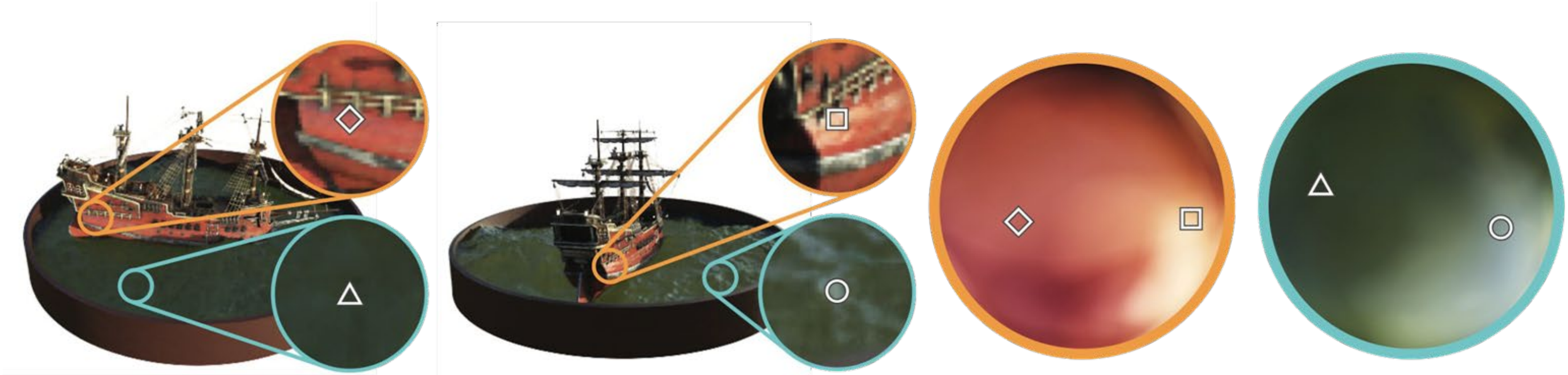
Slide credit: Jon Barron

Network Structure



Viewing directions as input

- The specular reflection (or other changes influenced by lighting) varies across different views



(a) View 1 (b) View 2 (c) Radiance Distributions

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Viewing directions as input

- The rendered color changes as the viewing direction
- L: image plane change with viewing direction
- R: fixing image plane while the viewing direction feeded to NeRF changes

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Viewing directions as input

- Another example

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Depth (geometry) Estimation

- The predicted density indicates the object surface
- The estimated depth perfectly shows
  the geometry of foreground object

NeRF: Representing Scenes as Neural Radiance Fields for
View Synthesis, ECCV 2020

Depth (geometry) Estimation

- Another example

NeRF: Representing Scenes as Neural Radiance Fields for
View Synthesis, ECCV 2020

Depth (geometry) Estimation

- By correctly estimate the depth of the scene, virtual objects are possible to interact with the real scene

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

NeRF: strength & weakness

Strength

- Photo-realistic texture
- Do not require 3D ground truth
- View-dependent effect

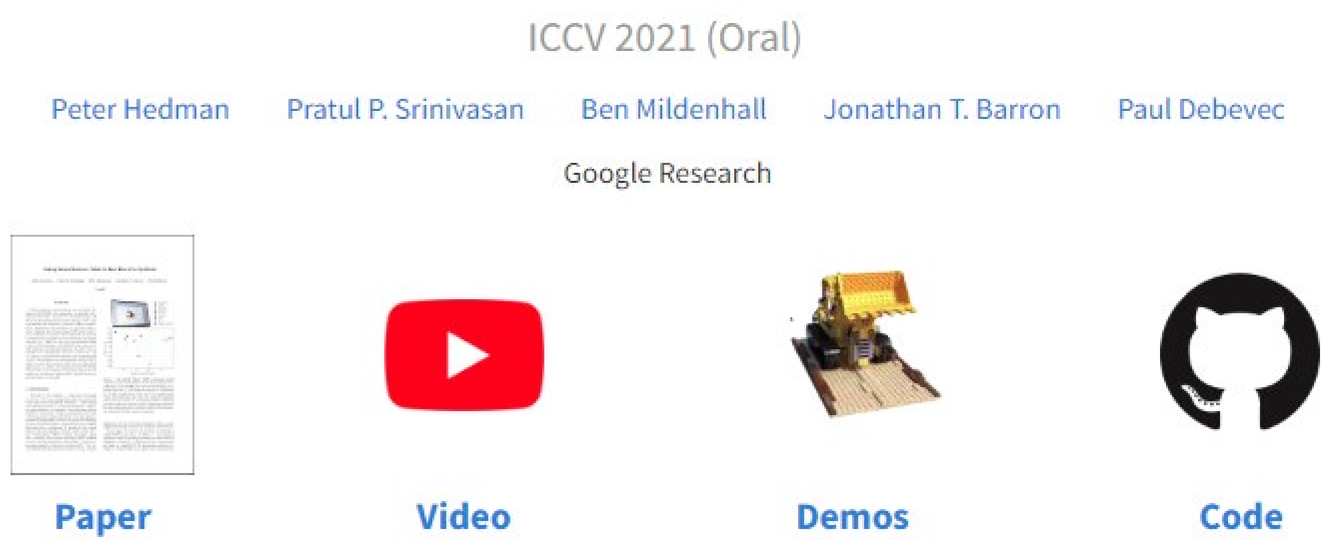
Weakness

- Only fit single scene
- Require much posed images
- Time-consuming rendering (30s per frame) <- Fatal for real-time applications !!

What to Cover Today?

- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
    - Extension of NeRF: Can We Do Faster??
    - Advanced Topics of NeRF

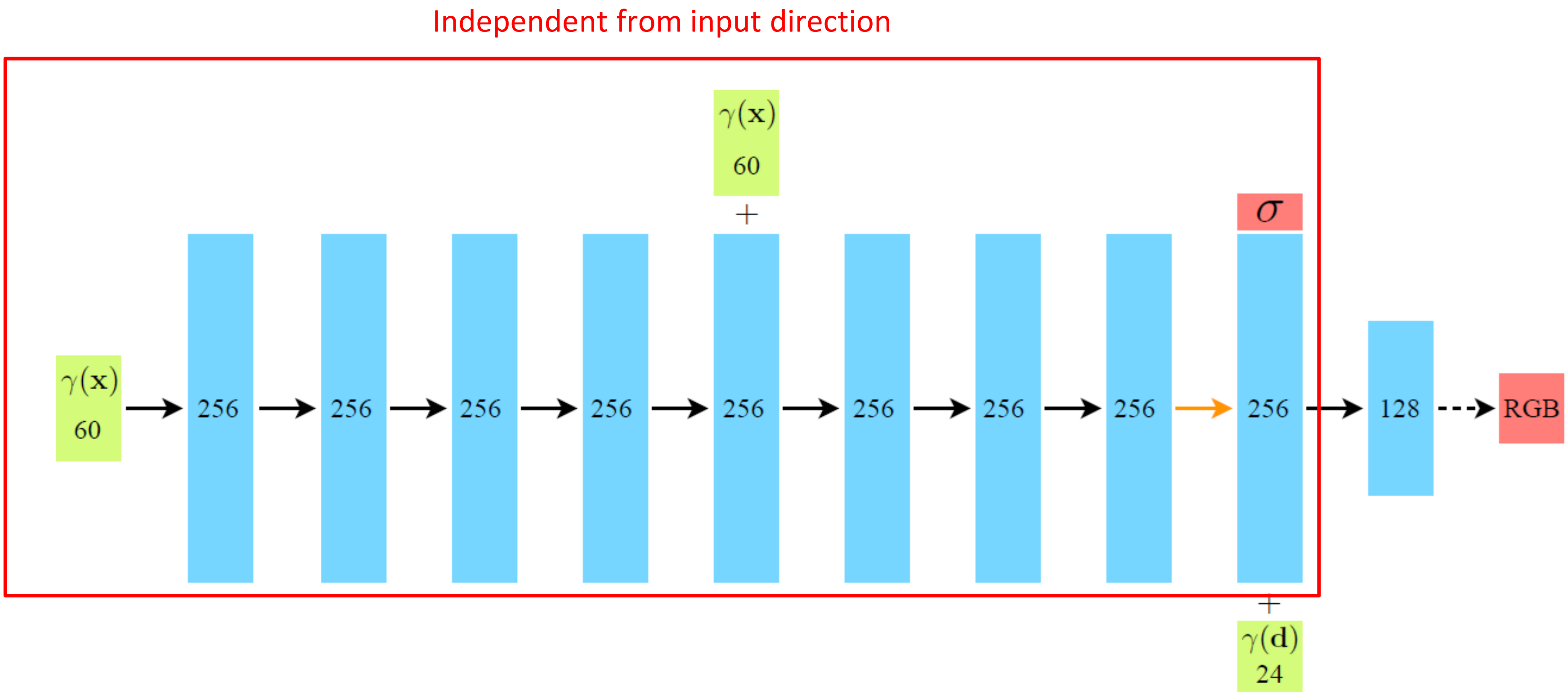
Baking Neural Radiance Fields for Real-Time View Synthesis



http://nerf.live/

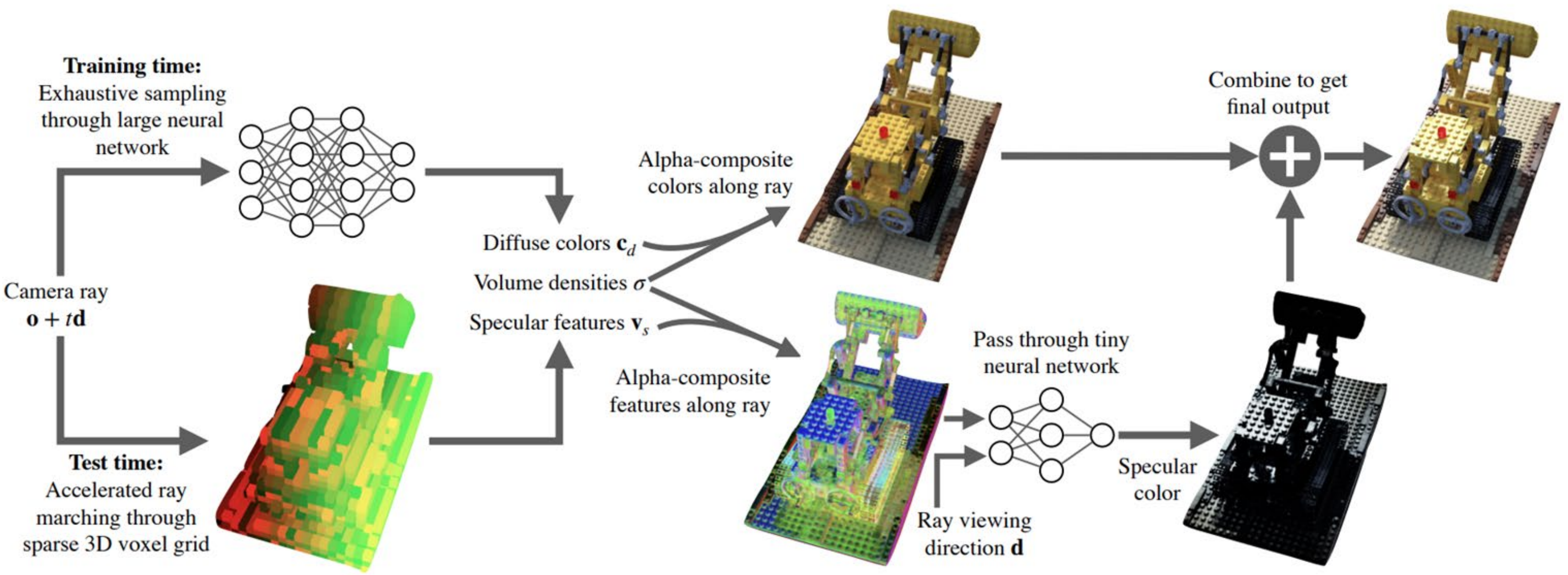
Basic idea

- In original NeRF, most information are independent from input direction
- Those information can be pre-computed and stored before rendering



Method: overview

- NeRF modified to output diffuse color, density, and 4-d specular features
- Color and features are accumulated along ray,
  and a small network produces a specular residual that is added to color

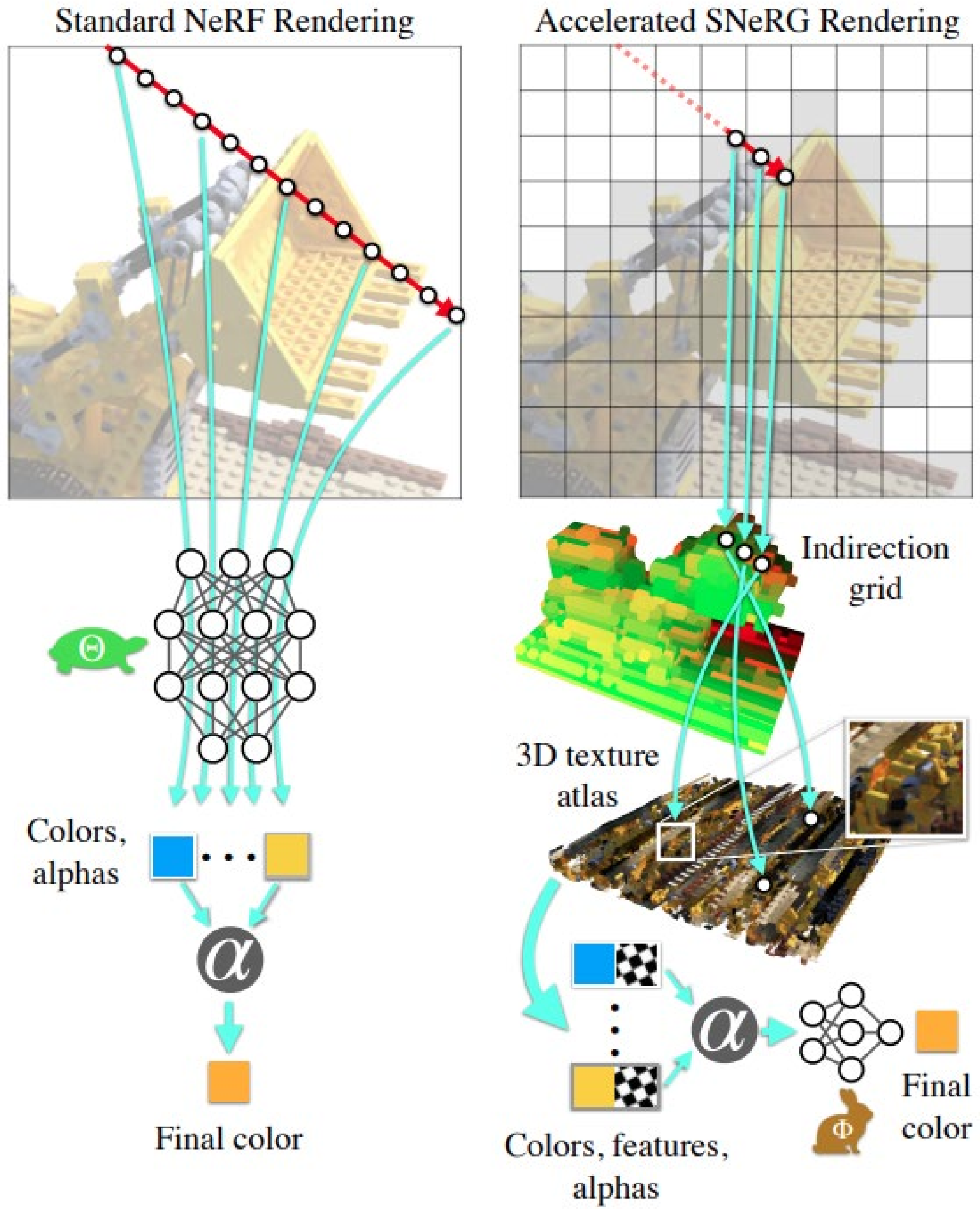


Slide credit: cs598dwh

Baking Neural Radiance Fields for Real-Time View Synthesis, ICCV 2021

Method: rendering

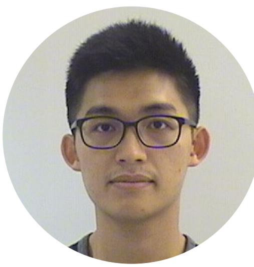
- Precompute diffused colors/features on voxel grid
- Voxels are stored sparsely and divided into local blocks
- In coarse grids, see if occupied; if so pointer to higher resolution color/feature info
- Compute specular component from features and add to color
- Result:
  30+ FPS on laptop,
  model < 100 MB



Slide credit: cs598dwh

Baking Neural Radiance Fields for Real-Time View Synthesis, ICCV 2021

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis


Zhi-Hao Lin Wei-Chiu Ma Hao-Yu Hsu Yu-Chiang Frank Wang Shenlong Wang

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

National Taiwan University 國立臺灣大學

Massachusetts Institute of Technology

Slide credit: Zhi-Hao Lin

https://zhihao-lin.github.io/neurmips/

Method:overview

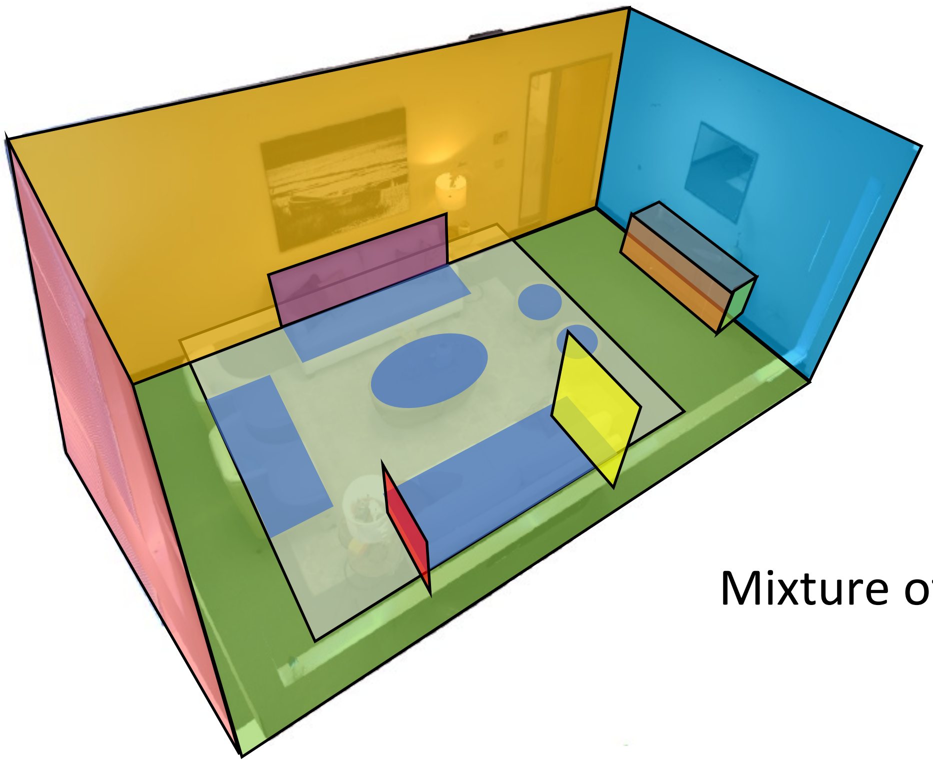
- Represent scene with mixture of local planar surfaces. (non-parallel)

Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

Method:overview

- Represent scene with mixture of local planar surfaces. (non-parallel)



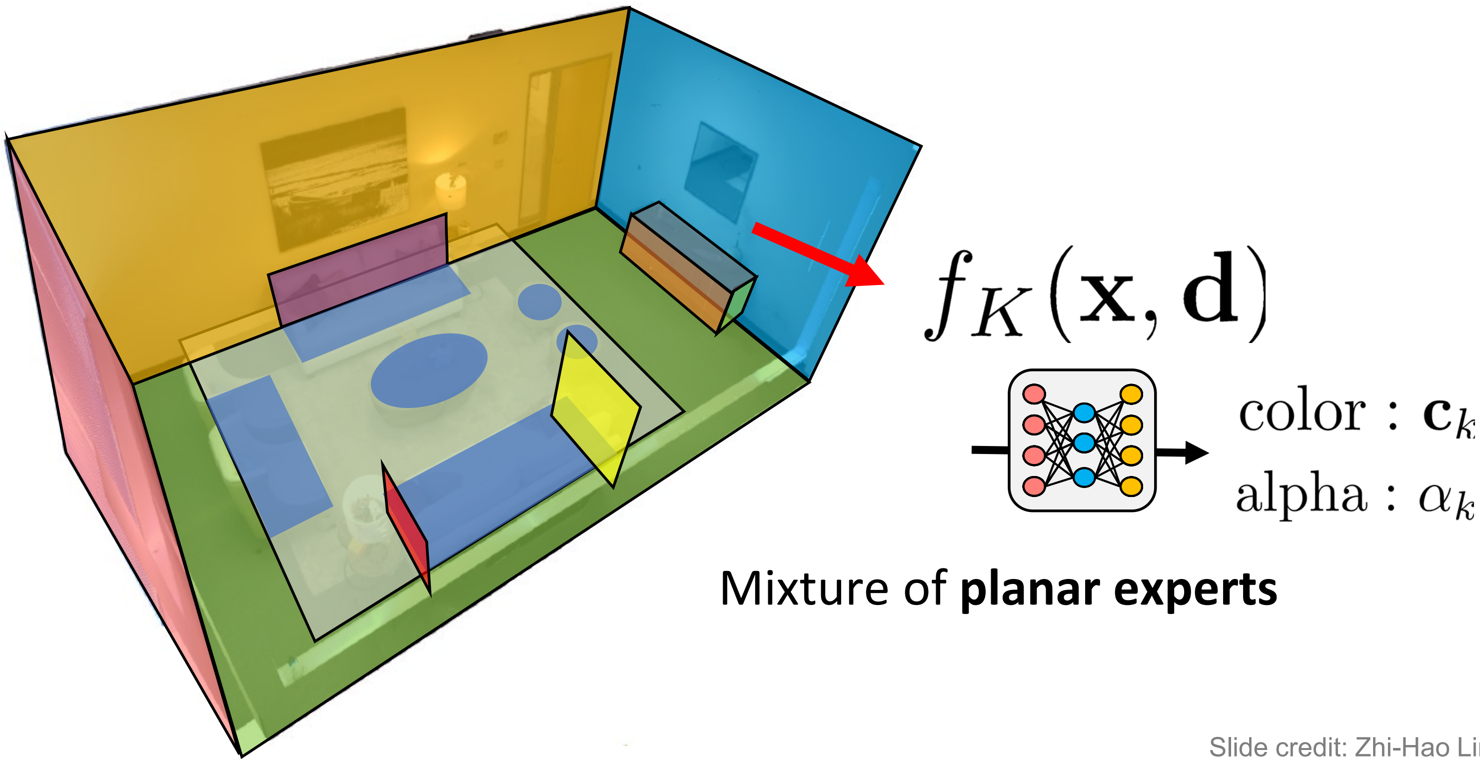
Mixture of **planar experts**

Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

Method:overview

- Represent scene with mixture of local planar surfaces. (non-parallel)

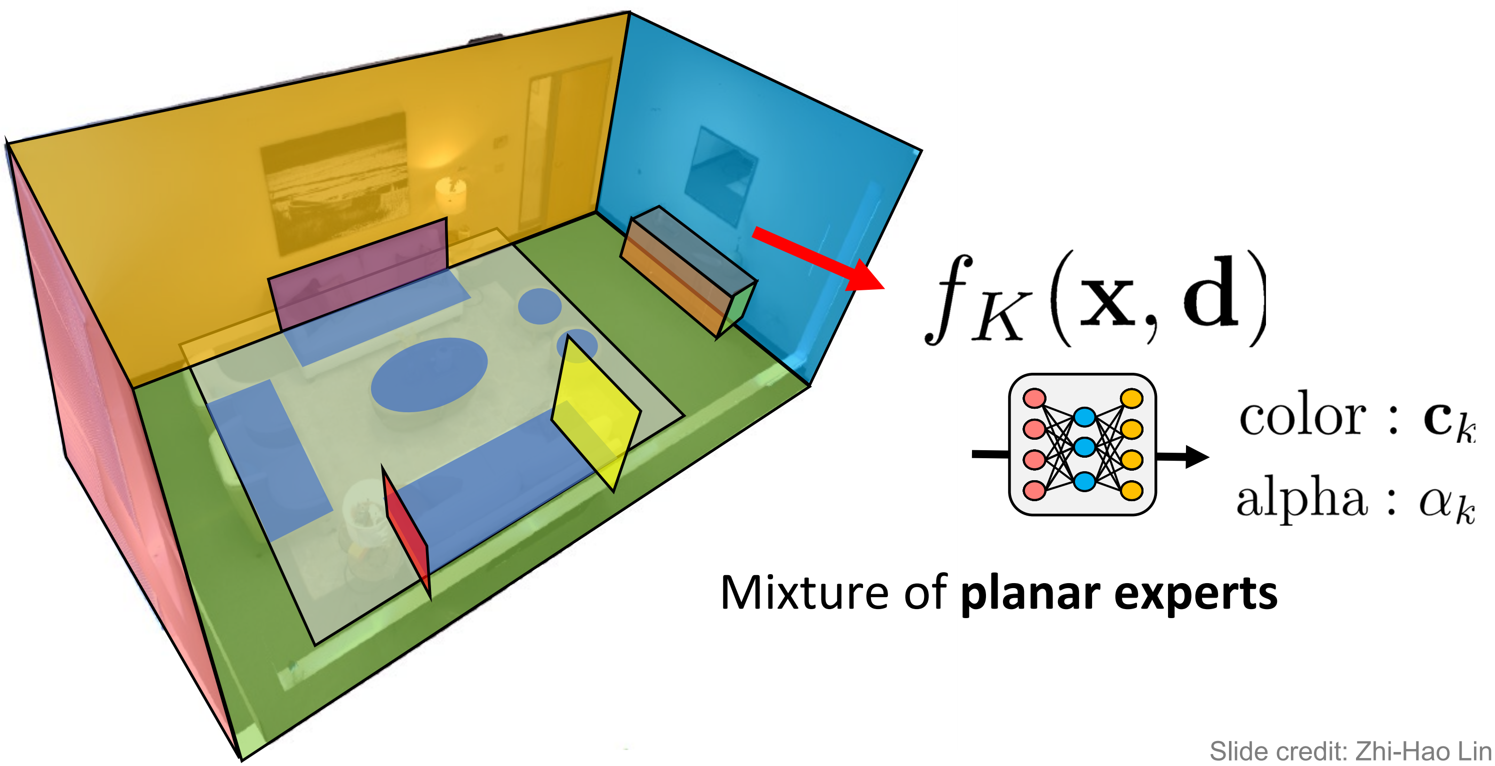


Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

Method: overview

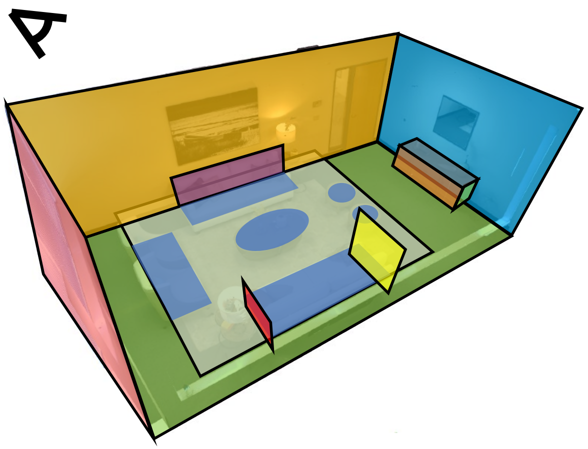
- Represent scene surface: avoid sampling in free space -> speed up
- Flexible plane geometry: allow NVS from wide-range view points



Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

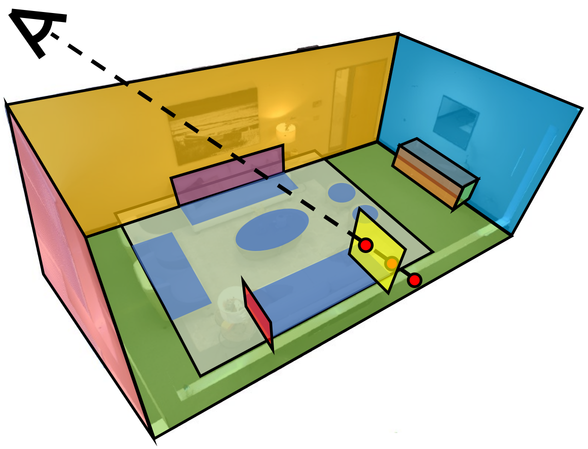
Input Ray and
Mixture of Planes



Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

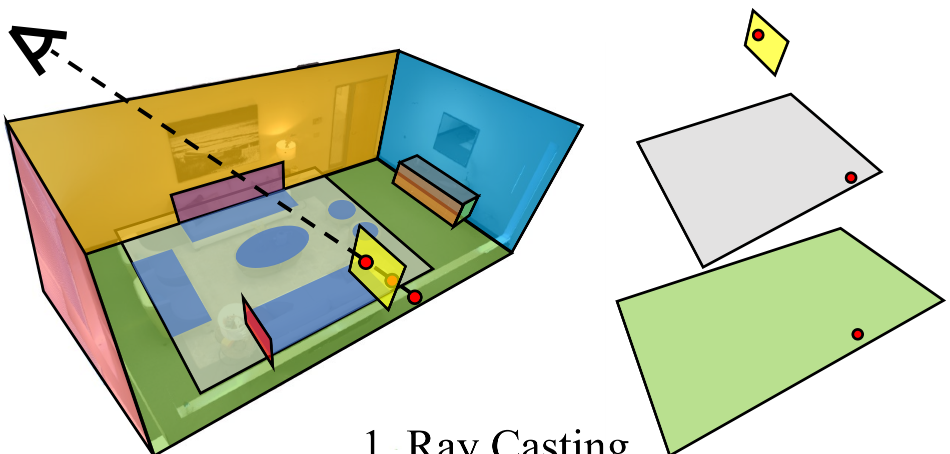
Input Ray and
Mixture of Planes



Slide credit: Zhi-Hao Lin

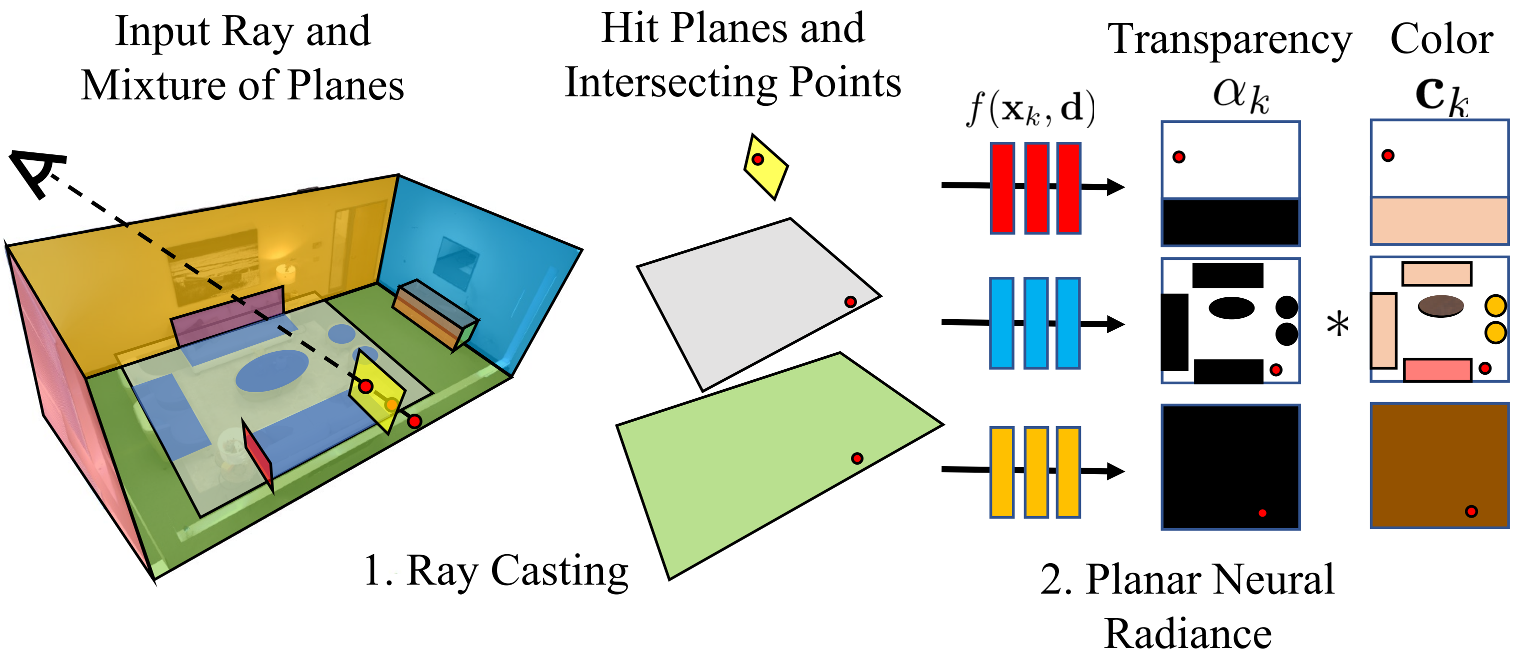
Input Ray and Mixture of Planes

Hit Planes and Intersecting Points

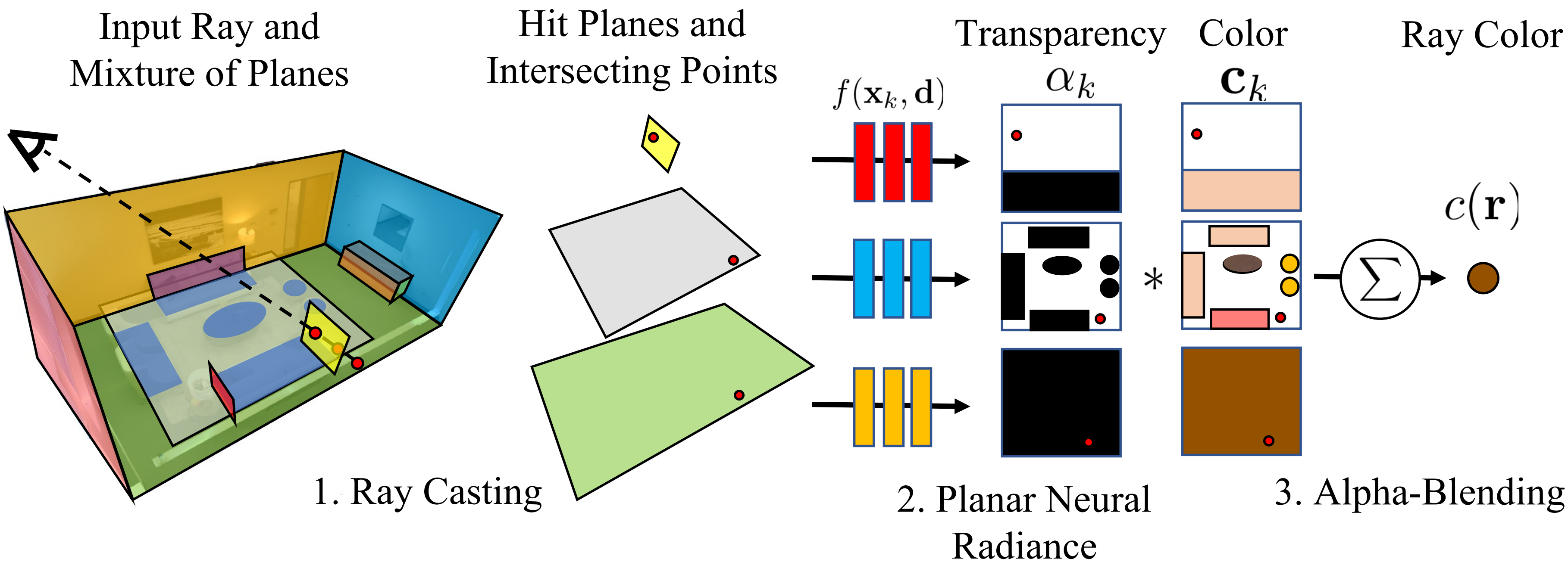


1. Ray Casting

Slide credit: Zhi-Hao Lin

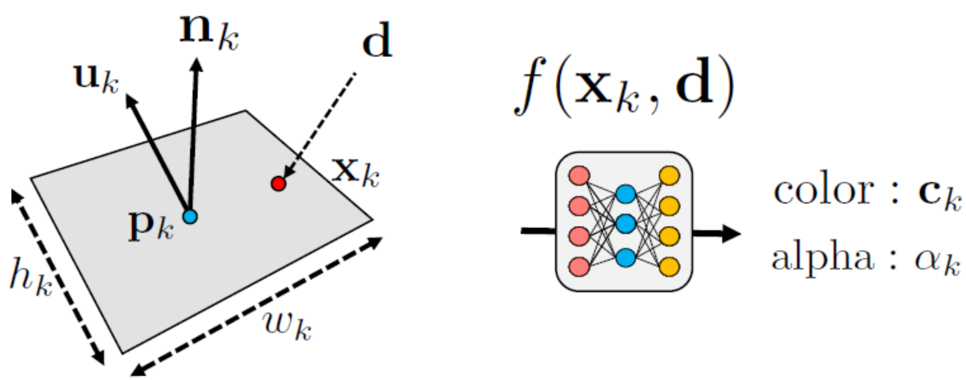


Slide credit: Zhi-Hao Lin



Slide credit: Zhi-Hao Lin

Method: model



- Position $\boldsymbol{p}_k \in R^3$
- orientation $\boldsymbol{n}_k, \boldsymbol{u}_k \in R^3$
- Size (w_k, h_k)

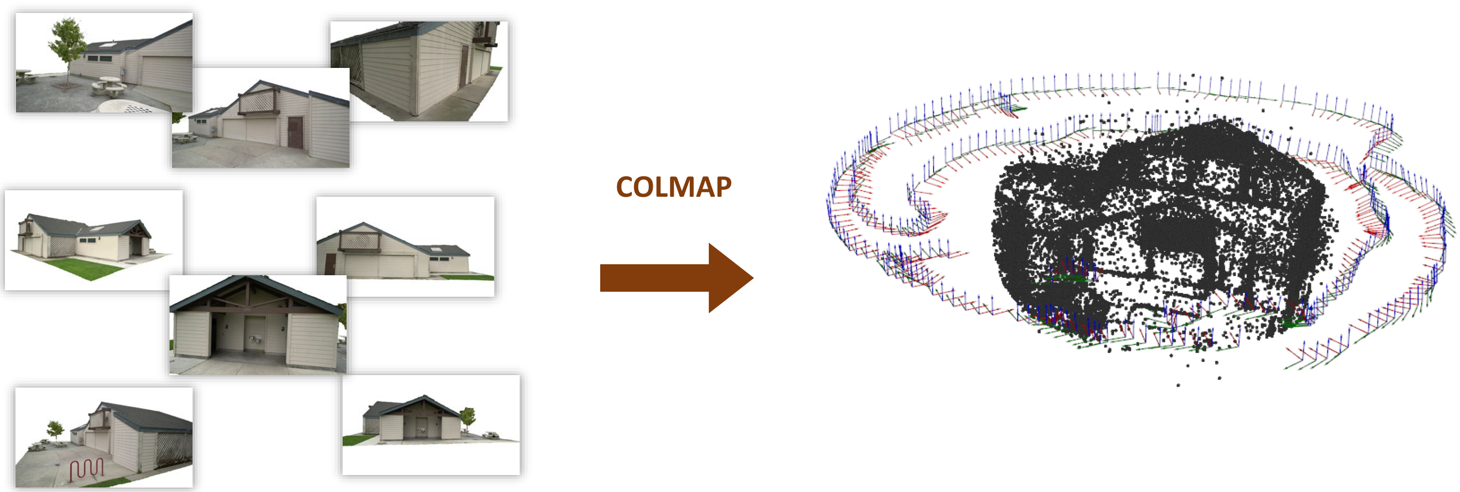
Neural radiance field network

- Input: position, direction
- Output: color, transparency

Slide credit: Zhi-Hao Lin

Method: initialization

- extract 3D point cloud from multiview images with COLMAP
- Initialize plane position, orientation on points



Multiview images

3D point cloud

Slide credit: Zhi-Hao Lin

Method: training

For training views, compute and optimize

- Geometry loss:

$$\mathcal{L}_g = \underbrace{\sum_i \min_k d(\mathbf{x}_i, \mathbf{s}_k)}_{\text{Point-rectangle distance}} + \lambda \underbrace{\sum_k (w_k h_k)^2}_{\text{Rectangle area}}$$

- Color loss:

$$\mathcal{L}_c = \frac{1}{B} \sum_{\mathbf{r}} \|c(\mathbf{r}) - c_{\text{gt}}(\mathbf{r})\|_2^2.$$

- Total:

$$\mathcal{L}_{total} = \mathcal{L}_g + \mathcal{L}_c$$

	NeX	NeRF	PlenOctree*	KiloNeRF*	Ours
# Params (M)	21.28	1.19	1457.2	6.21	3.11
FPS	0.142	0.106	78.04	4.19	19.16

Table 7. **Model size and inference speed on Replica.**

Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

Result

Slide credit: Zhi-Hao Lin

NeurMiPs: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022)

More references about NeRF improvements

- Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains (NeurIPS 2020) -> explain why positional encoding works
- PlenOctrees for Real-time Rendering of Neural Radiance Fields (ICCV 2021)
- Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields (ICCV 2021)
- KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs (ICCV 2021)
- Plenoxels : Radiance Fields without Neural Networks (CVPR 2022)

What to Cover Today?

- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
    - Extension of NeRF
    - Advanced Topics of NeRF

pixelNeRF: Neural Radiance Fields from One or Few Images

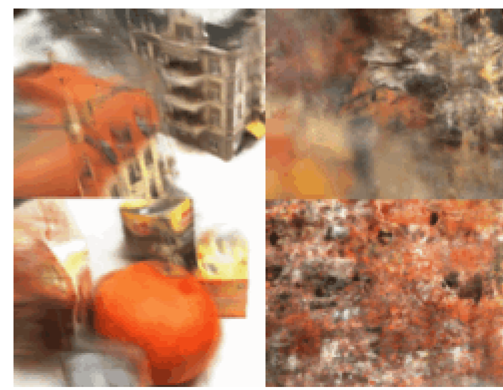
CVPR 2021

Alex Yu Vickie Ye Matthew Tancik Angjoo Kanazawa

UC Berkeley

Three view image

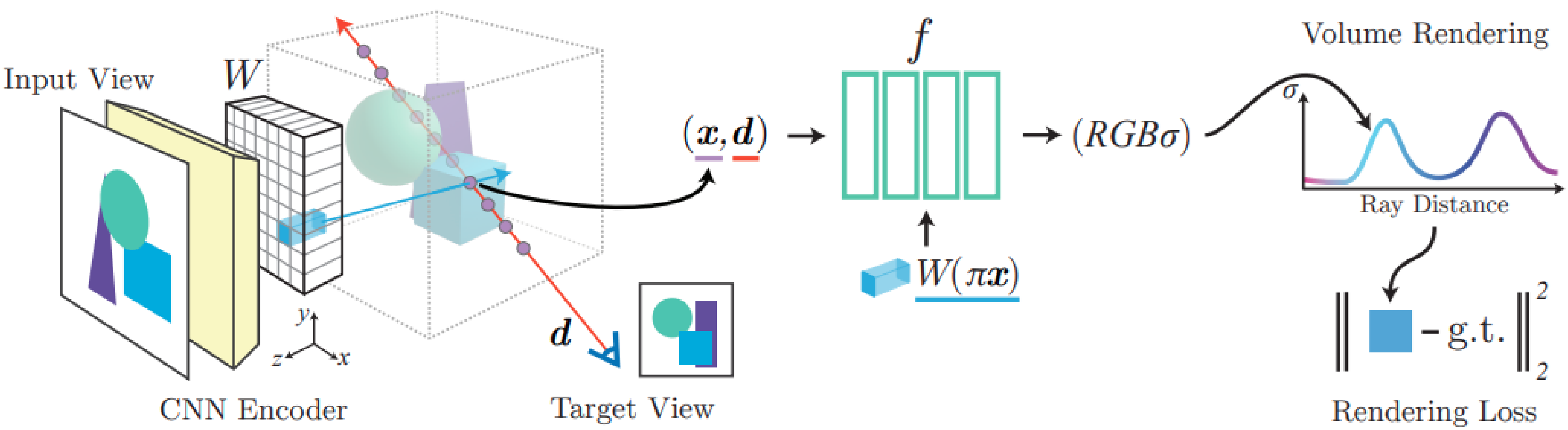
pixelNeRF NeRF



https://alexyu.net/pixelnerf/

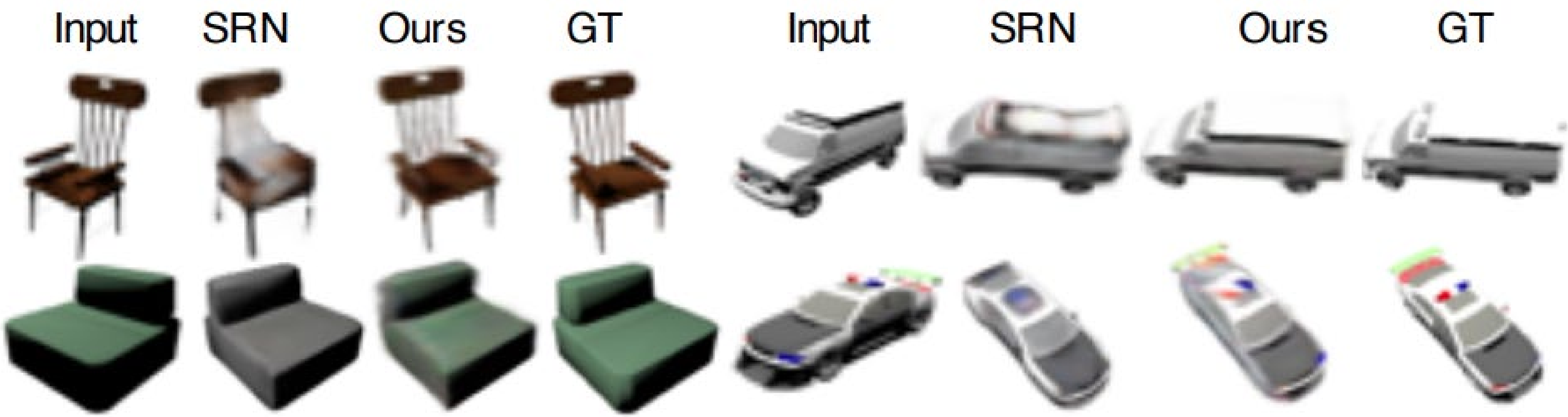
Method

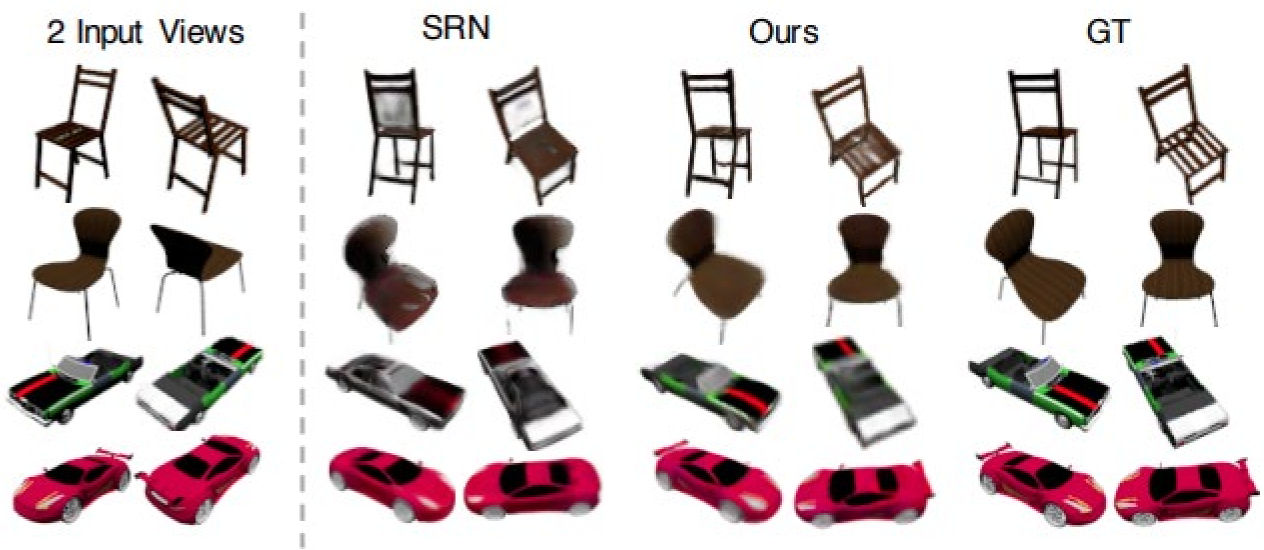
- Image feature as condition of NeRF
- The NeRF itself learns a object prior
  (e.g., what a general car/chair should look like)
- Able to fit different object/scene with only **one** NeRF model
- Only need one image (for encoding image feature) of the scene during testing on a new scene



pixelNeRF: Neural Radiance Fields from One or Few Images (CVPR 2021)

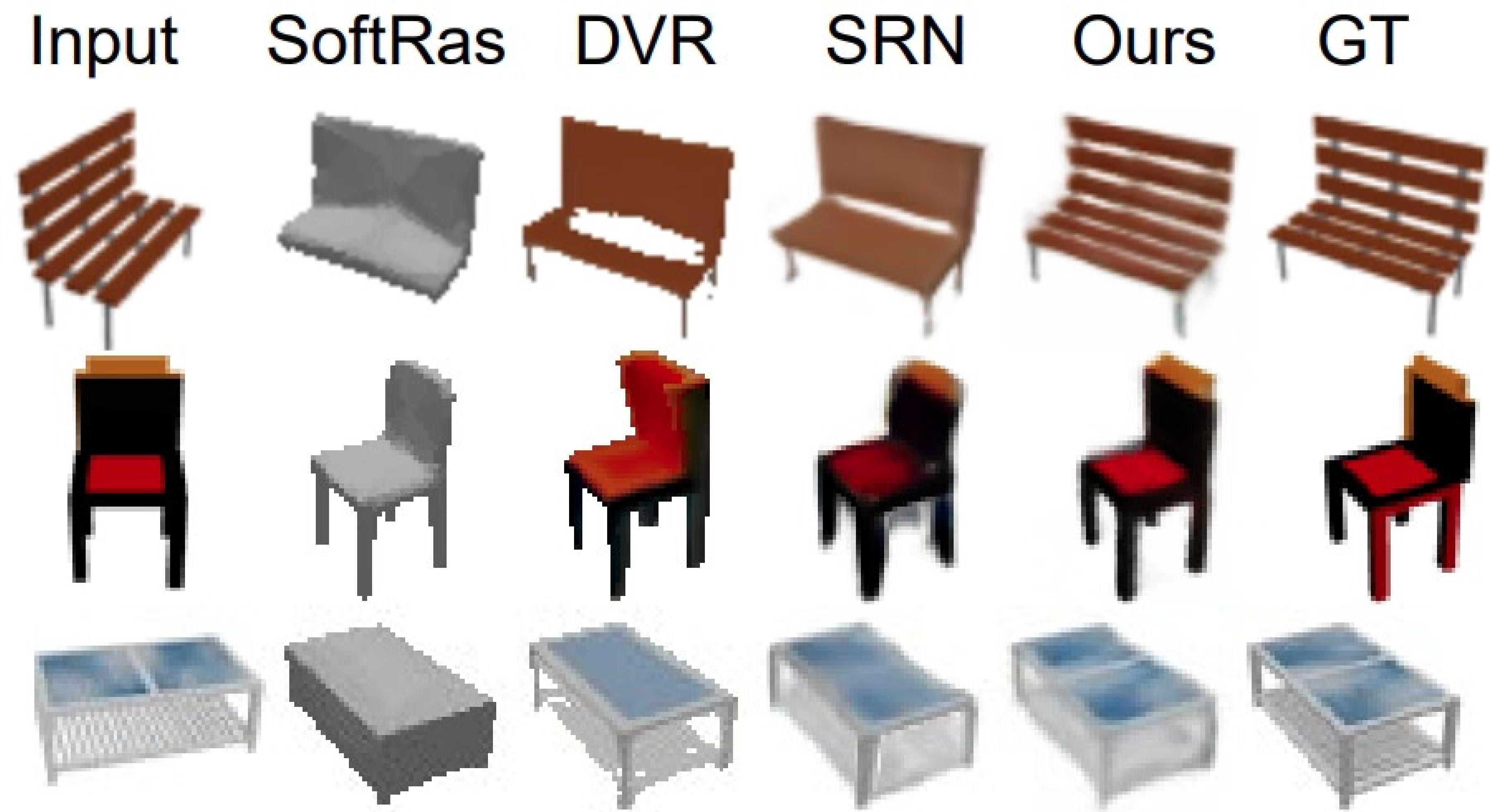
Results--single category





pixelNeRF: Neural Radiance Fields from One or Few Images (CVPR 2021)

Results--multi-category



pixelNeRF: Neural Radiance Fields from One or Few Images (CVPR 2021)

DREAMFUSION: TEXT-TO-3D USING 2D DIFFUSION

Ben Poole
Google Research

Ajay Jain
UC Berkeley

Jonathan T. Barron
Google Research

Ben Mildenhall
Google Research

Arxiv

https://dreamfusion3d.github.io/

Goal

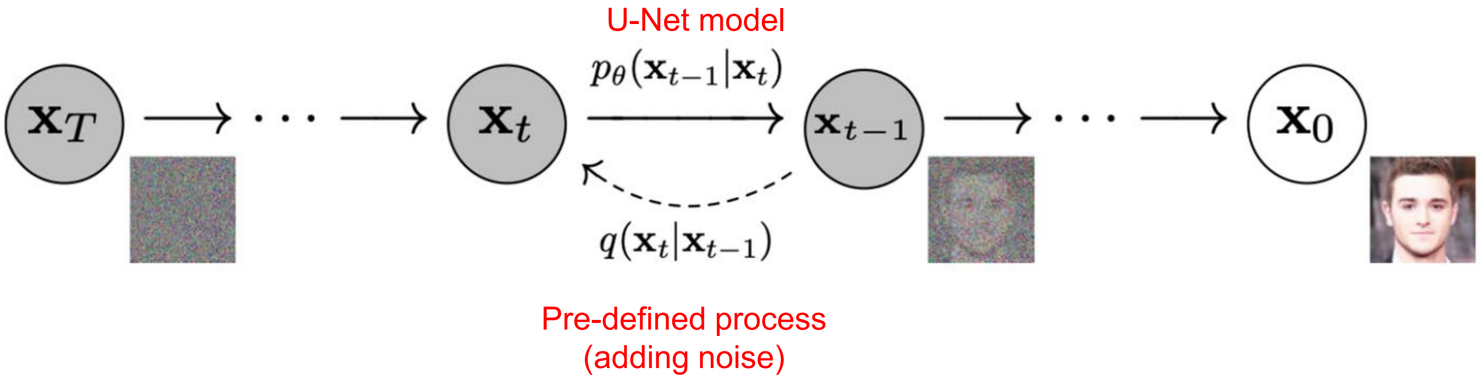
an orangutan making a clay bowl on a throwing wheel*

a raccoon astronaut holding his helmet†

a blue jay standing on a large basket of rainbow macarons*

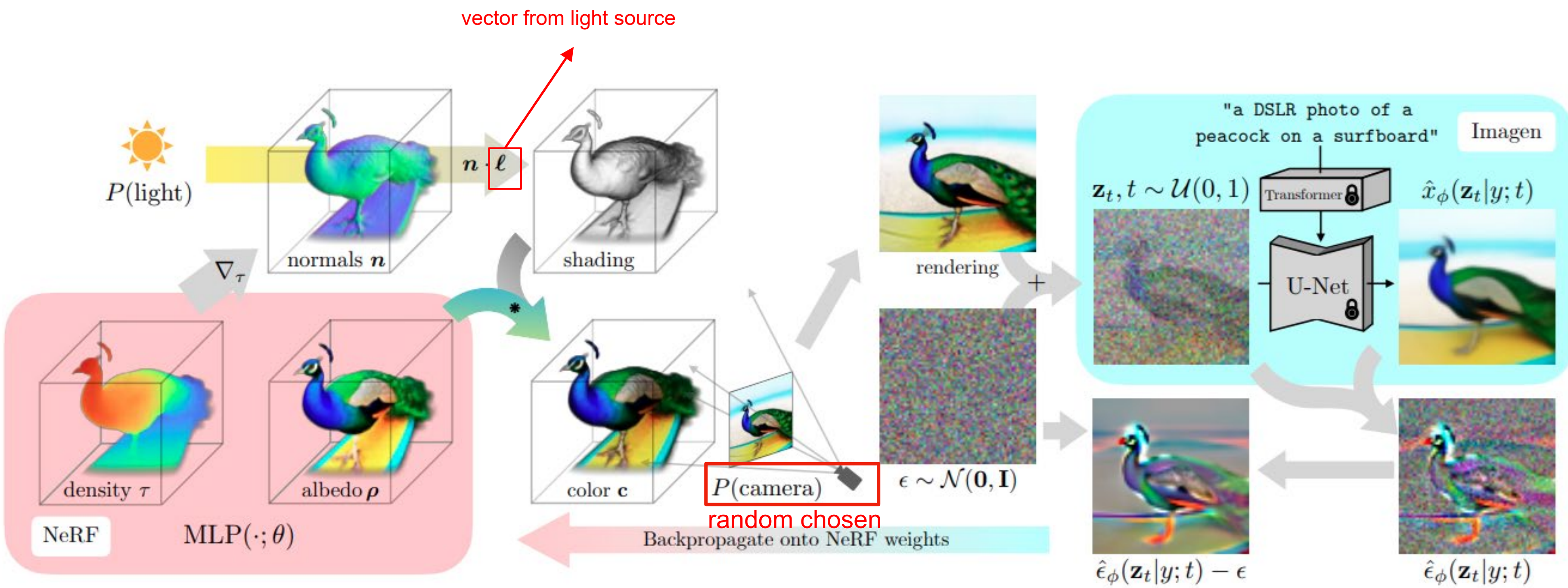
- Take description as input and generate corresponding 3D results (via 2D rendering)
- Without paired “text and 3D object”
- Combining NeRF and 2D text-to-image diffusion model

Recap: Diffusion model (intuitively)



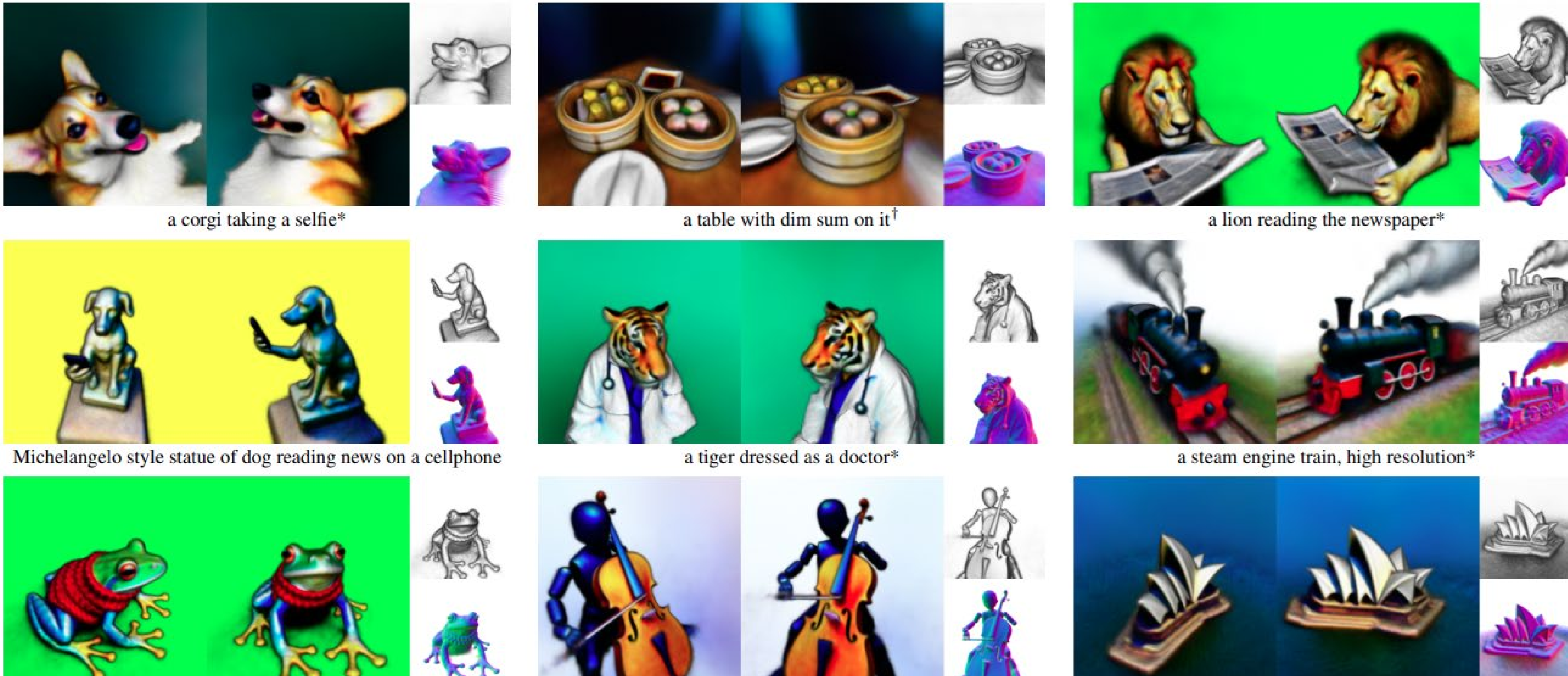
- Can be viewed as denoising from a Gaussian noise image
- Each step makes little progress of denoising (total about 1000 steps)
- Output image of each step can be seen as the **original image** combining with a **noise** using specific ratio
- The process can also be seen as predicting the **added noise**

Method



- The left part is a standard NeRF with shading condition
- Combine the rendered NeRF image with random noise to simulate a state of the text-to-image diffusion model
- The difference between the predicted noise and the inserted noise is treated as the rendering loss to guide NeRF

Result



a corgi taking a selfie*
a table with dim sum on it†
a lion reading the newspaper*
Michelangelo style statue of dog reading news on a cellphone
a tiger dressed as a doctor*
a steam engine train, high resolution*
a frog wearing a sweater*
a humanoid robot playing the cello*
Sydney opera house, aerial view†

Result

an all-utility vehicle driving across a stream†

a chimpanzee dressed like Henry VIII king of England*

a baby bunny sitting on top of a stack of pancakes†

a sliced loaf of fresh bread

a bulldozer clearing away a pile of snow*

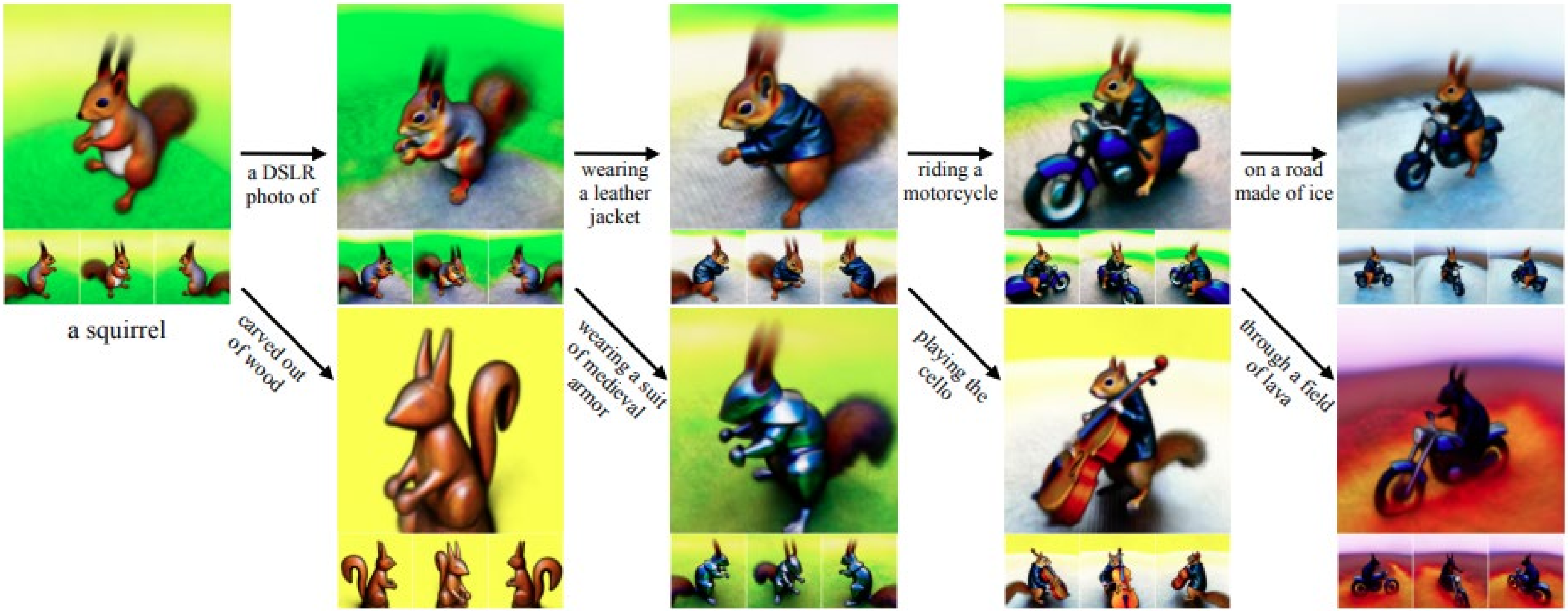
a classic Packard car*

zoomed out view of Tower Bridge made out of gingerbread and candy‡

a robot and dinosaur playing chess, high resolution*

a squirrel gesturing in front of an easel showing colorful pie charts

Result


a squirrel
a DSLR photo of
carved out of wood
wearing a leather jacket
wearing a suit of medieval armor
riding a motorcycle
playing the cello
on a road made of ice
through a field of lava

More references about further topics of NeRF

- Editing Conditional Radiance Fields (EditNeRF)(ICCV 2021)
- pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis (CVPR 2021)
- FENeRF: Face Editing in Neural Radiance Fields (CVPR 2022)
- StyleNeRF: A Style-based 3D-Aware Generator for High-resolution Image Synthesis (ICLR 2022)

What We’ve Covered Today?

- Introduction to 3D Vision
- Part I: 3D Perception
- Part II: 3D Reconstruction
- Neural Radiance Fields
  - Extension of NeRF
  - Advanced Topics of NeRF