SIGGRAPH 2021 Course on Advances in Neural Rendering

A. Tewari[1*], O. Fried[2*], J. Thies[3*], V. Sitzmann[4], S. Lombardi[5], Z. Xu[6], T. Simon[5], M. Nießner[7], E. Tretschk[8], L. Liu[8], B. Mildenhall[9], P. Srinivasan[9], R. Pandey[9], S. Orts-Escolano[9], S. Fanello[9], M. Guo[10], G. Wetzstein[10], J.-Y. Zhu[11], C. Theobalt[8], M. Agrawala[10], D. B Goldman[9], and M. Zollhöfer[5*]

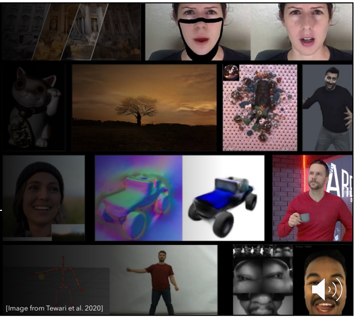
[1] *Max Planck Insitute for Informatics*
[2] *The Interdisciplinary Center Herzliya*
[3] *Max Planck Institute for Intelligent Systems*
[4] *Massachusetts Institute of Technology*
[5] *Facebook Reality Labs*
[6] *Adobe Research*
[7] *Technical University of Munich*
[8] *Max Planck Insitute for Informatics*
[9] *Google Inc*
[10] *Stanford University*
[11] *Carnegie Mellon University*
[*] *Organizers*

SIGGRAPH '21 Courses, August 09-13, 2021, Virtual Event, USA
ACM 978-1-4503-8361-5/21/08.
10.1145/3450508.3464573

Contents

Neural Rendering Course

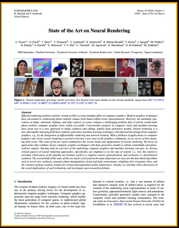
SIGGRAPH 2021



[Image from Tewari et al. 2020]

Michael Zollhoefer –
Facebook Reality Labs Research

Neural Rendering STAR [Tewari et al. 2020]



State of the Art on Neural Rendering

[Image from Tewari et al. 2020]

Michael Zollhoefer – Facebook Reality Labs Research

Program

Part 1

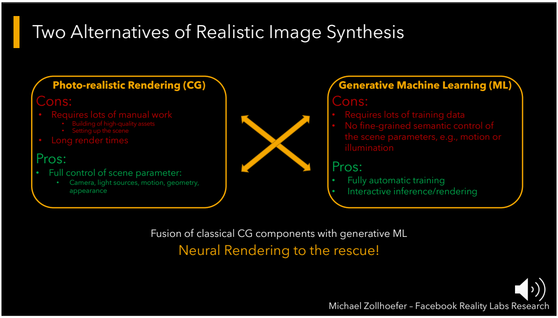
Time	Topic	Speakers
15 minutes	Introduction and Fundamentals	M. Zollhoefer
60 minutes	Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
80 minutes	Novel View Synthesis for Objects of Scenes	V. Sitzmann, D. B Goldman, B. Mildenhall, L. Liu
20 minutes	Learning to Relight	Z. Xu

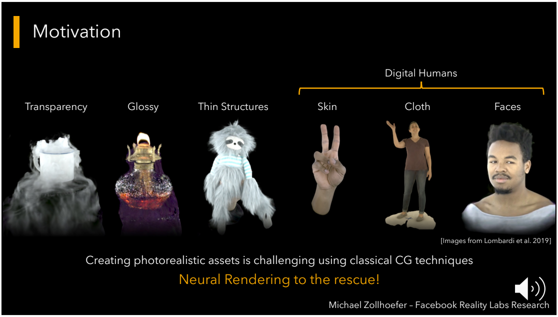
Part 2

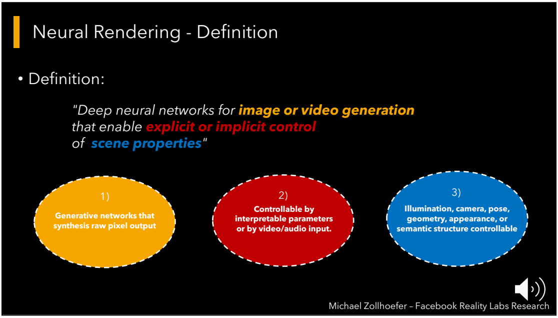
Time	Topic	Speakers
40 minutes	Learning to Relight	S. Orts-Escolano, P. Srinivasan
20 minutes	Compositional Scene Representations	M. Guo
60 minutes	Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
60 minutes	Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
10 minutes	Open Challenges and Conclusion	O. Fried
15 minutes	Discussion on Ethical Implications	M. Agrawala

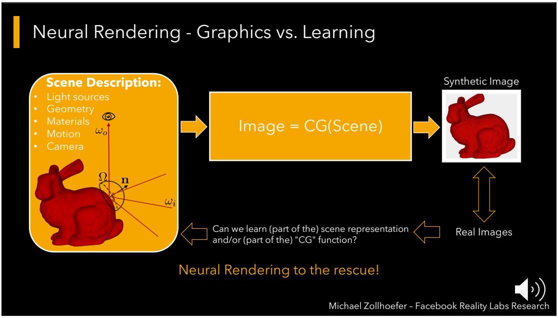
Part3 Panel Discussion & Questions (45minutes)

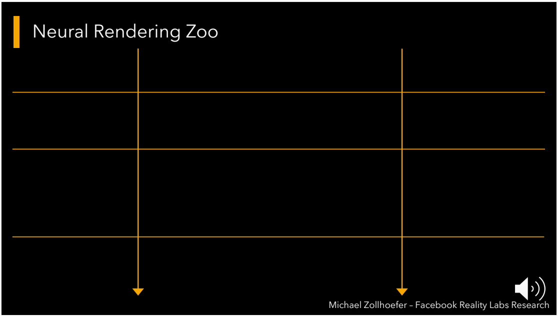
Michael Zollhoefer – Facebook Reality Labs Research

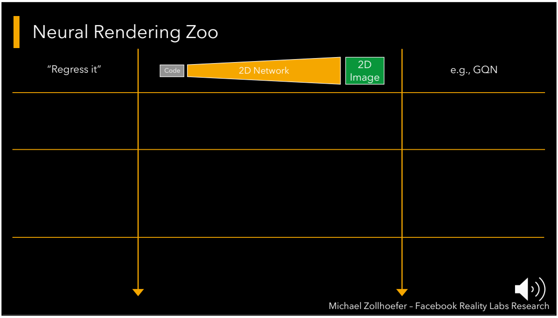

Two Alternatives of Realistic Image Synthesis
Photo-realistic Rendering (CG)
Cons:
Requires lots of manual work
Building of high-quality assets
Setting up the scene
Long render times
Pros:
Full control of scene parameter:
Camera, light sources, motion, geometry, appearance
Generative Machine Learning (ML)
Cons:
Requires lots of training data
No fine-grained semantic control of the scene parameters, e.g., motion or illumination
Pros:
Fully automatic training
Interactive inference/rendering
Fusion of classical CG components with generative ML
Neural Rendering to the rescue!
Michael Zollhoefer – Facebook Reality Labs Research

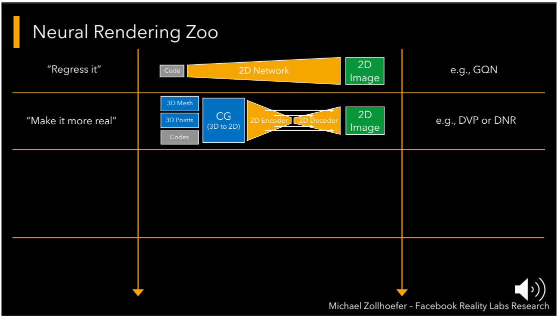

Motivation
Digital Humans
Transparency
Glossy
Thin Structures
Skin
Cloth
Faces
[Images from Lombardi et al. 2019]
Creating photorealistic assets is challenging using classical CG techniques
Neural Rendering to the rescue!
Michael Zollhoefer – Facebook Reality Labs Research

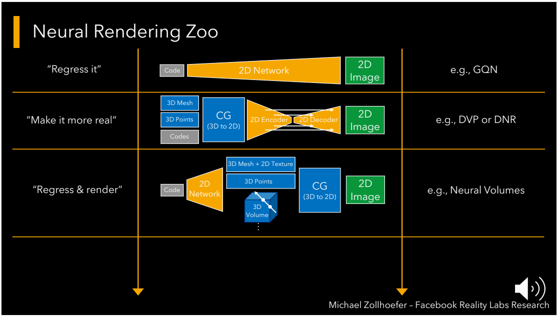

Neural Rendering - Definition
• Definition:
"Deep neural networks for image or video generation that enable explicit or implicit control of scene properties"
1) Generative networks that synthesis raw pixel output
2) Controllable by interpretable parameters or by video/audio input.
3) Illumination, camera, pose, geometry, appearance, or semantic structure controllable
Michael Zollhoefer – Facebook Reality Labs Research

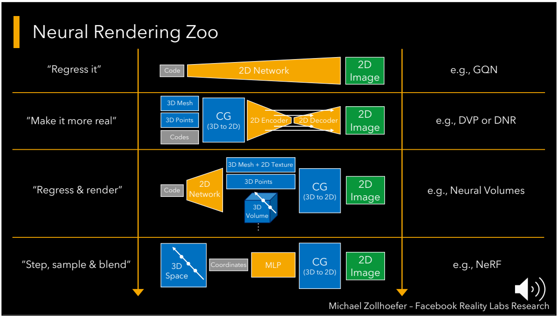

Neural Rendering - Graphics vs. Learning
Scene Description:
Light sources
Geometry
Materials
Motion
Camera
ω_o
Ω
n
ω_i
Image = CG(Scene)
Synthetic Image
Real Images
Can we learn (part of the) scene representation and/or (part of the) "CG" function?
Neural Rendering to the rescue!
Michael Zollhoefer – Facebook Reality Labs Research

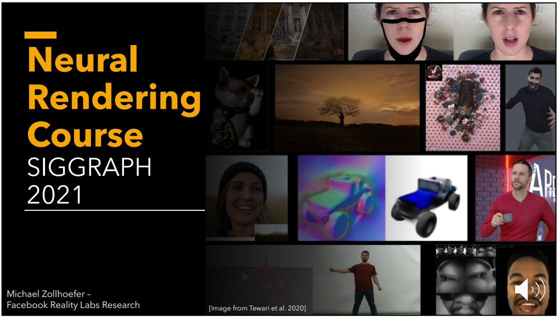

Neural Rendering Zoo
Michael Zollhoefer – Facebook Reality Labs Research


Neural Rendering Zoo
“Regress it”
Code
2D Network
2D Image
e.g., GQN
Michael Zollhoefer – Facebook Reality Labs Research


Neural Rendering Zoo
"Regress it"
Code
2D Network
2D Image
e.g., GQN
"Make it more real"
3D Mesh
3D Points
Codes
CG (3D to 2D)
2D Encoder
2D Decoder
2D Image
e.g., DVP or DNR
Michael Zollhoefer – Facebook Reality Labs Research


Neural Rendering Zoo
"Regress it"
Code
2D Network
2D Image
e.g., GQN
"Make it more real"
3D Mesh
3D Points
Codes
CG (3D to 2D)
2D Encoder
2D Decoder
2D Image
e.g., DVP or DNR
"Regress & render"
Code
2D Network
3D Mesh + 2D Texture
3D Points
3D Volume
CG (3D to 2D)
2D Image
e.g., Neural Volumes
Michael Zollhoefer – Facebook Reality Labs Research

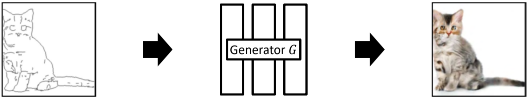

Neural Rendering Zoo
"Regress it"
Code
2D Network
2D Image
e.g., GQN
"Make it more real"
3D Mesh
3D Points
Codes
CG (3D to 2D)
2D Encoder
2D Decoder
2D Image
e.g., DVP or DNR
"Regress & render"
Code
2D Network
3D Mesh + 2D Texture
3D Points
3D Volume
CG (3D to 2D)
2D Image
e.g., Neural Volumes
"Step, sample & blend"
3D Space
Coordinates
MLP
CG (3D to 2D)
2D Image
e.g., NeRF
Michael Zollhoefer – Facebook Reality Labs Research


Neural
Rendering
Course
SIGGRAPH
2021
Michael Zollhoefer –
Facebook Reality Labs Research
[Image from Tewari et al. 2020]

Loss functions for Neural Rendering

Jun-Yan Zhu
Assistant Professor, CMU School of Computer Science
Carnegie Mellon University

Loss functions for Neural Rendering



Input x — Learnable rendering — Output Image G(x)

What is a good objective $\mathcal{L}$?
- Capture realism.
- Enforce correspondence.
- Useful for many tasks.
- Adapt to new tasks/data.

Problem Statement

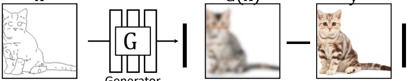
Loss function

$$\arg\min_G \mathcal{L}(G(x), y)$$

Generator (CNN, MLP, Transformer, Differentiable rendering) Input Info Output image

Designing Loss Functions

x G(x) y

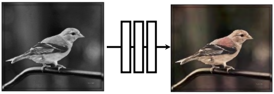


Input Generator Predicted output GT output

L2 regression $$\arg\min_G \mathbb{E}[\,||G(x) - y||\,]$$

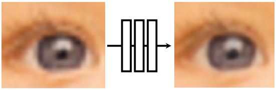
Designing Loss Functions

Image colorization



L2 regression

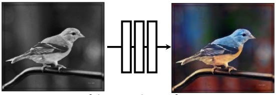
Super-resolution



L2 regression

Designing Loss Functions

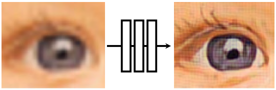
Image colorization



[Zhang et al., 2016]

Classification Loss:
Cross entropy objective, with colorfulness term

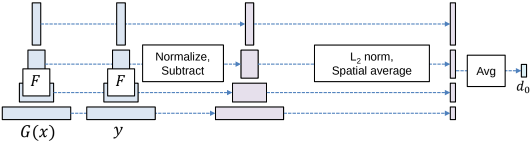
Super-resolution



Feature/Perceptual loss
Deep feature space matching objective

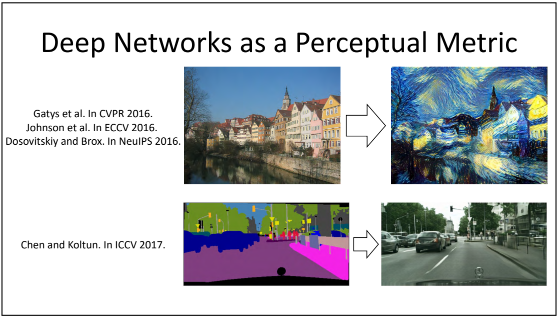
[Gatys et al., 2016], [Johnson et al., 2016], [Dosovitskiy and Brox, 2016]

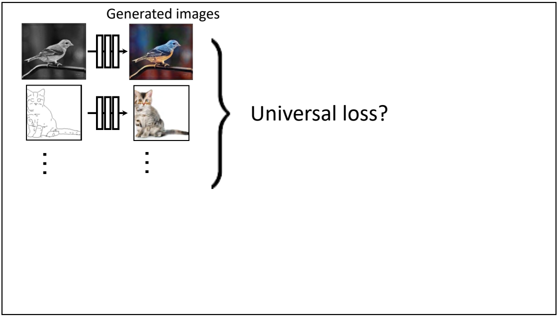
Deep Networks as a Perceptual Metric

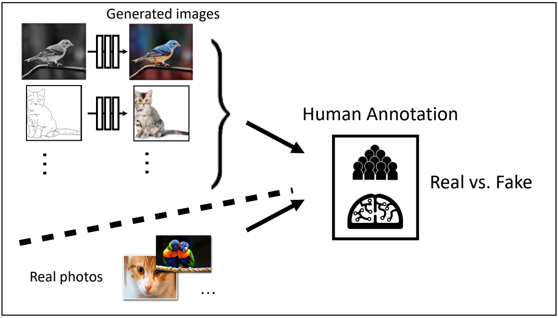


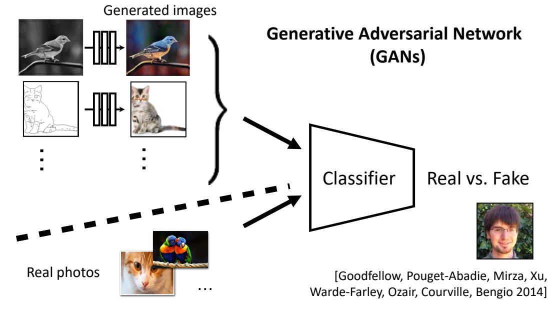
How well do “perceptual losses” describe perception?
75% agreement with Human Perception of Patch Similarity [Zhang et al., 2018]

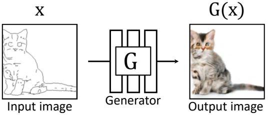
c.f. Gatys et al. CVPR 2016. Johnson et al. ECCV 2016. Dosovitskiy and Brox. NeurIPS 2016.


Deep Networks as a Perceptual Metric
Gatys et al. In CVPR 2016.
Johnson et al. In ECCV 2016.
Dosovitskiy and Brox. In NeuIPS 2016.
Chen and Koltun. In ICCV 2017.

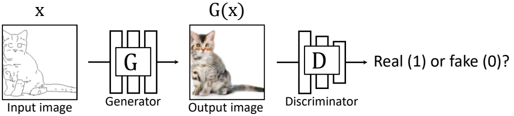

Generated images
Universal loss?


Generated images
Human Annotation
Real vs. Fake
Real photos
...





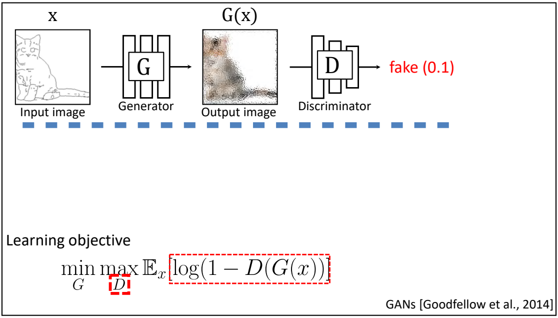
GANs [Goodfellow et al., 2014]

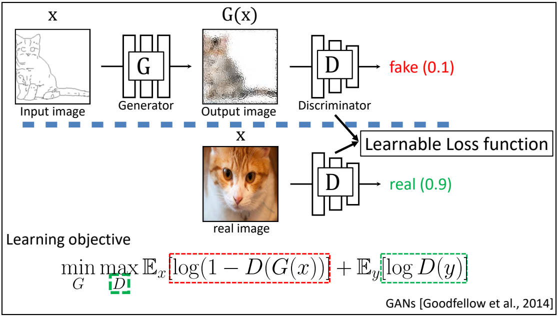


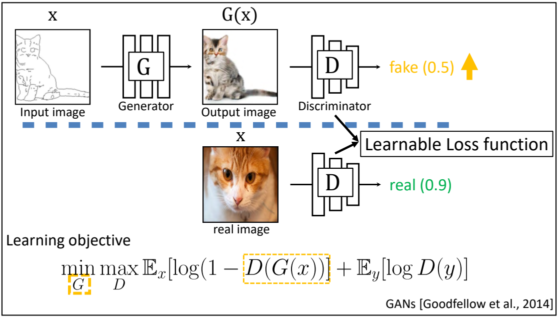
A two-player game:

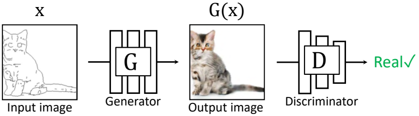
- G tries to generate fake images that can fool D.
- D tries to detect fake images.

GANs [Goodfellow et al., 2014]


x
G(x)
G
D
fake (0.1)
Input image
Generator
Output image
Discriminator
Learning objective
$\min_G \max_D \mathbb{E}_x[\log(1 - D(G(x)))]$
GANs [Goodfellow et al., 2014]


x
G(x)
G
D
fake (0.1)
Input image
Generator
Output image
Discriminator
Learnable Loss function
x
D
real (0.9)
real image
Learning objective
$\min_G \max_D \mathbb{E}_x[\log(1 - D(G(x)))] + \mathbb{E}_y[\log D(y)]$
GANs [Goodfellow et al., 2014]

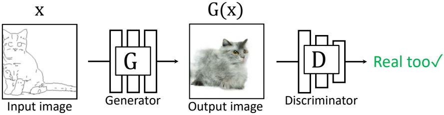

x
G(x)
G
D
fake (0.5)
Input image
Generator
Output image
Discriminator
Learnable Loss function
x
D
real (0.9)
real image
Learning objective
$\min_G \max_D \mathbb{E}_x[\log(1 - D(G(x)))] + \mathbb{E}_y[\log D(y)]$
GANs [Goodfellow et al., 2014]



Learning objective

$$\min_G \max_D \mathbb{E}_x[\log(1 - D(G(x))] + \mathbb{E}_y[\log D(y)]$$

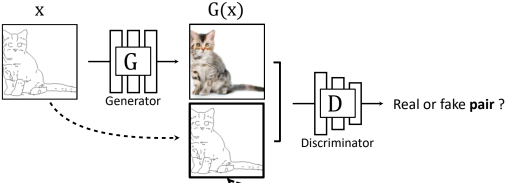
pix2pix [Isola, Zhu, Zhou, Efros, 2016]



Learning objective

$$\min_G \max_D \mathbb{E}_x[\log(1 - D(G(x))] + \mathbb{E}_y[\log D(y)]$$

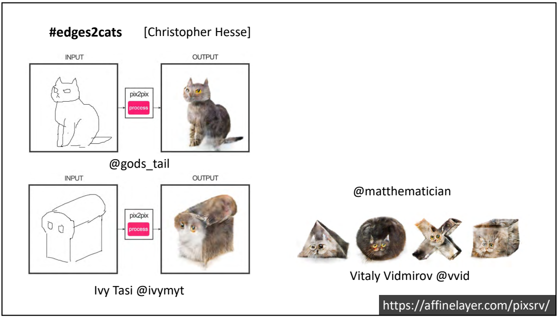
pix2pix [Isola, Zhu, Zhou, Efros, 2016]

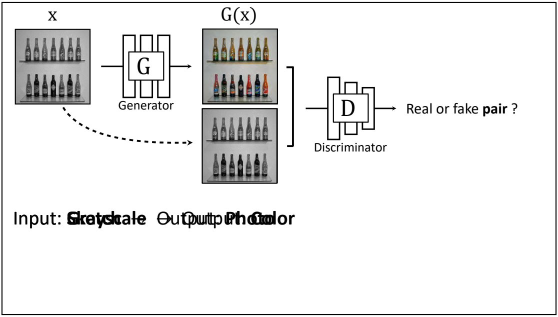


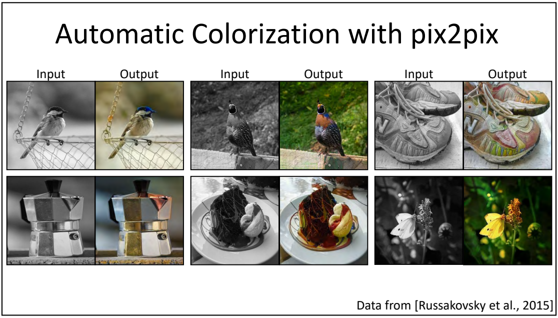
Learning objective

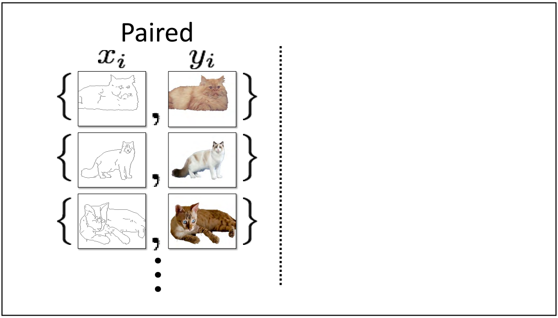
$$\min_G \max_D \mathbb{E}_x[\log(1 - D(\boxed{x}, G(x))] + \mathbb{E}_{x,y}[\log D(\boxed{x}, y)]$$

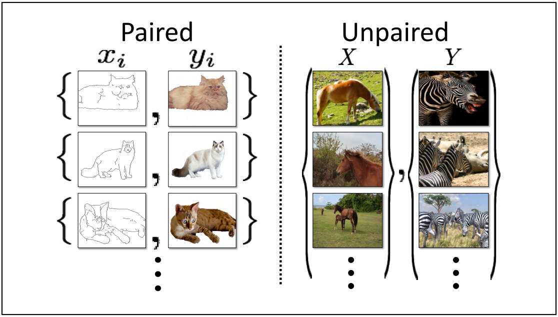
pix2pix [Isola, Zhu, Zhou, Efros, 2016]

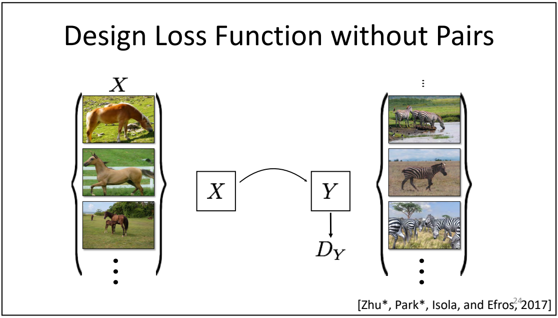

#edges2cats
[Christopher Hesse]
INPUT
OUTPUT
pix2pix
process
@gods_tail
INPUT
OUTPUT
pix2pix
process
Ivy Tasi @ivymyt
@matthematician
Vitaly Vidmirov @vvid
https://affinelayer.com/pixsrv/

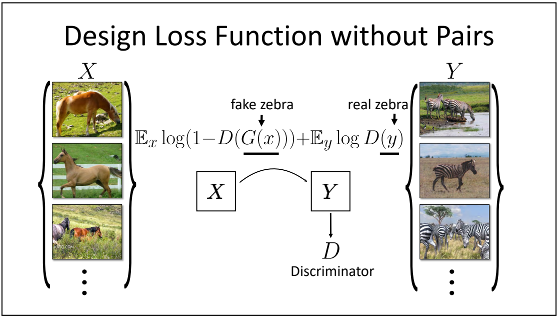

x
G(x)
G
Generator
D
Discriminator
Real or fake pair ?
Input: Grayscale Output: Color

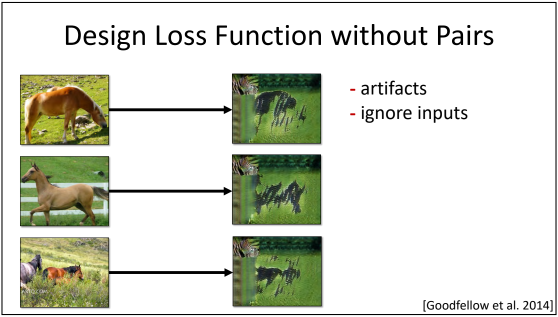

Automatic Colorization with pix2pix
Input
Output
Input
Output
Input
Output
Data from [Russakovsky et al., 2015]

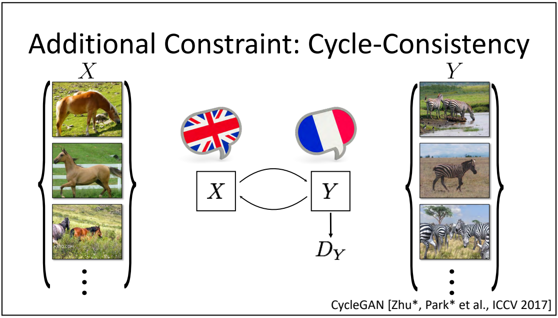

Paired
x_i
y_i


Paired
x_i
y_i
Unpaired
X
Y

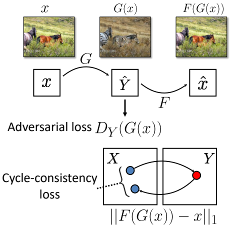

Design Loss Function without Pairs
X
X
Y
D_Y
[Zhu*, Park*, Isola, and Efros, 2017]


Design Loss Function without Pairs
X
Y
fake zebra
real zebra
$\mathbb{E}_x \log(1-D(G(x)))+\mathbb{E}_y \log D(y)$
X
Y
D
Discriminator


Design Loss Function without Pairs
- artifacts
- ignore inputs
[Goodfellow et al. 2014]


Additional Constraint: Cycle-Consistency
X
Y
X
Y
D_Y
CycleGAN [Zhu*, Park* et al., ICCV 2017]

Cycle-Consistent Adversarial Networks



Adversarial loss

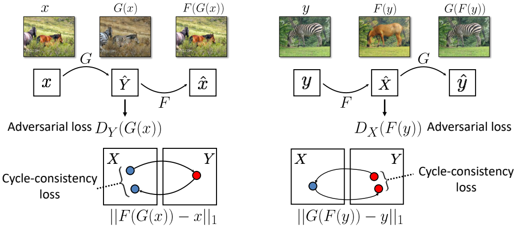
$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$

Cycle-consistency loss

$$\mathbb{E}_x ||F(G(x)) - x||_1$$

CycleGAN [Zhu*, Park* et al., ICCV 2017]

Cycle-Consistent Adversarial Networks



CycleGAN [Zhu*, Park* et al., ICCV 2017]

Results

Horse → Zebra

Orange → Apple

Customizing Gaming Experience

Grand Theft Auto v (GTA5)

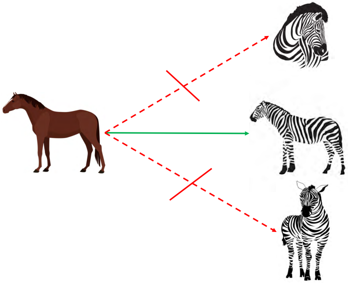
Street view images in German cities

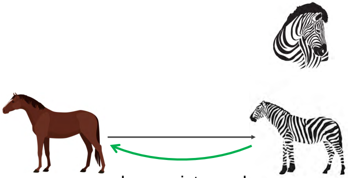
Data from [Richter et al., 2016], [Cordts et al, 2016]

Customizing Gaming Experience

Output image with Germany street view style

Input GTA5 CG

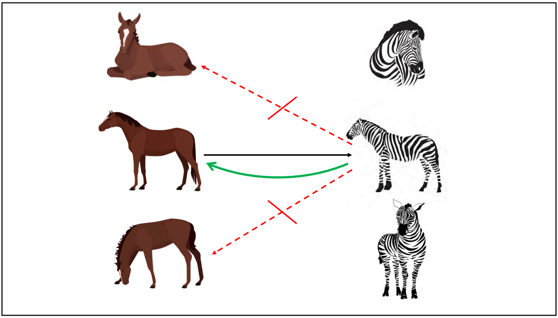


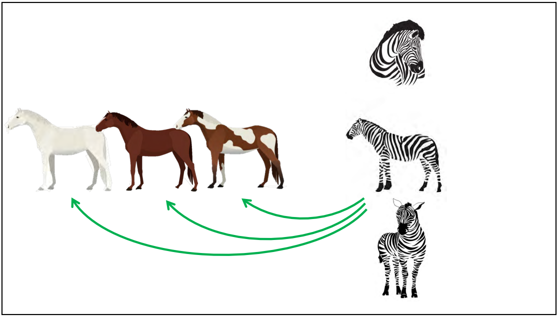


cycle-consistency loss

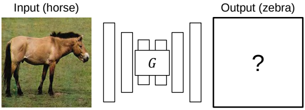
CycleGAN (Zhu et al., ICCV'17)
DiscoGAN (Kim et al., ICML'17)
DualGAN (Yi et al., ICCV'17)

Also used in
MUNIT (Huang et al., ECCV'18)
DRIT (Lee et al., ECCV'18)

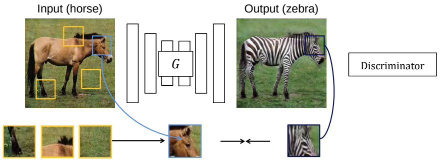




What makes for a good output?

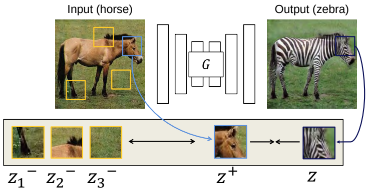


Retaining input **content**



CUT [Park, Efros, Zhang, Zhu, 2017]

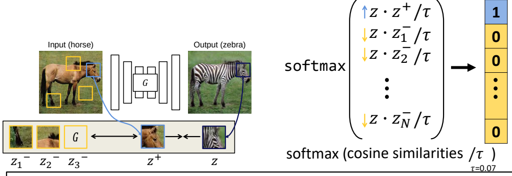
Retaining input **content**



Corresponding patches should have **high similarity**

CUT [Park, Efros, Zhang, Zhu, 2017]

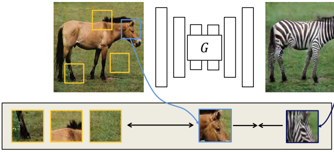
Patch-based Contrastive Loss



- InfoNCE loss (Gutmann et al., AISTATS18 , van den Oord et al., 2018) used in MoCo and SimCLR
- To produce positive pairs:
  - Handcrafted data augmentation (MoCo, SimCLR, etc.)
  - Input and synthesized image (ours)

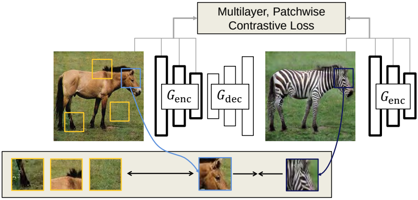
MoCo: He et al., CVPR20, SimCLR: Chen et al., ICML20

Patchwise contrastive loss



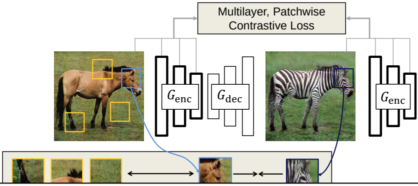
CUT [Park, Efros, Zhang, Zhu, 2017]

Patchwise contrastive loss



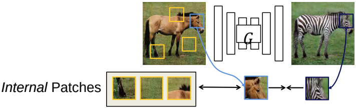
CUT [Park, Efros, Zhang, Zhu, 2017]

Patchwise contrastive loss

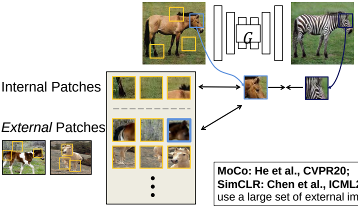


+ No fixed similarity metric (e.g., L1 or perceptual loss)
+ One-sided (no inverse mapping needed)

Internal vs External Patches



Internal vs External Patches



MoCo: He et al., CVPR20;
SimCLR: Chen et al., ICML20
use a large set of external images as negative samples

External patches make things worse

Cat

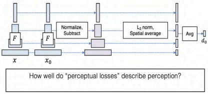
Yosemite Summer

Apple

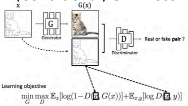
Paris

GTA

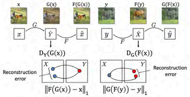
Perceptual/Feature Loss



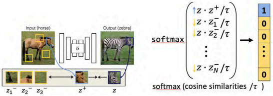
(Conditional) GAN Loss



Cycle-Consistency Loss



Patch-wise Contrastive Loss



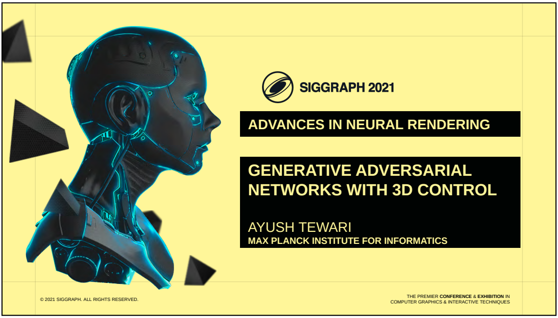
Other loss functions: Style Loss [Gatys et al.], Contextual Loss [Mechrez et al.], Domain-specific Loss (e.g., face), 3D-aware Loss (for geometric data)

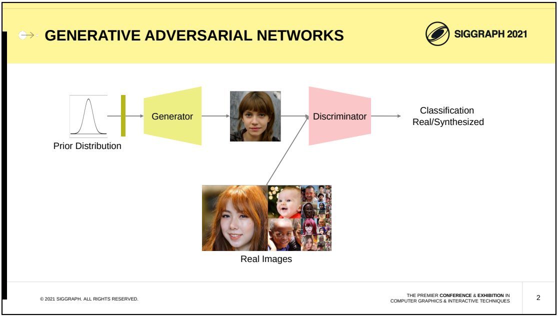
Collaborators

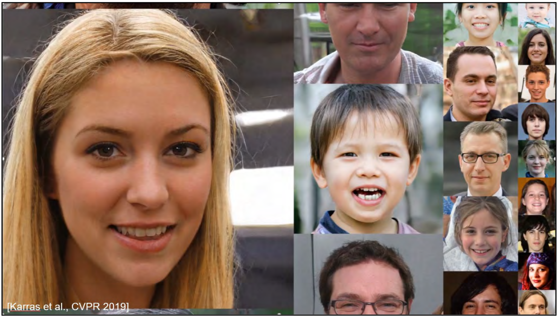
Thank You!

Code and models: https://github.com/junyanz/

See more details at CMU course "Learning-based Image Synthesis"


SIGGRAPH 2021
ADVANCES IN NEURAL RENDERING
GENERATIVE ADVERSARIAL NETWORKS WITH 3D CONTROL
AYUSH TEWARI
MAX PLANCK INSTITUTE FOR INFORMATICS
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


GENERATIVE ADVERSARIAL NETWORKS
SIGGRAPH 2021
Prior Distribution
Generator
Discriminator
Classification
Real/Synthesized
Real Images
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
2


[Karras et al., CVPR 2019]

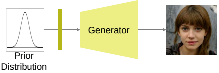
GENERATIVE ADVERSARIAL NETWORKS

SIGGRAPH 2021

[Brock et al., ICLR 2019]

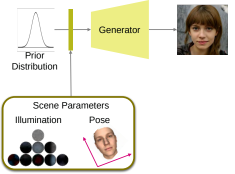
NEURAL RENDERING

SIGGRAPH 2021



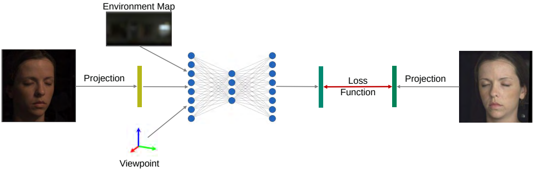
NEURAL RENDERING

SIGGRAPH 2021



SUPERVISED TRAINING

SIGGRAPH 2021



PhotoApp: Photorealistic appearance editing of head portraits [B R et al., SIGGRAPH 2021]

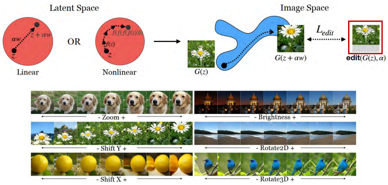
SUPERVISED TRAINING

SIGGRAPH 2021

PhotoApp: Photorealistic appearance editing of head portraits [B R et al., SIGGRAPH 2021]

TRAINING ON SYNTHETIC DATASETS

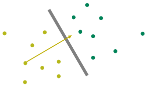
SIGGRAPH 2021

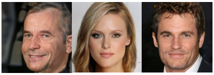


On the "Steerability" of Generative Adversarial Networks. [Jahanian et al., ICLR 2020]

TRAINING WITHOUT SUPERVISED PAIRS

SIGGRAPH 2021

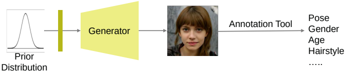




Interpreting the Latent Space of GANs for Semantic Face Editing. [Shen et al., CVPR 2020]

TRAINING WITHOUT SUPERVISED PAIRS

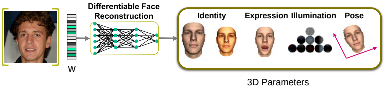
SIGGRAPH 2021

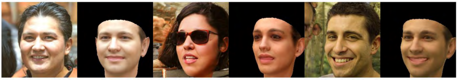


StyleFlow: Attribute-conditioned Exploration of StyleGAN-generated Images using Conditional Continuous Normalizing Flows. [Abdal et al., TOG 2021]

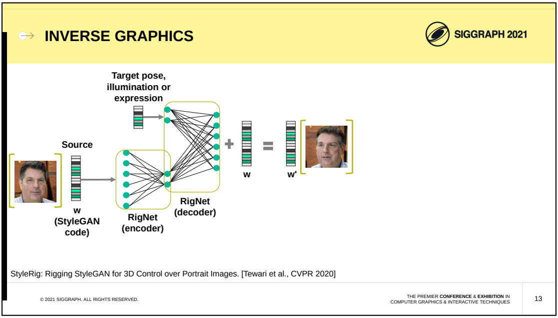
INVERSE GRAPHICS

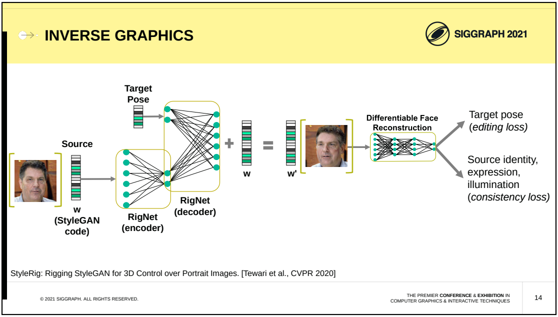
SIGGRAPH 2021





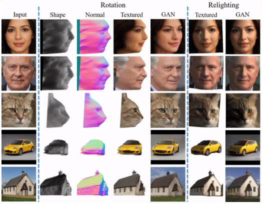
StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. [Tewari et al., CVPR 2020]


INVERSE GRAPHICS
SIGGRAPH 2021
Target pose, illumination or expression
Source
w (StyleGAN code)
RigNet (encoder)
RigNet (decoder)
+
w
=
w'
StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. [Tewari et al., CVPR 2020]
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
13


INVERSE GRAPHICS
SIGGRAPH 2021
Target Pose
Source
w (StyleGAN code)
RigNet (encoder)
RigNet (decoder)
+
w
=
w'
Differentiable Face Reconstruction
Target pose (editing loss)
Source identity, expression, illumination (consistency loss)
StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. [Tewari et al., CVPR 2020]
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
14

INVERSE GRAPHICS

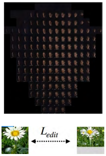
SIGGRAPH 2021

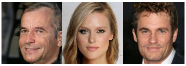


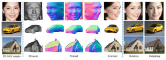
Do 2D GANs Know 3D Shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs. [Pan et al., ICLR 2021]

DIFFERENT AMOUNTS OF SUPERVISION

SIGGRAPH 2021







[B R et al., 2021]

[Jahanian et al., 2020]

[Shen et al., 2020]

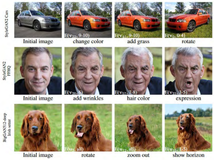
[Abdal et al., 2021]

[Tewari et al., 2020]

[Pan et al., 2021]

UNSUPERVISED METHODS

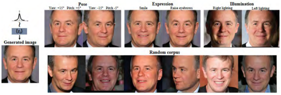
SIGGRAPH 2021



GANSpace: Discovering Interpretable GAN Controls. [Härkönen et al., NeurIPS 2020]

TRAINING THE GENERATOR

SIGGRAPH 2021

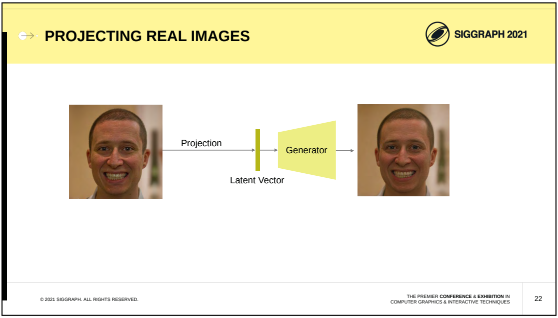


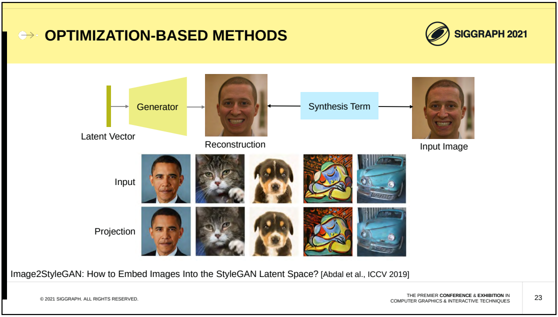
Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning.

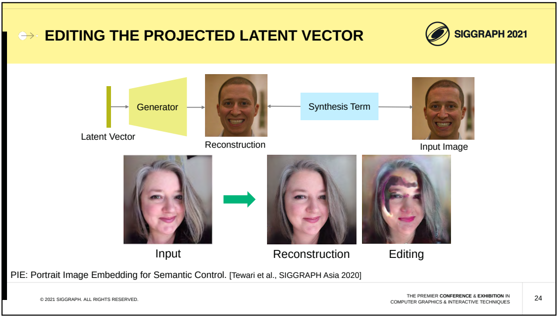
[Deng et al., CVPR 2020]

Image GANs meet Differentiable Rendering for Inverse Graphics and Interpretable 3D Neural Rendering.

[Zhang et al., ICLR 2021]

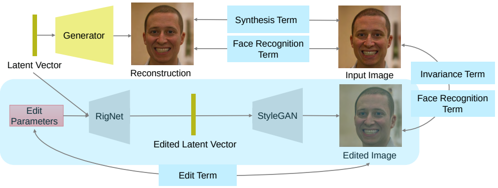

PROJECTING REAL IMAGES
SIGGRAPH 2021
Projection
Generator
Latent Vector
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
22


OPTIMIZATION-BASED METHODS
SIGGRAPH 2021
Generator
Synthesis Term
Latent Vector
Reconstruction
Input Image
Input
Projection
Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? [Abdal et al., ICCV 2019]
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
23


EDITING THE PROJECTED LATENT VECTOR
SIGGRAPH 2021
Generator
Synthesis Term
Latent Vector
Reconstruction
Input Image
Input
Reconstruction
Editing
PIE: Portrait Image Embedding for Semantic Control. [Tewari et al., SIGGRAPH Asia 2020]
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
24

EDITING THE PROJECTED LATENT VECTOR

SIGGRAPH 2021



PIE: Portrait Image Embedding for Semantic Control. [Tewari et al., SIGGRAPH Asia 2020]

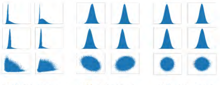
EDITING THE PROJECTED LATENT VECTOR

SIGGRAPH 2021

PIE: Portrait Image Embedding for Semantic Control. [Tewari et al., SIGGRAPH Asia 2020]

REGULARIZATION

SIGGRAPH 2021



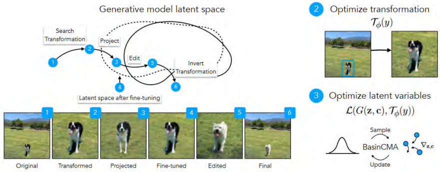
(a) Marginal and pairwise distributions in W space (b) Marginal and pairwise distributions in P space (c) Marginal and pairwise distributions in P_N space

Improved StyleGAN Embedding: Where are the Good Latents? [Abdal et al., 2020]

Improving Inversion and Generation Diversity in StyleGAN using a Gaussianized Latent Space. [Wuff et al., 2020]

TRANSFORMATION

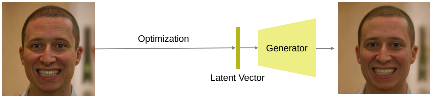
SIGGRAPH 2021



Transforming and Projecting Images into Class-conditional Generative Networks. [Huh et al., ECCV 2020]

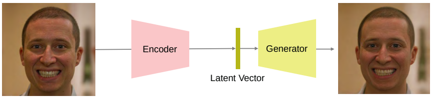
LEARNING-BASED METHODS

SIGGRAPH 2021



LEARNING-BASED METHODS

SIGGRAPH 2021



LEARNING-BASED METHODS

SIGGRAPH 2021

Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. [Richardson et al., CVPR 2021]

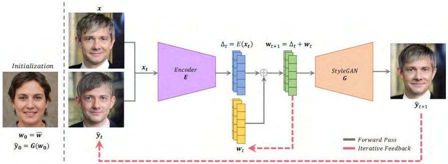
LEARNING-BASED METHODS

SIGGRAPH 2021

Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. [Richardson et al., CVPR 2021]

LEARNING-BASED METHODS

SIGGRAPH 2021



ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. [Alaluf et al., 2021]

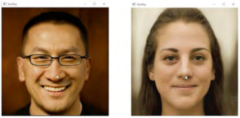
LEARNING-BASED METHODS

SIGGRAPH 2021

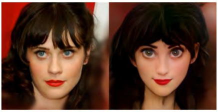
ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. [Alaluf et al., 2021]

NEURAL RENDERING WITH PRETRAINED GANS

SIGGRAPH 2021



[Tewari et al., 2020]

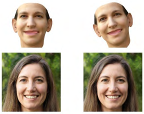


[Alaluf et al., 2021]

CHALLENGES

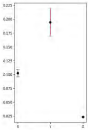
SIGGRAPH 2021

- What can be edited?



Original

In-plane rotation



Distribution of 3D parameters in the StyleGAN-generated dataset

StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. [Tewari et al., CVPR 2020]

CHALLENGES

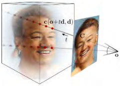
SIGGRAPH 2021

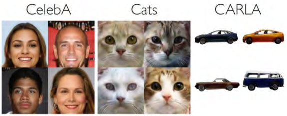
- What can be projected?

Designing an Encoder for StyleGAN Image Manipulation. [Tov et al., SIGGRAPH 2021]

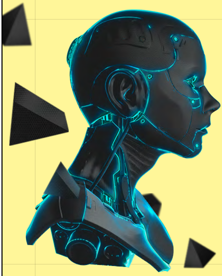
3D GANS

SIGGRAPH 2021





pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. [Chan et al., CVPR 2021]



SIGGRAPH 2021

ADVANCES IN NEURAL RENDERING

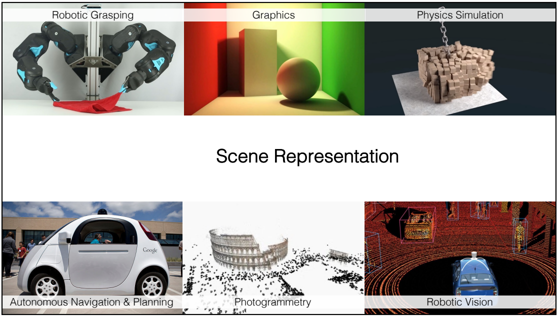
GENERATIVE ADVERSARIAL NETWORKS WITH 3D CONTROL

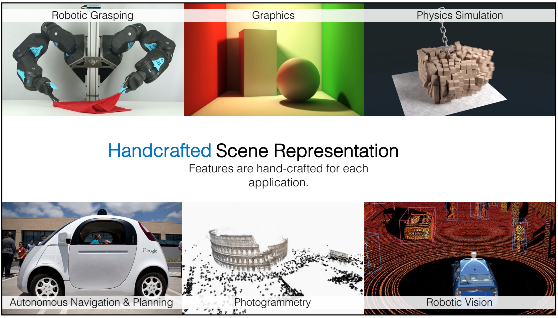
AYUSH TEWARI
MAX PLANCK INSTITUTE FOR INFORMATICS

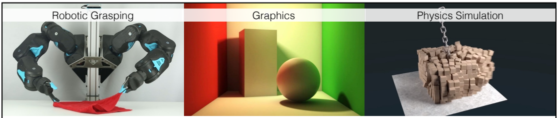
Neural Scene Representation and Rendering

Gordon Wetzstein
Stanford EE, CS

www.computationalimaging.org

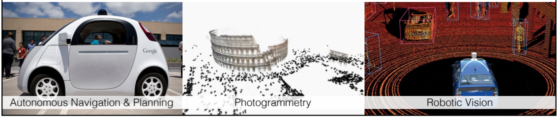






Neural Scene Representation

Learned feature representation of scene.



Neural Scene Representation

Learned feature representation of scene.

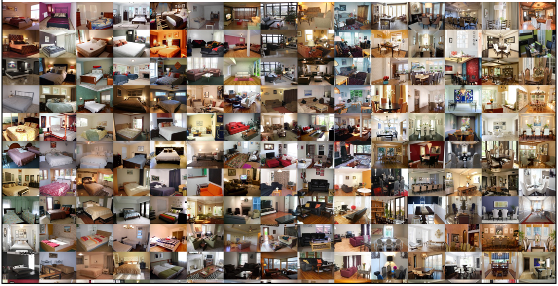
Learn strong priors over scenes.
Represent all properties of scenes.
Learn properties that are hard to model explicitly.

Neural Scene Representation

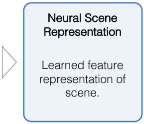
Learned feature representation of scene.

How to train?

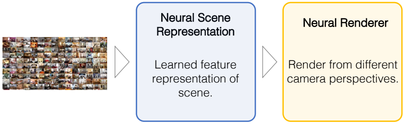
We have a lot of 2D data!



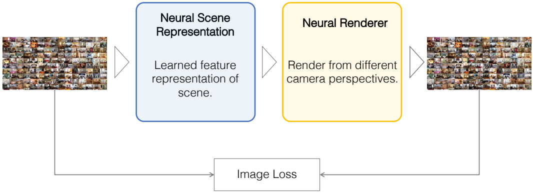
Infer Neural Scene Representation from 2D observations.



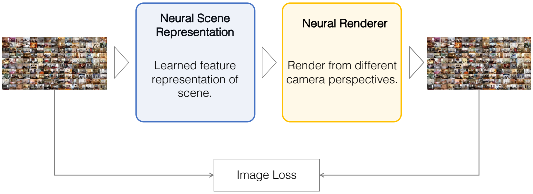
Formulate Neural Renderer.



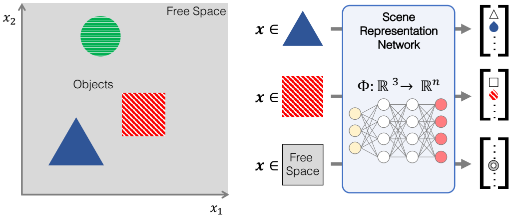
Finally: Predict training views & enforce loss on re-rendering error!



Self-supervised Scene Representation Learning

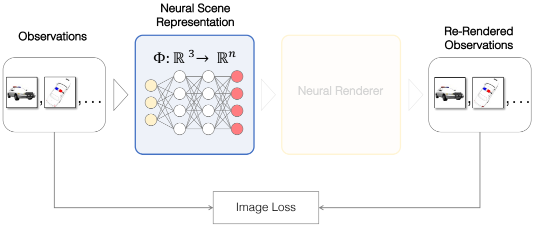


Scene Representation Network parameterizes scene as MLP.



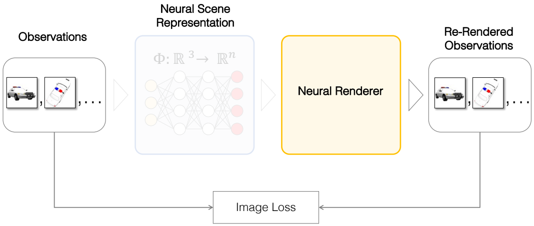
Sitzmann et al., NeurIPS 2019

Scene Representation Networks



Sitzmann et al., NeurIPS 2019

Scene Representation Networks



Sitzmann et al., NeurIPS 2019

Each scene represented by its own SRN.

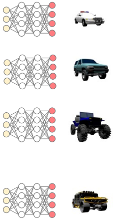
parameters $\phi_0 \in \mathbb{R}^l$

parameters $\phi_1 \in \mathbb{R}^l$

parameters $\phi_2 \in \mathbb{R}^l$

○○○

parameters $\phi_n \in \mathbb{R}^l$



Sitzmann et al., NeurIPS 2019

Manifold assumption.

ϕ_i live on k-dimensional subspace of $\mathbb{R}^l$, $k < l$.

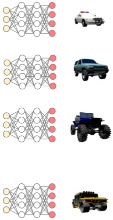
parameters $\phi_0 \in \mathbb{R}^l$

parameters $\phi_1 \in \mathbb{R}^l$

parameters $\phi_2 \in \mathbb{R}^l$

○○○

parameters $\phi_n \in \mathbb{R}^l$



Sitzmann et al., NeurIPS 2019

Represent each scene by low-dimensional embedding.

embedding $z_0 \in \mathbb{R}^k$

embedding $z_1 \in \mathbb{R}^k$

embedding $z_2 \in \mathbb{R}^k$

○○○

embedding $z_n \in \mathbb{R}^k$

Represent each scene with low-dimensional embedding

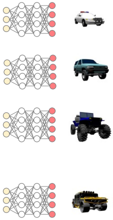
parameters $\phi_0 \in \mathbb{R}^l$

parameters $\phi_1 \in \mathbb{R}^l$

parameters $\phi_2 \in \mathbb{R}^l$

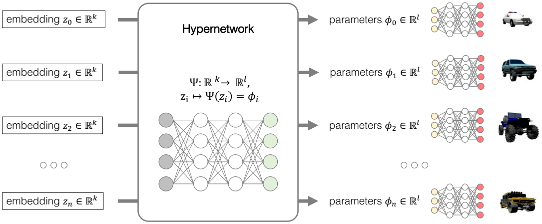
○○○

parameters $\phi_n \in \mathbb{R}^l$



Sitzmann et al., NeurIPS 2019

Map embeddings to SRN parameters via Hypernetwork.



Sitzmann et al., NeurIPS 2019

Novel View Synthesis – Baseline Comparison

Shapenet v2 – ***single-shot reconstruction*** of objects in held-out test set

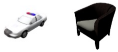
Training

- Shapenet cars / chairs.
- 50 observations per object.

Testing

- Cars / chairs from unseen test set
- Single observation!

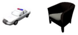
Input pose



Sitzmann et al., NeurIPS 2019

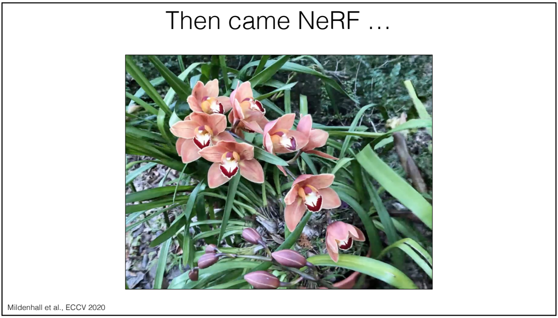
Novel View Synthesis – SRN Output

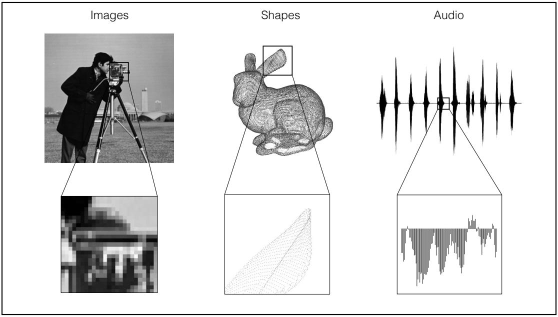
Shapenet v2 – ***single-shot reconstruction*** of objects in held-out test set

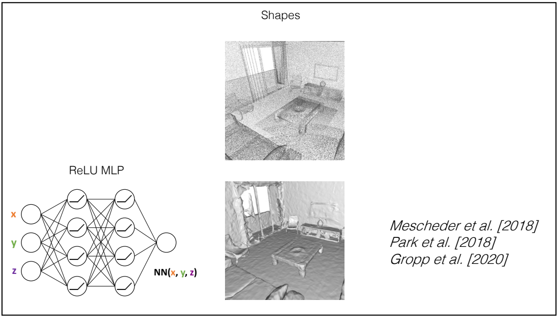


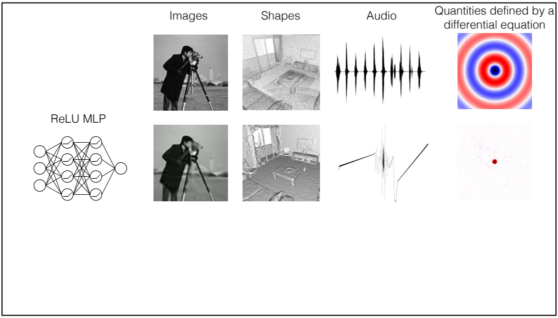
Input pose

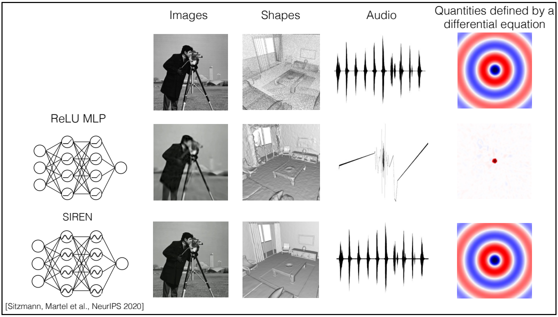
Sitzmann et al., NeurIPS 2019


Then came NeRF …
Mildenhall et al., ECCV 2020


Images
Shapes
Audio

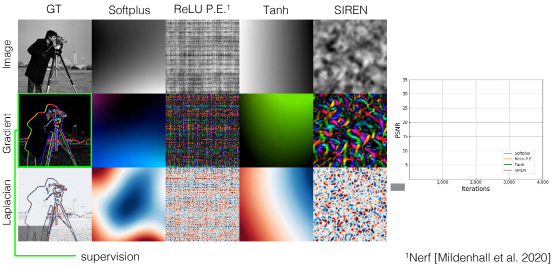

Shapes
ReLU MLP
x
y
z
NN(x, y, z)
Mescheder et al. [2018]
Park et al. [2018]
Gropp et al. [2020]





Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^2$ *spatial coord.*	$f(\mathbf{x}) \in \mathbb{R}^2$ *gray level*	$\mathcal{L}_{\text{Poisson}} = \int_{\Omega} \lVert \nabla \Phi(\mathbf{x}) - \nabla f(\mathbf{x}) \rVert \, \mathrm{d}\mathbf{x}$

Poisson's Equation

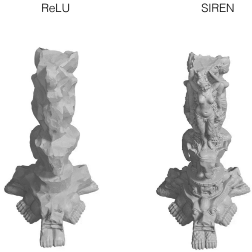


[1]Nerf [Mildenhall et al. 2020]

Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^3$ *spatial coord.*	$f(\mathbf{x}) \in \mathbb{R}$ *signed distance*	$\mathcal{L}_{\text{Eikonal}} = \int_{\Omega_0} \lvert\Phi(\mathbf{x})\rvert + (1 - \langle \nabla\Phi(\mathbf{x}), \nabla f(\mathbf{x}) \rangle)\, d\mathbf{x} + \int_{\Omega} \lVert \lvert\nabla\Phi(\mathbf{x})\rvert - 1 \rVert \, d\mathbf{x}$

Implicit Geometric Regularization for Learning Shapes [Gropp et al. 2020]

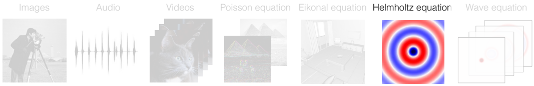
3D Shapes - solving the Eikonal equation



[Sitzmann, Martel et al., NeurIPS 2020]

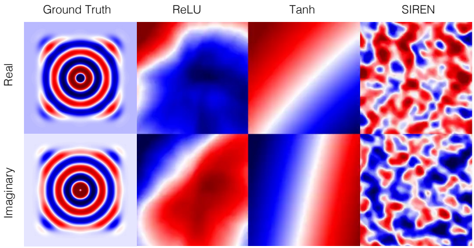
5 layers, 256 hidden units

ReLU SIREN



Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^2$ *spatial coords.*	$f(\mathbf{x}) \in \mathbb{C}$ *complex wave field*	$\mathcal{L}_{\text{Helmholtz}} = \int_\Omega \|(\Delta + m(x)\omega^2)\Phi(\mathbf{x}) - f(\mathbf{x})\| \, \mathrm{d}\mathbf{x}$

Solving the Helmholtz Equation



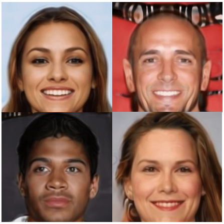
[Sitzmann, Martel et al., NeurIPS 2020]

SIREN is great at overfitting to individual signals,
can it generalize?

π-GAN

π-GAN is a generative-adversarial approach to unsupervised 3D-representation learning from images.

Given a collection of unlabeled images, π-GAN is capable of synthesizing high-quality, view-consistent images from arbitrary camera poses.



Chan et al., CVPR 2021

Rendering 2D Images from Generated Scenes

Rather than directly producing an image, as in a traditional GAN, the π-GAN generator produces a 3D radiance field.

The radiance field can be rendered from arbitrary angles to produce view-consistent 2D images.

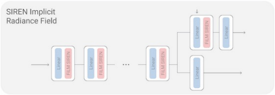
Chan et al., CVPR 2021

Network Architecture

The backbone of π-GAN is a SIREN, which parameterizes the radiance field.

Given 5D positions and viewing directions, the backbone produces color and density samples. These samples are composited along ray using the volumetric rendering method from NeRF [ECCV 2020]

We condition the radiance field to represent a class of scenes using a mapping network and FiLM conditioning.



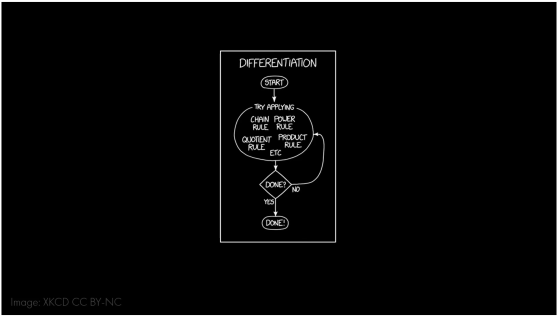
Chan et al., CVPR 2021

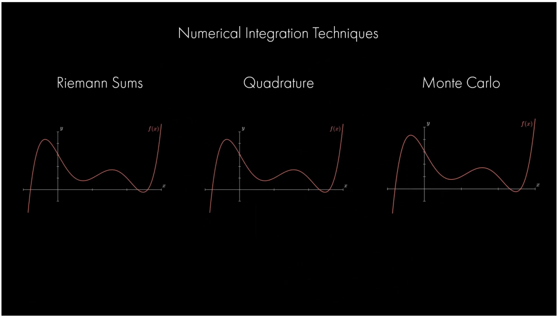
Results

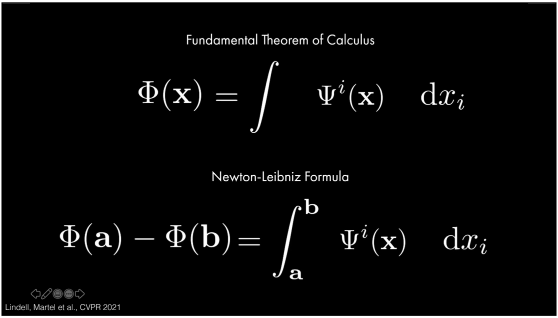
HoloGAN GRAF π-GAN

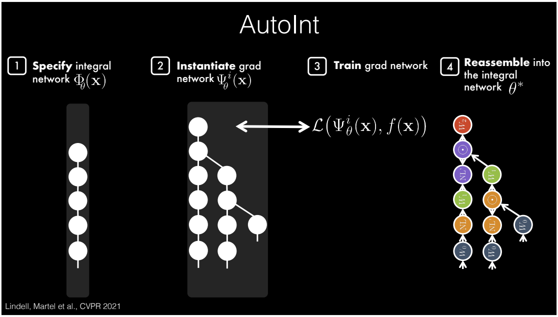
Chan et al., CVPR 2021

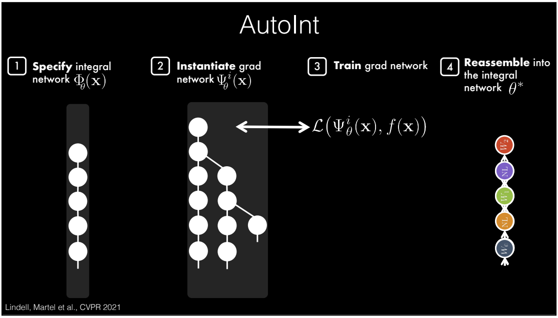

Neural Volume Rendering is Slow!

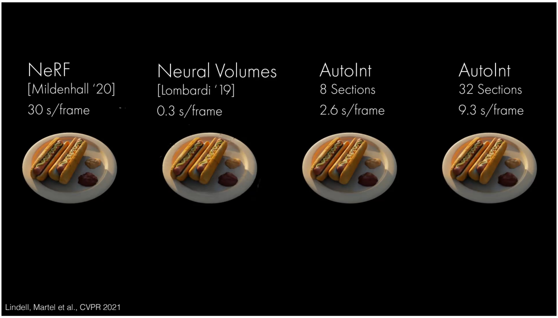

DIFFERENTIATION
START
TRY APPLYING
CHAIN RULE
POWER RULE
QUOTIENT RULE
PRODUCT RULE
ETC
DONE?
NO
YES
DONE!
Image: XKCD CC BY-NC

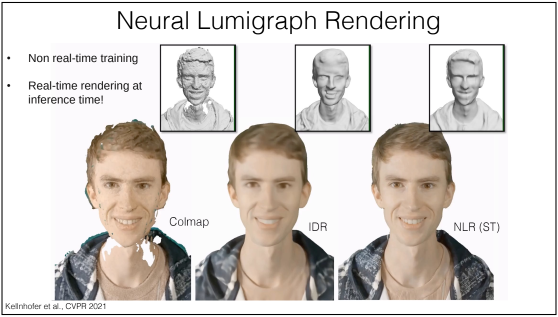

Numerical Integration Techniques
Riemann Sums
Quadrature
Monte Carlo

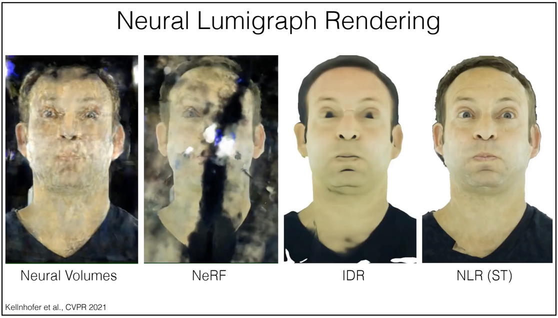

Fundamental Theorem of Calculus
$\Phi(\mathbf{x}) = \int \Psi^i(\mathbf{x}) \, dx_i$
Newton-Leibniz Formula
$\Phi(\mathbf{a}) - \Phi(\mathbf{b}) = \int_{\mathbf{a}}^{\mathbf{b}} \Psi^i(\mathbf{x}) \, dx_i$
Lindell, Martel et al., CVPR 2021


AutoInt
1 **Specify** integral network $\Phi_\theta(\mathbf{x})$
2 **Instantiate** grad network $\Psi_\theta^i(\mathbf{x})$
3 **Train** grad network
4 **Reassemble** into the integral network θ^*
$\mathcal{L}(\Psi_\theta^i(\mathbf{x}), f(\mathbf{x}))$
W_2
$\odot$
NL
W_1
$\odot$
NL
W_0
Lindell, Martel et al., CVPR 2021


AutoInt
1 **Specify** integral network $\Phi_\theta(\mathbf{x})$
2 **Instantiate** grad network $\Psi_\theta^i(\mathbf{x})$
3 **Train** grad network
4 **Reassemble** into the integral network θ^*
$\mathcal{L}(\Psi_\theta^i(\mathbf{x}), f(\mathbf{x}))$
W_2
NL
W_1
NL
W_0
Lindell, Martel et al., CVPR 2021

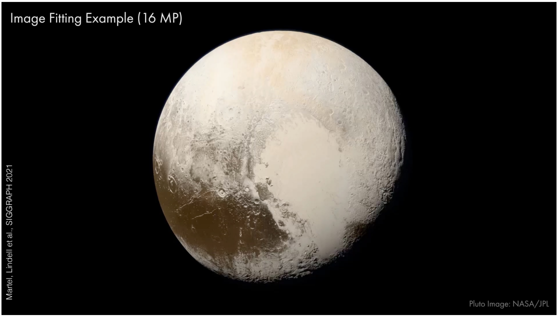

NeRF
[Mildenhall '20]
30 s/frame
Neural Volumes
[Lombardi '19]
0.3 s/frame
AutoInt
8 Sections
2.6 s/frame
AutoInt
32 Sections
9.3 s/frame
Lindell, Martel et al., CVPR 2021

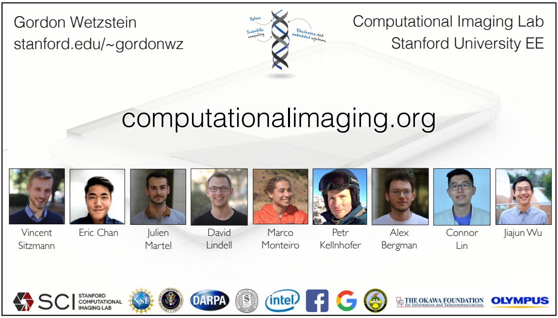

Neural Lumigraph Rendering
• Non real-time training
• Real-time rendering at inference time!
Colmap
IDR
NLR (ST)
Kellnhofer et al., CVPR 2021

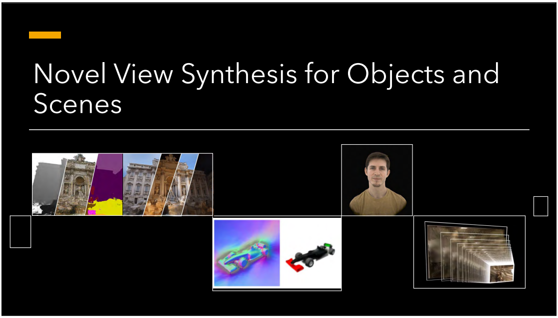

Neural Lumigraph Rendering
Neural Volumes
NeRF
IDR
NLR (ST)
Kellnhofer et al., CVPR 2021

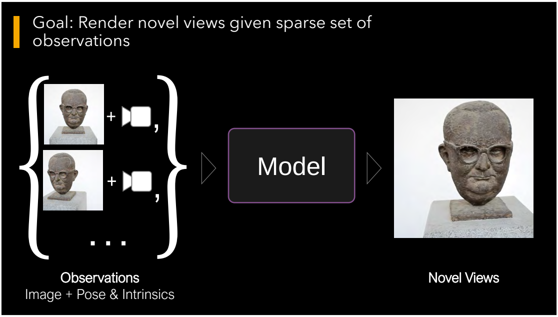
Summary

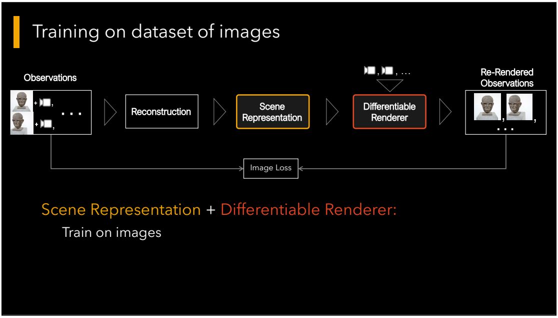
- Scene representation networks are different from CNNs, RNNs, or other network architectures → active area of research
- Need representations that are: scalable, editable, fast to query, expressive, have well-behaved gradients, generalizable, …
- Neural rendering techniques, like volume rendering or sphere tracing, are not real time (in most cases far from it), but some recent progress has been made
- Lots of applications in graphics and beyond: view synthesis, compression, learning priors, solving inverse problems in imaging and geometry processing, scene understanding, path planning, 3D semantic segmentation & classification, …
- **This field is just starting out, lots more work necessary!**

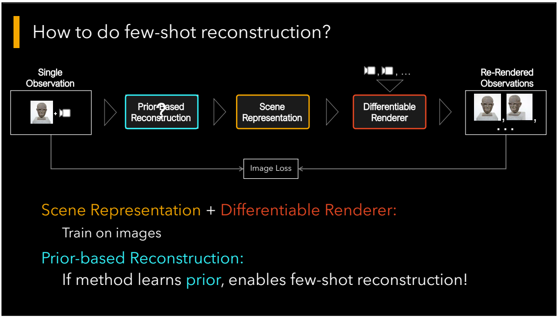


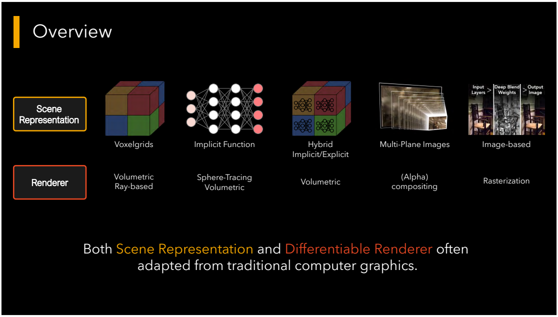

Gordon Wetzstein
stanford.edu/~gordonwz
Computational Imaging Lab
Stanford University EE
computationalimaging.org
Vincent Sitzmann
Eric Chan
Julien Martel
David Lindell
Marco Monteiro
Petr Kellnhofer
Alex Bergman
Connor Lin
Jiajun Wu
SCI STANFORD COMPUTATIONAL IMAGING LAB
DARPA
intel
THE OKAWA FOUNDATION for Information and Telecommunications
OLYMPUS

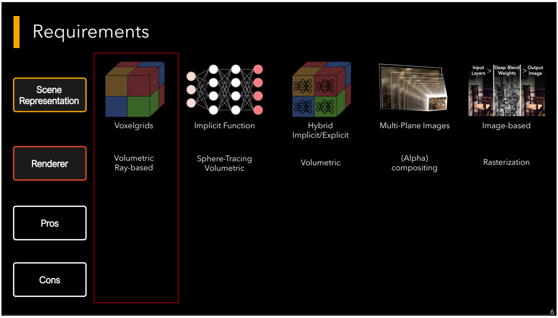

Novel View Synthesis for Objects and Scenes

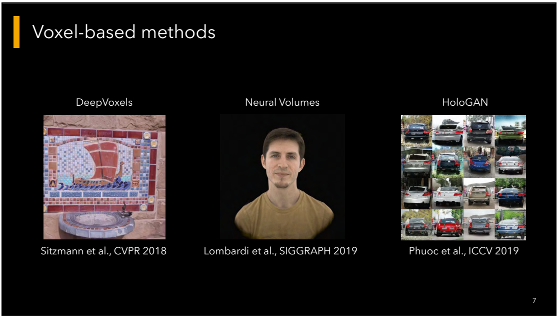

Goal: Render novel views given sparse set of observations
Model
Observations
Image + Pose & Intrinsics
Novel Views

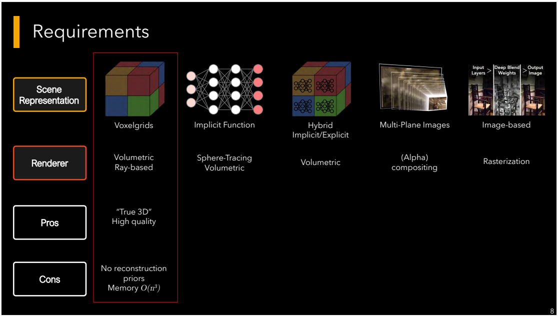

Training on dataset of images
Observations
Reconstruction
Scene Representation
Differentiable Renderer
Re-Rendered Observations
Image Loss
Scene Representation + Differentiable Renderer:
Train on images

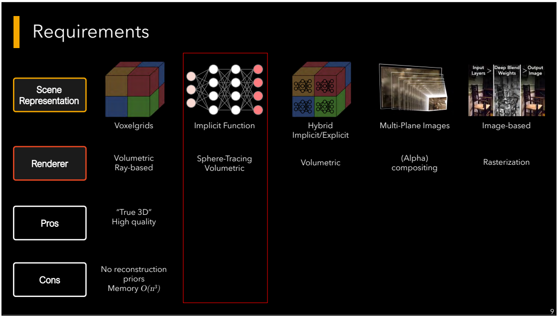

How to do few-shot reconstruction?
Single Observation
Prior-Based Reconstruction
Scene Representation
Differentiable Renderer
Re-Rendered Observations
Image Loss
Scene Representation + Differentiable Renderer:
Train on images
Prior-based Reconstruction:
If method learns prior, enables few-shot reconstruction!

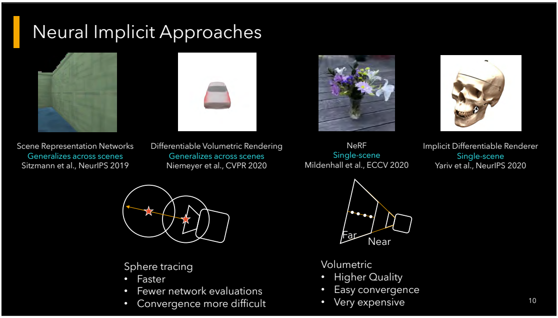

Overview
Scene Representation
Renderer
Voxelgrids
Implicit Function
Hybrid Implicit/Explicit
Multi-Plane Images
Image-based
Input Layers
Deep Blend Weights
Output Image
Volumetric Ray-based
Sphere-Tracing Volumetric
Volumetric
(Alpha) compositing
Rasterization
Both Scene Representation and Differentiable Renderer often adapted from traditional computer graphics.

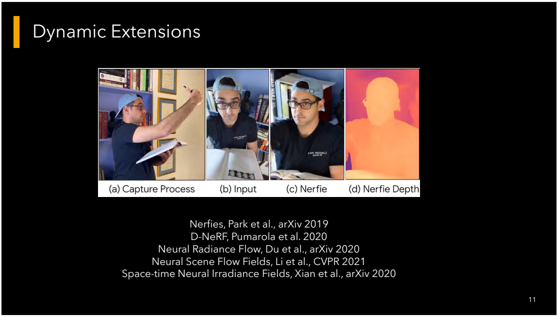

Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Implicit Function
Hybrid Implicit/Explicit
Multi-Plane Images
Image-based
Input Layers
Deep Blend Weights
Output Image
Volumetric Ray-based
Sphere-Tracing Volumetric
Volumetric
(Alpha) compositing
Rasterization
6

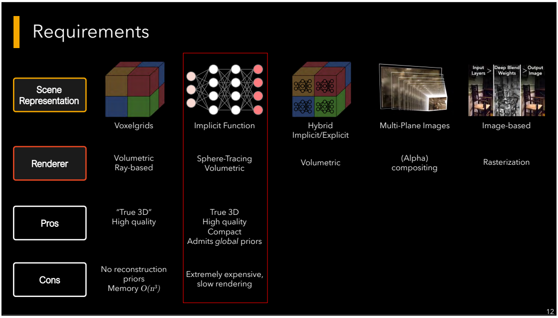

Voxel-based methods
DeepVoxels
Neural Volumes
HoloGAN
Sitzmann et al., CVPR 2018
Lombardi et al., SIGGRAPH 2019
Phuoc et al., ICCV 2019
7

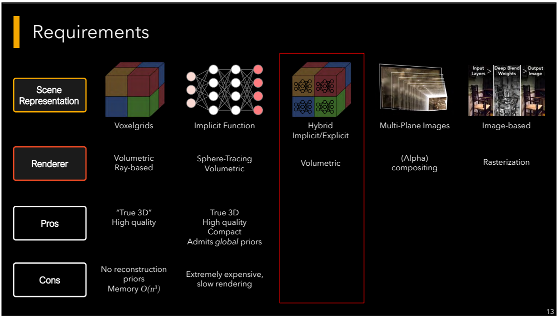

Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Implicit Function
Hybrid Implicit/Explicit
Multi-Plane Images
Input Layers
Deep Blend Weights
Output Image
Image-based
Volumetric Ray-based
Sphere-Tracing Volumetric
Volumetric
(Alpha) compositing
Rasterization
"True 3D" High quality
No reconstruction priors
Memory $O(n^3)$
8

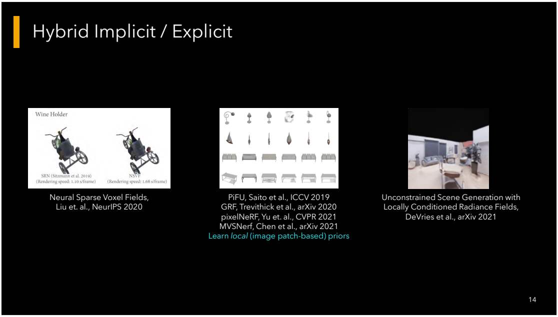

Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Implicit Function
Hybrid Implicit/Explicit
Multi-Plane Images
Input Layers
Deep Blend Weights
Output Image
Image-based
Volumetric Ray-based
Sphere-Tracing Volumetric
Volumetric
(Alpha) compositing
Rasterization
"True 3D" High quality
No reconstruction priors
Memory $O(n^3)$
9

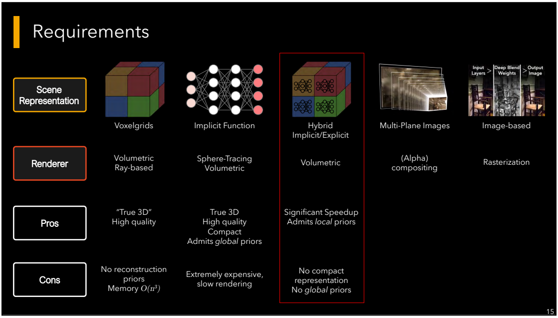

Neural Implicit Approaches
Scene Representation Networks
Generalizes across scenes
Sitzmann et al., NeurIPS 2019
Differentiable Volumetric Rendering
Generalizes across scenes
Niemeyer et al., CVPR 2020
NeRF
Single-scene
Mildenhall et al., ECCV 2020
Implicit Differentiable Renderer
Single-scene
Yariv et al., NeurIPS 2020
Far
Near
Sphere tracing
• Faster
• Fewer network evaluations
• Convergence more difficult
Volumetric
• Higher Quality
• Easy convergence
• Very expensive
10

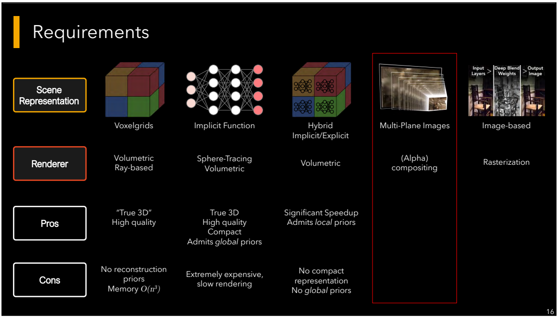

Dynamic Extensions
(a) Capture Process
(b) Input
(c) Nerfie
(d) Nerfie Depth
Nerfies, Park et al., arXiv 2019
D-NeRF, Pumarola et al. 2020
Neural Radiance Flow, Du et al., arXiv 2020
Neural Scene Flow Fields, Li et al., CVPR 2021
Space-time Neural Irradiance Fields, Xian et al., arXiv 2020
11

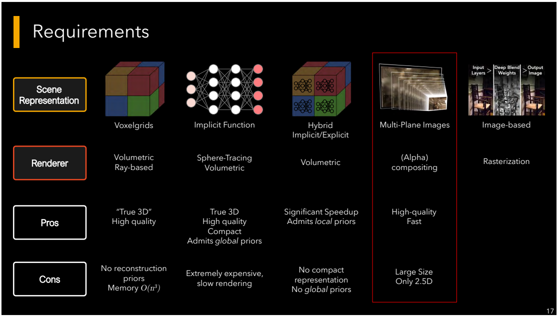

Requirements
Scene Representation
Voxelgrids
Implicit Function
Hybrid Implicit/Explicit
Multi-Plane Images
Input Layers
Deep Blend Weights
Output Image
Image-based
Renderer
Volumetric Ray-based
Sphere-Tracing Volumetric
Volumetric
(Alpha) compositing
Rasterization
Pros
"True 3D" High quality
True 3D High quality Compact Admits global priors
Cons
No reconstruction priors Memory $O(n^3)$
Extremely expensive, slow rendering
12


Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Volumetric
Ray-based
"True 3D"
High quality
No reconstruction priors
Memory $O(n^3)$
Implicit Function
Sphere-Tracing
Volumetric
True 3D
High quality
Compact
Admits *global* priors
Extremely expensive, slow rendering
Hybrid Implicit/Explicit
Volumetric
Multi-Plane Images
(Alpha) compositing
Input Layers
Deep Blend Weights
Output Image
Image-based
Rasterization
13


Hybrid Implicit / Explicit
Wine Holder
SRN (Sitzmann et al. 2019)
(Rendering speed: 1.10 s/frame)
NSVF
(Rendering speed: 1.68 s/frame)
Neural Sparse Voxel Fields,
Liu et. al., NeurIPS 2020
PiFU, Saito et al., ICCV 2019
GRF, Trevithick et al., arXiv 2020
pixelNeRF, Yu et. al., CVPR 2021
MVSNerf, Chen et al., arXiv 2021
Learn *local* (image patch-based) priors
Unconstrained Scene Generation with
Locally Conditioned Radiance Fields,
DeVries et al., arXiv 2021
14


Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Volumetric
Ray-based
"True 3D"
High quality
No reconstruction priors
Memory $O(n^3)$
Implicit Function
Sphere-Tracing
Volumetric
True 3D
High quality
Compact
Admits *global* priors
Extremely expensive, slow rendering
Hybrid Implicit/Explicit
Volumetric
Significant Speedup
Admits *local* priors
No compact representation
No *global* priors
Multi-Plane Images
(Alpha) compositing
Input Layers
Deep Blend Weights
Output Image
Image-based
Rasterization
15


Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Volumetric Ray-based
"True 3D" High quality
No reconstruction priors Memory $O(n^3)$
Implicit Function
Sphere-Tracing Volumetric
True 3D High quality Compact Admits *global* priors
Extremely expensive, slow rendering
Hybrid Implicit/Explicit
Volumetric
Significant Speedup Admits *local* priors
No compact representation No *global* priors
Multi-Plane Images
(Alpha) compositing
Input Layers
Deep Blend Weights
Output Image
Image-based
Rasterization
16


Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Volumetric Ray-based
"True 3D" High quality
No reconstruction priors Memory $O(n^3)$
Implicit Function
Sphere-Tracing Volumetric
True 3D High quality Compact Admits *global* priors
Extremely expensive, slow rendering
Hybrid Implicit/Explicit
Volumetric
Significant Speedup Admits *local* priors
No compact representation No *global* priors
Multi-Plane Images
(Alpha) compositing
High-quality Fast
Large Size Only 2.5D
Input Layers
Deep Blend Weights
Output Image
Image-based
Rasterization
17

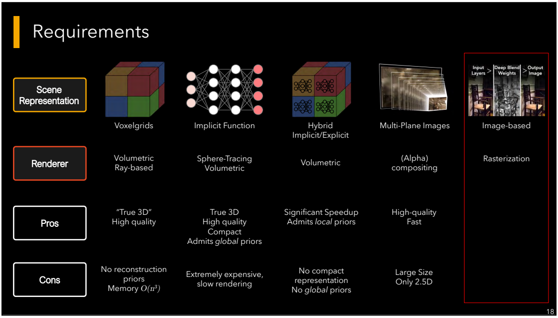
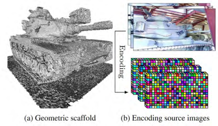

Requirements
Scene Representation
Renderer
Pros
Cons
Voxelgrids
Volumetric Ray-based
"True 3D" High quality
No reconstruction priors Memory $O(n^3)$
Implicit Function
Sphere-Tracing Volumetric
True 3D High quality Compact Admits *global* priors
Extremely expensive, slow rendering
Hybrid Implicit/Explicit
Volumetric
Significant Speedup Admits *local* priors
No compact representation No *global* priors
Multi-Plane Images
(Alpha) compositing
High-quality Fast
Large Size Only 2.5D
Input Layers
Deep Blend Weights
Output Image
Image-based
Rasterization
18

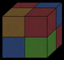
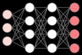
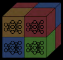
Image-based methods



Stable View Synthesis
Riegler et al., CVPR 2021

IBRNet, Wang et al., CVPR 2021

Requirements

Scene Representation	 Voxelgrids	 Implicit Function	 Hybrid Implicit/Explicit	 Multi-Plane Images	 Image-based
Renderer	Volumetric Ray-based	Sphere-Tracing Volumetric	Volumetric	(Alpha) compositing	Rasterization / Volumetric
Pros	"True 3D" High quality	True 3D High quality Compact Admits *global* priors	Significant Speedup Admits *local* priors	High-quality Fast	High-quality Fast
Cons	No reconstruction priors Memory $O(n^3)$	Extremely expensive, slow rendering	No compact representation No *global* priors Memory $O(n^3)$	Large Size Only 2.5D	Not compact: Needs source images.

Summary: Open Challenges

Expensive Rendering

- Rendering requires *hundreds of samples per ray – at train and test time.*
- How to do non-Lambertian effects? Multi-bounce barely tractable.

Generalization

- Local conditioning enables stronger generalization, but doesn't learn object-/scene-centric representations. Can we have both?

Scene Understanding

- Lots of important applications outside of computer graphics worth exploring!

Neural Volumetric Rendering: NeRF, etc.

SIGGRAPH 2021

Neural Rendering Course

Ben Mildenhall
Google Research
bmild.github.io

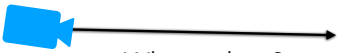
1

Neural Volumetric Rendering

2

Neural Volumetric Rendering

querying the radiance value along rays through 3D space



What colour?

3

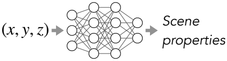
Neural **Volumetric** Rendering

continuous, differentiable
rendering model without
concrete ray/surface intersections

4

Neural Volumetric Rendering

using a neural network as a
scene representation, rather
than a voxel grid of data



5

Motivation: novel view synthesis

Inputs: sparse, unstructured
photographs of a scene

Outputs: representation allowing us to
render *new* views of that scene

6

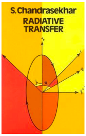
Overview

- Volumetric rendering math
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)
- NeRF improvements and extensions

Overview

- Volumetric rendering math
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)
- NeRF improvements and extensions

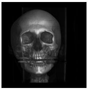
Traditional volumetric rendering



- Theory of volume rendering co-opted from physics in the 1980s: absorption, emission, out-scattering/in-scattering
- Adapted for visualising medical data and linked with alpha compositing
- Modern path tracers use sophisticated Monte Carlo methods to render volumetric effects

Chandrasekhar 1950, *Radiative Transfer*
Kajia 1984, *Ray Tracing Volume Densities*
Levoy 1988, *Display of Surfaces from Volume Data*
Max 1995, *Optical Models for Direct Volume Rendering*
Porter and Duff 1984, *Compositing Digital Images*
Novak et al 2018, *Monte Carlo methods for physically based volume rendering*

Traditional volumetric rendering


Medical data visualisation [Levoy]

- Theory of volume rendering co-opted from physics in the 1980s: absorption, emission, out-scattering/in-scattering
- Adapted for visualising medical data and linked with alpha compositing
- Modern path tracers use sophisticated Monte Carlo methods to render volumetric effects

Alpha compositing [Porter and Duff]

Chandrasekhar 1950, *Radiative Transfer*
Kajia 1984, *Ray Tracing Volume Densities*
Levoy 1988, *Display of Surfaces from Volume Data*
Max 1995, *Optical Models for Direct Volume Rendering*
Porter and Duff 1984, *Compositing Digital Images*
Novak et al 2018, *Monte Carlo methods for physically based volume rendering*

10

Traditional volumetric rendering

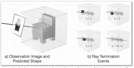

Physically-based Monte Carlo rendering [Novak et al]

- Theory of volume rendering co-opted from physics in the 1980s: absorption, emission, out-scattering/in-scattering
- Adapted for visualising medical data and linked with alpha compositing
- Modern path tracers use sophisticated Monte Carlo methods to render volumetric effects

Chandrasekhar 1950, *Radiative Transfer*
Kajia 1984, *Ray Tracing Volume Densities*
Levoy 1988, *Display of Surfaces from Volume Data*
Max 1995, *Optical Models for Direct Volume Rendering*
Porter and Duff 1984, *Compositing Digital Images*
Novak et al 2018, *Monte Carlo methods for physically based volume rendering*

11

Volumetric rendering and machine learning



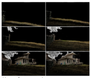
"Probabilistic" voxel grid rendering [Tulsiani et al]

- Various volume-rendering-esque methods devised for 3D shape reconstruction methods
- Scaled up to higher resolution volumes to achieve excellent view synthesis results

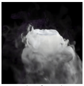
Tulsiani et al 2017, *Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency*
Henzler et al 2019, *Escaping Plato's Cave: 3D Shape From Adversarial Rendering*
Zhou et al 2018, *Stereo Magnification: Learning View Synthesis using Multiplane Images*
Lombardi et al 2019, *Neural Volumes: Learning Dynamic Renderable Volumes from Images*

12

Volumetric rendering and machine learning



Slices from a volumetric scene representation [Zhou et al]



View synthesis from a dynamic voxel grid [Lombardi et al]

- Various volume-rendering-esque methods devised for 3D shape reconstruction methods
- Scaled up to higher resolution voxel grids, ML methods can achieve excellent view synthesis results

Tulsiani et al 2017, *Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency*
Henzler et al 2019, *Escaping Plato's Cave: 3D Shape From Adversarial Rendering*
Zhou et al 2018, *Stereo Magnification: Learning View Synthesis using Multiplane Images*
Lombardi et al 2019, *Neural Volumes: Learning Dynamic Renderable Volumes from Images*

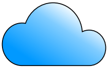
13

Volumetric formulation for NeRF

Max and Chen 2010, *Local and Global Illumination in the Volume Rendering Integral*

14

Volumetric formulation for NeRF

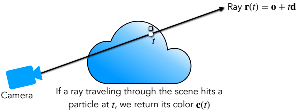


Scene is a cloud of tiny colored particles

Max and Chen 2010, *Local and Global Illumination in the Volume Rendering Integral*

15

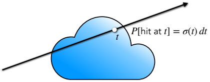
Volumetric formulation for NeRF



If a ray traveling through the scene hits a particle at t, we return its color $\mathbf{c}(t)$

16

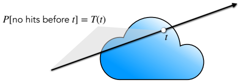
Volumetric formulation for NeRF



This notion is *probabilistic:* chance that ray stops in a small interval around t is $\sigma(t)\, dt$.
σ is known as the "volume density"

17

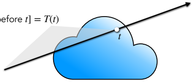
Volumetric formulation for NeRF



To determine if t is the *first* hit, need to know $T(t)$: probability that the ray didn't hit any particles earlier.
$T(t)$ is called "transmittance"

18

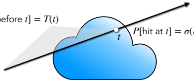
Volumetric formulation for NeRF



To determine if t is the *first* hit, need to know $T(t)$:
probability that the ray didn't hit any particles earlier.
$T(t)$ is called "transmittance"
We assume σ is known and want to use it to calculate T

19

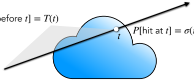
Volumetric formulation for NeRF



σ and T are related by the probability fact that
P[no hits before $t + dt$] = P[no hits before t] $\times$ P[no hit at t]

20

Volumetric formulation for NeRF



These are related by the probability fact that

$$T(t+dt) \quad = \quad T(t) \quad \times \quad (1 - \sigma(t)dt)$$

21

Volumetric formulation for NeRF

$$T(t + dt) = T(t)(1 - \sigma(t)dt)$$

Volumetric formulation for NeRF

$$T(t + dt) = T(t)(1 - \sigma(t)dt)$$

Split up differential $\Rightarrow$ $T(t) + T'(t)dt = T(t) - T(t)\sigma(t)dt$

Volumetric formulation for NeRF

$$T(t + dt) = T(t)(1 - \sigma(t)dt)$$

Split up differential $\Rightarrow$ $T(t) + T'(t)dt = T(t) - T(t)\sigma(t)dt$

Rearrange $\Rightarrow$ $\dfrac{T'(t)}{T(t)}dt = -\sigma(t)dt$

Volumetric formulation for NeRF

$$T(t + dt) = T(t)(1 - \sigma(t)dt)$$

Split up differential $\Rightarrow$ $T(t) + T'(t)dt = T(t) - T(t)\sigma(t)dt$

Rearrange $\Rightarrow$ $\frac{T'(t)}{T(t)}dt = -\sigma(t)dt$

Integrate $\Rightarrow$ $\log T(t) = -\int_{t_0}^{t} \sigma(s)ds$

Volumetric formulation for NeRF

Thus, the probability that a ray first hits a particle at t is

$$T(t)\sigma(t)\,dt = \exp\left(-\int_{t_0}^{t} \sigma(t)\right)\sigma(t)\,dt$$

Volumetric formulation for NeRF

Thus, the probability that a ray first hits a particle at t is

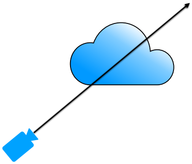
$$T(t)\sigma(t)\,dt = \exp\left(-\int_{t_0}^{t} \sigma(t)\right)\sigma(t)\,dt$$

And expected color returned by the ray will be

$$\int_{t_0}^{t_1} T(t)\sigma(t)\mathbf{c}(t)\,dt$$

Note the nested integral!

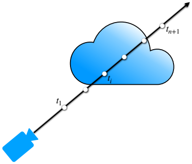
Approximating the nested integral



We use quadrature to approximate the nested integral,

28

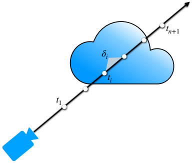
Approximating the nested integral



We use quadrature to approximate the nested integral,
splitting the ray up into n segments with endpoints $\{t_1, t_2, \ldots, t_{n+1}\}$

29

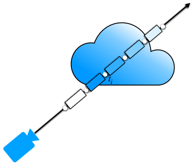
Approximating the nested integral



We use quadrature to approximate the nested integral,
splitting the ray up into n segments with endpoints $\{t_1, t_2, \ldots, t_{n+1}\}$
with lengths $\delta_i = t_{i+1} - t_i$

30

Approximating the nested integral



We assume volume density and color are roughly constant within each interval

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt$$

This allows us to break the outer integral

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

This allows us to break the outer integral into a sum of analytically tractable integrals

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} \boxed{T(t)}\sigma_i \mathbf{c}_i\,dt$$

Catch: piecewise constant density and color
do not imply constant transmittance!

34

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} \boxed{T(t)}\sigma_i \mathbf{c}_i\,dt$$

Catch: piecewise constant density and color
do not imply constant transmittance!

Important to account for how early part of a
segment blocks later part when σ_i is high

35

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

For $t \in [t_i, t_{i+1}]$, $T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i\,ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i\,ds\right)$

36

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

For $t \in [t_i, t_{i+1}]$, $T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i\,ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i\,ds\right)$

$$\exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) = T_i$$

"How much is blocked by all previous segments?"

37

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

For $t \in [t_i, t_{i+1}]$, $T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i\,ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i\,ds\right)$

"How much is blocked partway through the current segment?" $\exp\left(-\sigma_i(t - t_i)\right)$

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

39

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

Substitute $$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp\left(-\sigma_i(t - t_i)\right)\,dt$$

40

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp\left(-\sigma_i(t - t_i)\right)\,dt$$

Integrate $$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \frac{\exp\left(-\sigma_i(t_{i+1} - t_i)\right) - 1}{-\sigma_i}$$

41

Approximating the nested integral

$$\int T(t)\sigma(t)\mathbf{c}(t)\,dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i\,dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp\left(-\sigma_i(t - t_i)\right)\,dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \frac{\exp\left(-\sigma_i(t_{i+1} - t_i)\right) - 1}{-\sigma_i}$$

Cancel σ_i $$= \sum_{i=1}^{n} T_i \mathbf{c}_i \left(1 - \exp(-\sigma_i \delta_i)\right)$$

42

Connection to alpha compositing

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$

$$= \sum_{i=1}^{n} T_i \mathbf{c}_i (1 - \exp(-\sigma_i \delta_i))$$

Connection to alpha compositing

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i) \Rightarrow$$

$$= \sum_{i=1}^{n} T_i \mathbf{c}_i (1 - \exp(-\sigma_i \delta_i))$$

$$\text{color} = \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i = \sum_{i=1}^{n} T_i \mathbf{c}_i (1 - \exp(-\sigma_i \delta_i))$$

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

Summary: volume rendering integral estimate

Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

$$\mathbf{c} \approx \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

colors

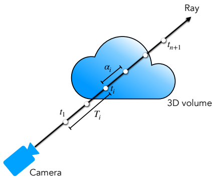
weights

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:

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$



Summary: volume rendering integral estimate

Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

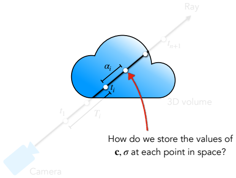
$$\mathbf{c} \approx \sum_{i=1}^{n} \underbrace{T_i \alpha_i}_{\text{weights}} \underbrace{\mathbf{c}_i}_{\text{colors}}$$

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:

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$



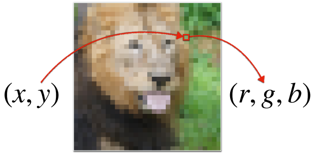
46

Overview

- Volumetric rendering math
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)
- NeRF improvements and extensions

47

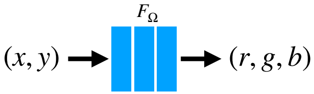
Toy problem: storing 2D image data



Usually we store an image as a
2D grid of RGB color values

48

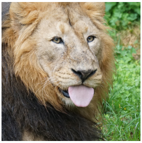
Toy problem: storing 2D image data



What if we train a simple fully-connected network (MLP) to do this instead?

49

Naive approach fails!


Ground truth image


Neural network output

50

Problem:

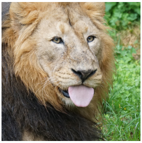
"Standard" coordinate-based MLPs cannot represent high frequency functions

51

Solution:

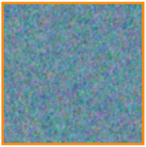
Pass input coordinates through a
high frequency mapping first

Problem solved



Ground truth image

Neural network output **without** high frequency mapping



Neural network output **with** high frequency mapping

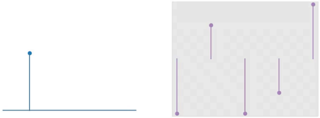
Input coordinate mapping

- Simple formula: apply a tall skinny matrix **B** to input coordinate vector **x**, then pass through *sin* and *cos*:

$$\gamma(\mathbf{x}) = (\sin(2\pi \mathbf{B}\mathbf{x}), \cos(2\pi \mathbf{B}\mathbf{x}))$$

- Passing network a subset of the Fourier basis functions. Same effect from:
  - Positional encoding
  - Fourier features
  - SIREN

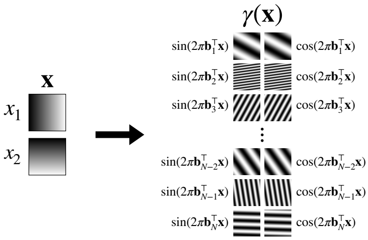
Simple 1D example



Spatial position   Positional encoding features

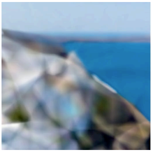
55

Simple 2D example

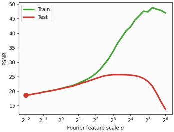


56

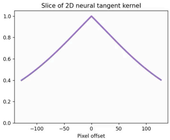
Scaling frequency matrix **B** traverses underfitting-overfitting curve



Network output



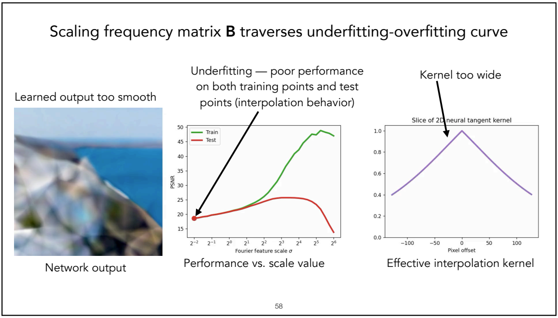
Performance vs. scale value

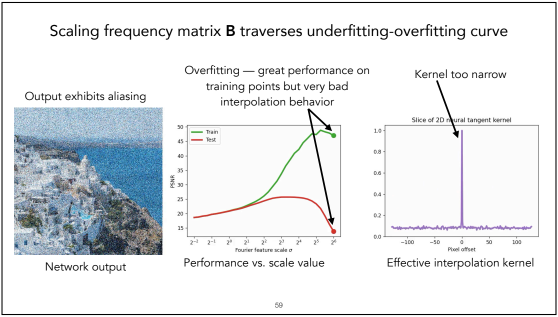


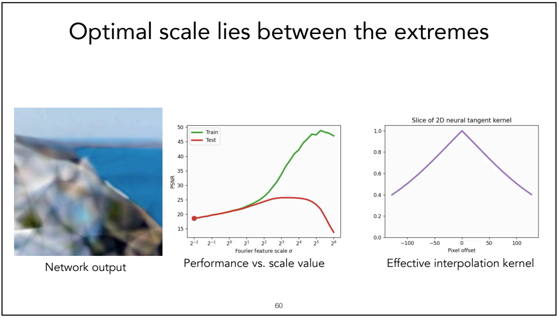
Effective interpolation kernel

Tancik et al 2020, *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*

57


Scaling frequency matrix **B** traverses underfitting-overfitting curve
Learned output too smooth
Network output
Underfitting — poor performance on both training points and test points (interpolation behavior)
Train
Test
PSNR
Fourier feature scale σ
Performance vs. scale value
Kernel too wide
Slice of 2D neural tangent kernel
Pixel offset
Effective interpolation kernel
58


Scaling frequency matrix **B** traverses underfitting-overfitting curve
Output exhibits aliasing
Network output
Overfitting — great performance on training points but very bad interpolation behavior
Train
Test
PSNR
Fourier feature scale σ
Performance vs. scale value
Kernel too narrow
Slice of 2D neural tangent kernel
Pixel offset
Effective interpolation kernel
59


Optimal scale lies between the extremes
Network output
Train
Test
PSNR
Fourier feature scale σ
Performance vs. scale value
Slice of 2D neural tangent kernel
Pixel offset
Effective interpolation kernel
60

Overview

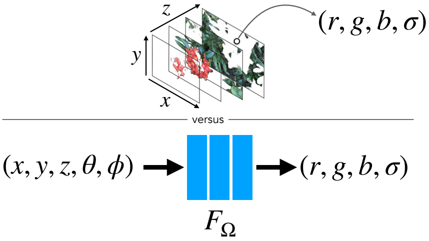
- Volumetric rendering math
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)
- NeRF improvements and extensions

61

NeRF = volume rendering + coordinate-based network

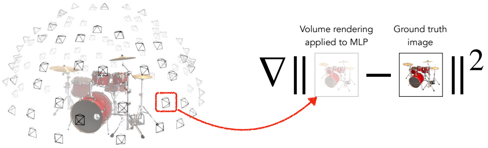
62

Neural network replaces large N-d array



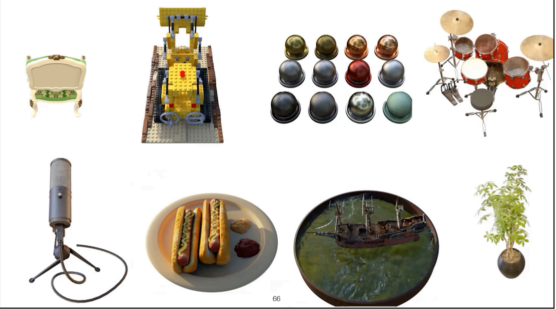
63

Train network to reproduce input views of scene
using gradient descent



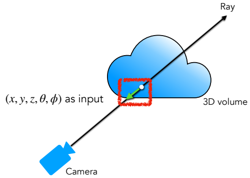
64

65

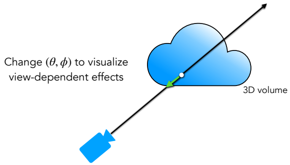


66

Viewing directions as input



Viewing directions as input



Visualizing view-dependent effects

Radiance distribution for point on side of ship

Radiance distribution for point on water's surface

Visualizing view-dependent effects

Regular NeRF rendering

Manipulating input viewing directions

70

Visualizing learned density field as geometry

Regular NeRF rendering

Expected ray termination depth

71

Visualizing learned density field as geometry

Regular NeRF rendering

Expected ray termination depth

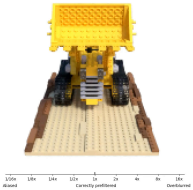
72

Overview

- Volumetric rendering math
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)
- NeRF improvements and extensions

NeRF problems

- Scene representation is not anti-aliased



[Barron] prefiltered positional encoding

Barron et al 2021, *Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields*

NeRF problems

- Rendering is very slow

[Hedman] realtime online viewer

Reiser et al 2021, *KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs*
Garbin et al 2021, *FastNeRF: High-Fidelity Neural Rendering at 200FPS*
Yu et al 2021, *PlenOctrees For Real-time Rendering of Neural Radiance Fields*
Hedman et al 2021, *Baking Neural Radiance Fields for Real-Time View Synthesis*

NeRF problems

- Network must be retrained for every scene
- Requires many input images

[Wang] network never trained on this scene!

Trevithick et al 2020, *GRF: Learning a General Radiance Field for 3D Scene Representation and Rendering*
Wang et al 2021, *IBRNet: Learning Multi-View Image-Based Rendering*
Yu et al 2021, *pixelNeRF: Neural Radiance Fields from One or Few Images*

NeRF problems

[Park] trained on selfie video

[Srinivasan] trained on multiple lighting conditions

[Tancik] trained on tourist photos

- Needs scene to be static and have fixed lighting

Bi et al 2020, *Neural Reflectance Fields for Appearance Acquisition*
Park et al 2020, *Nerfies: Deformable Neural Radiance Fields*
Li et al 2021, *Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes*
Srinivasan et al 2021, *NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis*
Tancik et al 2021, *Learned Initializations for Optimizing Coordinate-Based Neural Representations*
Martin-Brualla et al 2021, *NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections*

• • •

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

Neural Volumetric Rendering: NeRF, etc.

SIGGRAPH 2021
Neural Rendering Course

Ben Mildenhall
Google Research
bmild.github.io

79

Fast Rendering of Neural Radiance Fields

Lingjie Liu

Max Planck Institute for Informatics

Graphics Vision & Video | max planck institut informatik | FACEBOOK AI | NUS National University of Singapore

Background

Background

- It is challenging to use classical computer graphics techniques for photo-realistic free-viewpoint rendering.
  - Often infeasible to acquire a detailed appearance and geometry model.

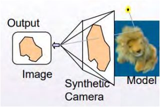


Image from [Cohen et al. 1999]

Background

- Neural scene representations and neural rendering for free-viewpoint rendering.
  - Scene representation: Using feature representations to describe geometry and appearance information.
  - Rendering: Using learned representations to synthesize novel views.

A set of posed images

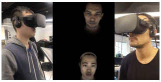
→ Scene Representations →

Synthesized Novel Views

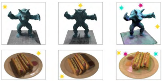
4 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Background

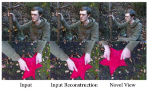
- Applications of neural scene representations and neural rendering



AR / VR



Relighting



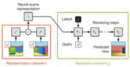
Dynamic Scene Rendering

Free-viewpoint Rendering

5 Fast Rendering of Neural Radiance Fields, Lingjie Liu

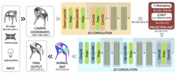
Background

Scene representations for neural rendering:



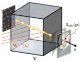
Generative Query Networks
[Eslami et al. 2018]

[Flynn et al., 2016; Zhou et al., 2018b; Mildenhall et al. 2019]
Multiplane Images (MPIs)



RenderNet [Nguyen-Phuoc et al. 2018]
Voxel Grids + CNN decoder

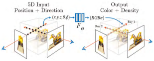
DeepVoxels
[Sitzmann et al. 2019]



Neural Volumes
[Lombardi et al. 2019]

Voxel Grids + Ray Marching

SRN [Sitzmann et al. 2019b]



NeRF [Mildenhall et al. 2020]

Implicit Fields

6 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Background

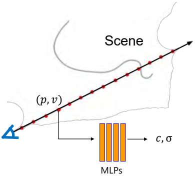


Illustration of volume rendering in NeRF

Background

- NeRF (Mildenhall et al. 2020)

Rendering speed: 100 s/frame

Image resolution: 1920x1080

To render an image at 1920x1080 pixels, how many calls of the MLPs are needed?

(1920x1080) x 192 = 398, 131, 200

It takes about 100 seconds to render such an image using an NVIDIA V100 GPU

Background

- NeRF (Mildenhall et al. 2020)

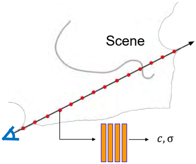


Illustration of volume rendering in NeRF

To render an image at 1920x1080 pixels, how many calls of the MLPs are needed?

(1920x1080) x 192 = 398, 131, 200

It takes about 100 seconds to render such an image using an NVIDIA V100 GPU

Two possible ideas to accelerate the rendering process:

1. Reduce sampling points.
2. Reduce the runtime for one pass.

FACEBOOK AI

Neural Sparse Voxel Fields

Lingjie Liu*, Jiatao Gu*, Kyaw Zaw Lin, Tat-Seng Chua, Christian Theobalt (* equal contribution)

NeurIPS 2020 Spotlight

FACEBOOK AI

Our Method -- Neural Sparse Voxel Fields **(NSVF)**

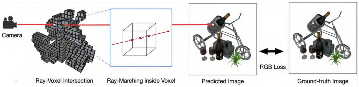
- Avoid sampling points in empty space as much as possible.
- Neural Sparse Voxel Fields (NSVF), a hybrid scene representation for fast and high-quality free-viewpoint rendering.

Illustration of NSVF

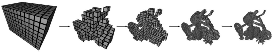
FACEBOOK AI

Our Method (NSVF)

- Scene Representation - Neural Sparse Voxel Fields (NSVF).
- Volume Rendering with NSVF.



- Progressive Learning: we train NSVF progressively with the differentiable volume rendering operation from a set of posed 2D images.



Scene Representation - NSVF

The scene is modeled as a set of voxel-bounded implicit functions:

$$F_\theta(\boldsymbol{p}, \boldsymbol{v})$$

The relevant non-empty parts of a scene are contained within a set of sparse bounding voxels:

$$\mathcal{V} = \{V_1 \dots V_K\}$$

13 Fast Rendering of Neural Radiance Fields, Lingjie Liu

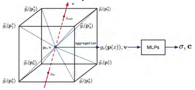
Scene Representation - NSVF

A voxel-bounded implicit field

- For a given point **p** inside voxel Vi, the voxel-bounded implicit field is defined as:

$$F_\theta^i : (\boldsymbol{g}_i(\boldsymbol{p}), \boldsymbol{v}) \rightarrow (\boldsymbol{c}, \sigma), \forall \boldsymbol{p} \in V_i,$$

voxel embedding *ray direction* *color* *density*



- Voxel embedding is defined as:

Trilinear interpolation *Voxel features (e.g. learnable voxel embeddings)*

$$g_i(\boldsymbol{p}) = \boxed{\zeta}\left(\boxed{\chi}\left(\widetilde{g}_i(\boldsymbol{p}_1^*), \dots, \boxed{\widetilde{g}_i}(\boldsymbol{p}_8^*)\right)\right)$$

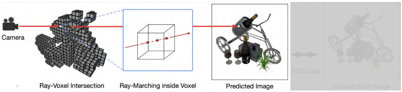
Positional encoding

14 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Volume Rendering with NSVF

Rendering NSVF is fast because it avoids sampling points in the empty space.

- Ray-voxel Intersection.
- Ray marching inside voxels.

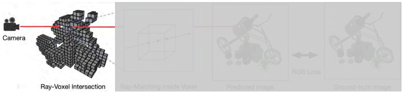


15 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Volume Rendering with NSVF

Ray-voxel Intersection

- Apply Axis Aligned Bounding Box (AABB) intersection test [Haines, 1989] for each ray.
- AABB is very efficient for NSVF, handling millions of ray-voxel intersections in real time.

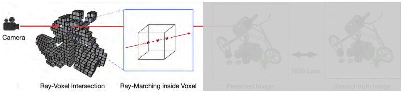


16 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Volume Rendering with NSVF

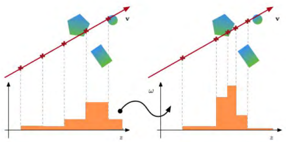
Ray Marching inside Voxels

- Uniformly sample points along the ray inside each intersected voxel, and evaluate NSVF to get the color and density of each sampled point.



17 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Volume Rendering with NSVF



Coarse sampling Importance sampling

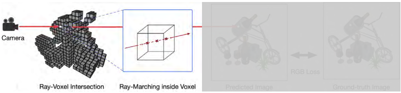
NeRF's sampling method

18 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Volume Rendering with NSVF

Early Termination

- Avoid taking unnecessary accumulation steps behind the surface;
- Stop evaluating points earlier when the accumulated densities close to 1



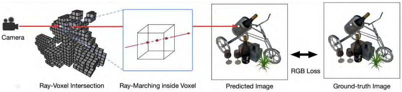
19 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Progressive Learning

- Because our rendering process is differentiable, the model can be trained end-to-end with 2D posed images as input for supervision.

$$\mathcal{L} = \sum_{(\boldsymbol{p}_0,\boldsymbol{v})\in R} \|\boldsymbol{C}(\boldsymbol{p}_0,\boldsymbol{v}) - \boldsymbol{C}^*(\boldsymbol{p}_0,\boldsymbol{v})\|_2^2 + \lambda \cdot \Omega\left(A(\boldsymbol{p}_0,\boldsymbol{v})\right)$$

Predicted color *Ground truth color*



20 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Progressive Learning

A progressive training strategy to learn NSVF from coarse to fine

- Voxel Initialization
- Self-Pruning
- Progressive Training

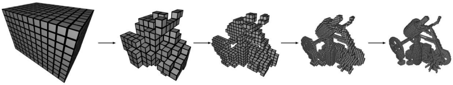


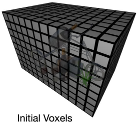
Illustration of self-pruning and progressive training

21 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Progressive Learning

Voxel Initialization

- The initial bounding box encloses the whole scene with sufficient margin. We eventually subdivide the bounding box into ~1000 voxels.
- If a coarse geometry is available, the initial voxels can also be initialized by voxelizing the coarse geometry.



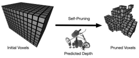
Initial Voxels

Progressive Learning

Self-Pruning

- We improve rendering efficiency by pruning "empty" voxels.
  - Determine whether a voxel is empty or not by checking the maximum predicted density from sampled points inside the voxel.

$$V_i \text{ is pruned } \textbf{if } \min_{j=1...G} \exp(-\underbrace{\boxed{\sigma}}_{\text{density}}(g_i(\boldsymbol{p}_j))) > \gamma, \quad \boldsymbol{p}_j \in V_i, V_i \in \mathcal{V},$$



  - Since this pruning process does not rely on other processing modules or input cues, we call it "*self-pruning*".

Progressive Learning

Progressive Training

- Self-pruning enables us to progressively allocate our resources.
- Progressive training:
  - Halve the size of voxels → Split each voxel into 8 sub-voxels.
  - Halve the size of ray marching steps.
  - The feature representations of the new vertices are initialized via trilinear interpolation of feature representations at the original eight voxel vertices.

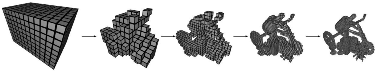


Illustration of self-pruning and progressive training

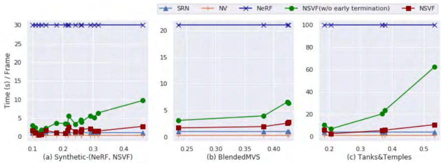
Results

Models	Synthetic-NeRF PSNR	Synthetic-NeRF SSIM	Synthetic-NeRF LPIPS	Synthetic-NSVF PSNR	Synthetic-NSVF SSIM	Synthetic-NSVF LPIPS	BlendedMVS PSNR	BlendedMVS SSIM	BlendedMVS LPIPS	Tanks and Temples PSNR	Tanks and Temples SSIM	Tanks and Temples LPIPS
SRN	22.26	0.846	0.170	24.33	0.882	0.141	20.51	0.770	0.294	24.10	0.847	0.251
NV	26.05	0.893	0.160	25.83	0.892	0.124	23.03	0.793	0.243	23.70	0.834	0.260
NeRF	31.01	0.947	0.081	30.81	0.952	0.043	24.15	0.828	0.192	25.78	0.864	0.198
NSVF^0	**31.75**	**0.954**	0.048	**35.18**	**0.979**	**0.015**	26.89	**0.898**	0.114	**28.48**	**0.901**	0.155
NSVF	31.74	0.953	**0.047**	35.13	**0.979**	**0.015**	**26.90**	**0.898**	**0.113**	28.40	0.900	**0.153**

*NSVF^0 is without early termination
*NSVF is executed with early termination ($\varepsilon = 0.01$)

25 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Results



x-axis: foreground to background ratio
y-axis: rendering time in second

*NSVF^0 is without early termination **(Green curve)**
*NSVF is executed with early termination ($\varepsilon = 0.01$) **(Red curve)**

26 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Results

Robot (From SyntheticNSVF dataset)

NeRF (Mildenhall et al. 2020)
(Rendering speed: 30 s/frame)

Ours (NSVF)
(Rendering speed: 0.6 s/frame)

27 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Graphics Vision & Video mpi max planck institut informatik FACEBOOK AI NUS National University of Singapore

Zoom-in & Zoom-out Effects

Graphics Vision & Video mpi max planck institut informatik FACEBOOK AI NUS National University of Singapore

Rendering of Dynamic Scenes

Normals of NSVF result

NSVF

(Input sequence from Fraunhofer Heinrich Hertz Institute)

Graphics Vision & Video mpi max planck institut informatik FACEBOOK AI NUS National University of Singapore

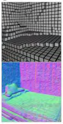
Rendering of Large-scale Indoor Scenes

Interactive camera control

Users' view (Rendered mesh)

NSVF

Sparse voxels



Normals

(Input sequence and 3D mesh from ScanNet [Dai et al. 2017])

Scene Editing and Composition

Interactive editing

NSVF

Main Limitation

- Real-time performance
  - Although our method is typically 10x faster than Nerf, it is still far from real time performance.
  - NeRF 0.06 FPS v.s. NSVF 1.1 FPS v.s. Real-time Rendering >25 FPS

Towards Real-time NeRF Rendering

Caching the Network Outputs with a Sparse Voxel Octree.

- The key idea is to use caching to trade memory for computational efficiency at inference time.

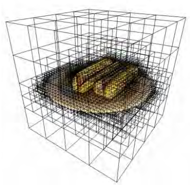


Image from [Yu et al., 2021]

Caching the Network Outputs with a Sparse Voxel Octree.

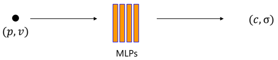
- The key idea is to use caching to trade memory for computational efficiency at inference time.

- There are three related papers:
- PlenOctrees for Real-time Rendering of Neural Radiance Fields, Yu et al., Arxiv 2021 ~200FPS
- FastNeRF: High-Fidelity Neural Rendering at 200FPS, Garbin et al., Arxiv 2021 ~200FPS
- Baking Neural Radiance Fields for Real-Time View Synthesis, Hedman et al., Arxiv 2021 ~84FPS

Caching the Network Outputs with a Sparse Voxel Octree.

- The key idea is to use caching to trade memory for computational efficiency at inference time.

- Specifically,
  (1) Train a NeRF-like network to predict density and color for each sampled point.



Caching the Network Outputs with a Sparse Voxel Octree.

- The key idea is to use caching to trade memory for computational efficiency at inference time.
- Specifically,
  (1) Train a NeRF-like network to predict density and color for each sampled point.
  (2) After training, extract the volumetric content and represent it using a sparse voxel Octree.
  (3) Precompute the network outputs for each octree leaf.

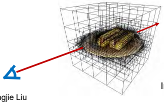
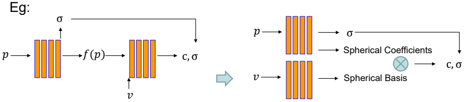


Image from [Yu et al., 2021]

Caching the Network Outputs with a Sparse Voxel Octree.

- Directly using the NeRF model for this caching operation is not feasible.
- Solution: Split NeRF's neural network into two separate networks: pose-dependent network and view-dependent network.
- Eg:



Original NeRF Network

Network Design in [Yu et al. 2021]

- A similar idea of learning basis functions is also used in: NeX: Real-time View Synthesis with Neural Basis Expansion. Wizadwongsa et al., CVPR 2021.

Using Multiple Shallow Networks

- A single high-capacity MLP for representing the entire scene can be replaced with thousands of small MLPs for the decomposed parts of the scene.
- KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs, Reiser et al., Arxiv 2021 ~13 FPS

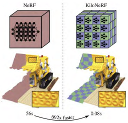


Image from [Reiser et al., 2021]

Using Multiple Shallow Networks

- A single high-capacity MLP for representing the entire scene can be replaced with thousands of small MLPs for the decomposed parts of the scene.
- The similar idea is also used in:

  DeRF: Decomposed Radiance Fields, Rebain et al., CVPR 2021 ~0.18 FPS

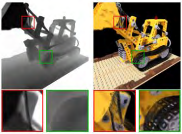
Image from [Rebain et al., 2021]

Graphics Vision & Video | mpii max planck institut informatik | FACEBOOK AI | NUS National University of Singapore

Depth-guided Sampling

- Predicting depths for more efficient sampling:

  DONeRF: Towards Real-Time Rendering of Neural Radiance Fields using Depth Oracle Networks, Neff et al., Arxiv 2021 ~15 FPS



(a) Depth Prediction (b) 3 Local Samples

Image from [Neff et al. 2021]

Graphics Vision & Video | mpii max planck institut informatik | FACEBOOK AI | NUS National University of Singapore

Learning Integral by a Neural Network

- A general framework to integrate signals with implicit neural representation, which can be used in volume rendering.

  AutoInt: Automatic Integration for Fast Neural Volume Rendering, Lindell et al., CVPR 2021. ~0.4 FPS

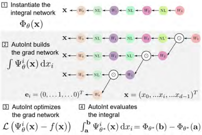


Image from [Lindell et al. 2021]

Graphics Vision & Video | mpii max planck institut informatik | FACEBOOK AI | NUS National University of Singapore

Related Work

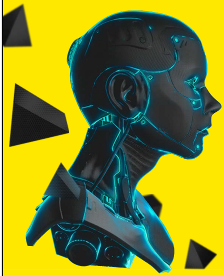
- NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, Mildenhall et al., ECCV 2020
- Neural Sparse Voxel Fields, Liu et al., NeurIPS 2020
- AutoInt: Automatic Integration for Fast Neural Volume Rendering, Lindell et al., CVPR 2021
- DeRF: Decomposed Radiance Fields, Rebain et al., CVPR 2021
- DONeRF: Towards Real-Time Rendering of Neural Radiance Fields using Depth Oracle Networks, Neff et al., Arxiv 2021
- FastNeRF: High-Fidelity Neural Rendering at 200FPS, Garbin et al., Arxiv 2021
- KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs, Reiser et al., Arxiv 2021
- PlenOctrees for Real-time Rendering of Neural Radiance Fields, Yu et al., Arxiv 2021
- Baking Neural Radiance Fields for Real-Time View Synthesis, Hedman et al., Arxiv 2021
- NeX: Real-time View Synthesis with Neural Basis Expansion. Wizadwongsa et al., CVPR 2021

43 Fast Rendering of Neural Radiance Fields, Lingjie Liu

Graphics Vision & Video | mpii max planck institut informatik | FACEBOOK AI | NUS National University of Singapore

Thank you

44 Fast Rendering of Neural Radiance Fields, Lingjie Liu



SIGGRAPH 2021

Advances in neural rendering

Towards instant 3D capture with GeLaTO and NeRFies

Dan B Goldman[1] and Keunhong Park[1,2]
[1]Google Research, [2]University of Washington

High quality 3D capture

SIGGRAPH 2021

- Expensive cameras
- Big rooms

©Imaging Research Center, UMBC

Towards instant 3D capture

SIGGRAPH 2021

GeLaTO:

- High-quality real-time rendering within category
- Key point: **Few-shot reconstruction with pretrained category models**

NeRFies:

- Casual capture using a handheld cell-phone camera
- Key point: **Modeling geometry and appearance of deforming objects**

SIGGRAPH 2021

GeLaTO: Generative Latent Textured Objects
ECCV'20

Ricardo Martin-Brualla, Rohit Pandey, Sofien Bouaziz, Matthew Brown, Dan B Goldman

THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

Neural Rendering

SIGGRAPH 2021

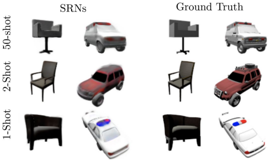
Category-level Modeling:

- Scene Representation Networks
- Texture Fields

Moderate image quality

Allow for few-shot reconstruction

Does not improve with more views!



Sitzmann et al., Scene Representation Networks, NeurIPS '19

THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

5

Objective

SIGGRAPH 2021

Model an entire object category

Use prior knowledge for high quality reconstruction

Representation suitable for real-time rendering

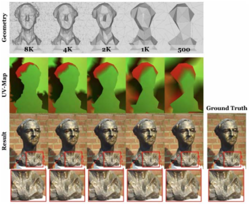
THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

6

Observation

SIGGRAPH 2021

Neural Textures are somewhat robust to coarse geometry!

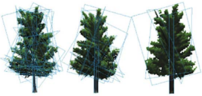


Thies et al., Deferred Neural Rendering: Image Synthesis using Neural Textures, SIGGRAPH'19

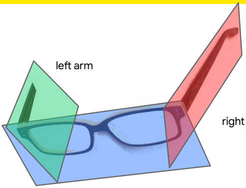
Neural Proxies

SIGGRAPH 2021

Describe geometry with 3 quads with neural textures attached



neural version of billboard clouds

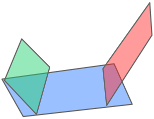


right arm

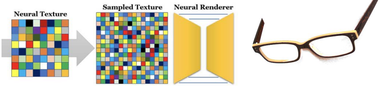
front

Neural Proxies

SIGGRAPH 2021



proxies & neural textures



Deferred Neural Rendering: Image Synthesis using Neural Textures, Thies et al, SIGGRAPH'19

Neural Proxies

SIGGRAPH 2021

Neural Texture needs to encode:

- Deviations from proxy's geometry
- View-dependent effects
- Occlusion between proxies

Eyewear as a test case

- Small (easy to capture)
- Lots of intra-category variation
- Interesting view-dependence

Eyeglasses Image Dataset

SIGGRAPH 2021

Captured a dataset of 576 poses of 85 eyeglasses

Combination of difference matting and Laplacian matting to extract clean mattes

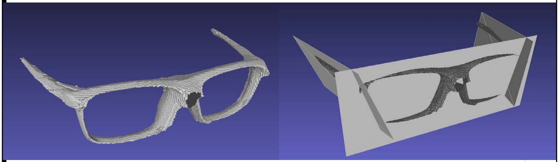
captures 4 conditions per pose computed RGB and alpha

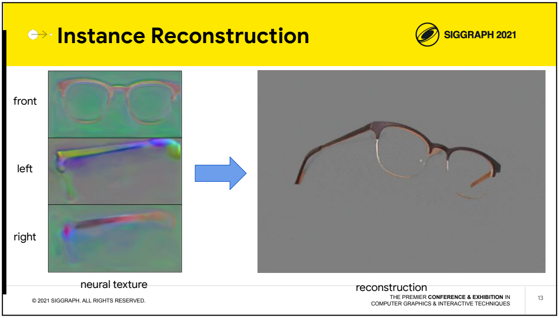
Proxy Generation

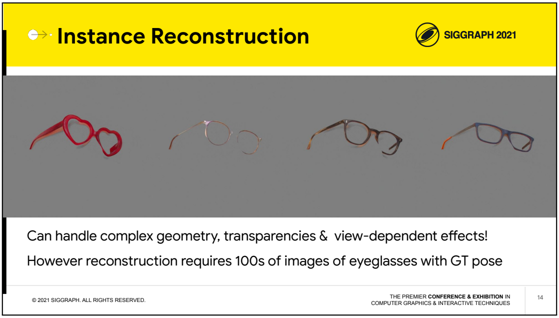
SIGGRAPH 2021

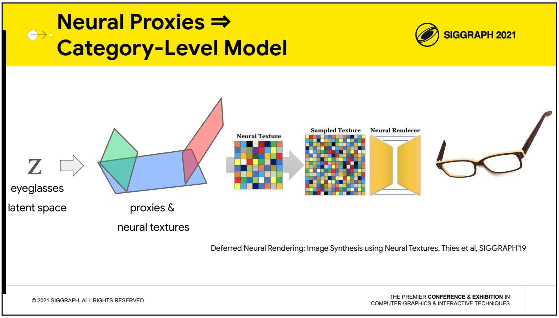
Compute rough voxelized shape using voxel carving

Approximating a plane to medial axis using a ROI



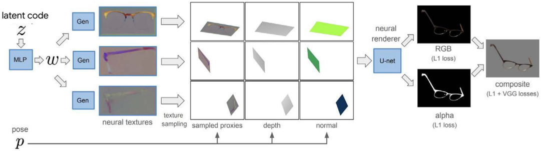

Instance Reconstruction
SIGGRAPH 2021
front
left
right
neural texture
reconstruction
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
13


Instance Reconstruction
SIGGRAPH 2021
Can handle complex geometry, transparencies & view-dependent effects!
However reconstruction requires 100s of images of eyeglasses with GT pose
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
14


Neural Proxies ⇒
Category-Level Model
SIGGRAPH 2021
Z
eyeglasses
latent space
proxies &
neural textures
Neural Texture
Sampled Texture
Neural Renderer
Deferred Neural Rendering: Image Synthesis using Neural Textures, Thies et al, SIGGRAPH'19
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

Architecture

SIGGRAPH 2021

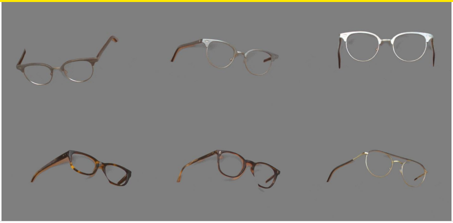


THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

16

Neural Billboards Interpolations

SIGGRAPH 2021



THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

17

Few-shot Reconstruction

SIGGRAPH 2021

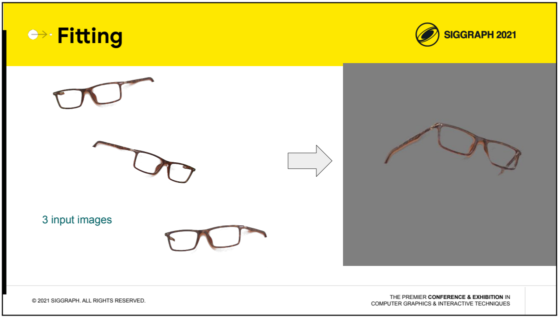
To reconstruct an **unseen instance**:

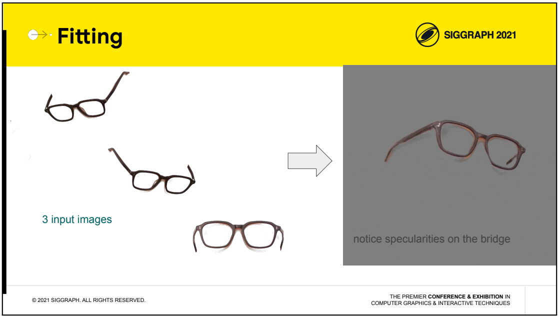
- Regress latent code z
- Regress transformed latent code w
- **Optimize neural textures**
- **+ fine-tune network parameters**

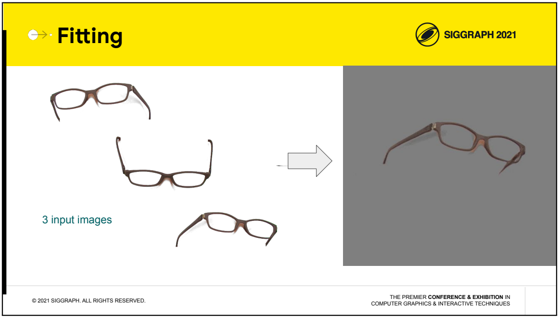
3 input views

THE PREMIER **CONFERENCE & EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

18


Fitting
SIGGRAPH 2021
3 input images
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


Fitting
SIGGRAPH 2021
3 input images
notice specularities on the bridge
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


Fitting
SIGGRAPH 2021
3 input images
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

Comparisons

SIGGRAPH 2021

Latent space model:

- Variational Autoencoder (**VAE**)
  - Encoder is not used after training category model
- Generative Latent Optimization (**GLO**)
  - Latent codes are randomly initialized and optimized during training

Proxy Compositing:

- **stack:** rendered proxies are stacked and passed to the neural renderer
- **z-buffer:** rendered proxies composited using depth before neural renderer

Instance Reconstruction

SIGGRAPH 2021

	view interpolation		
Model	VAE	GLO	GLO
Composite	stack	z-buffering	stack
PSNR	39.70	41.21	41.32
$PSNR_M$	21.79	23.29	23.42
SSIM	0.9897	0.9916	0.9917
Mask IoU	0.9379	0.9556	0.9556

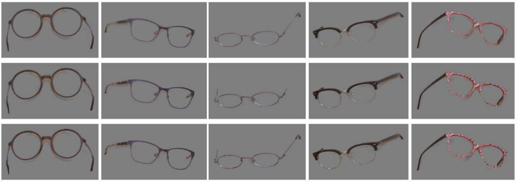
Instance Reconstruction

SIGGRAPH 2021

VAE

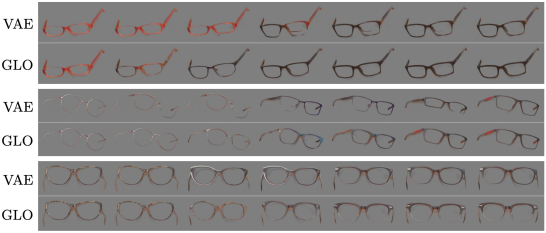
GLO

GT



Instance Interpolation

SIGGRAPH 2021



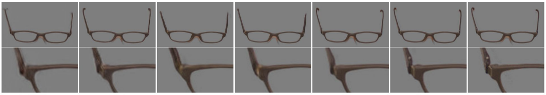
Few-Shot Reconstruction

SIGGRAPH 2021

Model Composite	VAE stack	GLO z-buffering	GLO stack
	few-shot reconstruction		
PSNR	35.59	36.14	37.19
$PSNR_M$	17.94	18.65	19.64
SSIM	0.9793	0.9819	0.9842
Mask IoU	0.8686	0.8725	0.9012

Few-Shot vs Instance Reconstruction

SIGGRAPH 2021



$N = 30$	$N = 100$	$N = 3$	$N = 10$	$N = 30$	$N = 100$	GT
optimized from scratch		finetuning category-level model				

Few-Shot vs Instance Reconstruction

SIGGRAPH 2021

Better reconstruction with >3x fewer views

	optimized from scratch		finetuning category model			
Input images	30	100	3	10	30	100
PSNR	38.75	40.05	36.53	39.35	41.61	43.42
$PSNR_M$	21.48	22.43	19.01	21.78	24.00	25.80
SSIM	0.9858	0.9897	0.9824	0.9890	0.9921	0.9942
Mask IoU	0.9293	0.9407	0.8864	0.9350	0.9585	0.9682

28

Few-Shot vs Instance Reconstruction

SIGGRAPH 2021

Better reconstruction with >3x fewer views

	optimized from scratch		finetuning category model			
Input images	30	100	3	10	30	100
PSNR	38.75	40.05	36.53	39.35	41.61	43.42
$PSNR_M$	21.48	22.43	19.01	21.78	24.00	25.80
SSIM	0.9858	0.9897	0.9824	0.9890	0.9921	0.9942
Mask IoU	0.9293	0.9407	0.8864	0.9350	0.9585	0.9682

29

Few-Shot vs Instance Reconstruction

SIGGRAPH 2021

Better reconstruction with >3x fewer views

	optimized from scratch		finetuning category model			
Input images	30	100	3	10	30	100
PSNR	38.75	40.05	36.53	39.35	41.61	43.42
$PSNR_M$	21.48	22.43	19.01	21.78	24.00	25.80
SSIM	0.9858	0.9897	0.9824	0.9890	0.9921	0.9942
Mask IoU	0.9293	0.9407	0.8864	0.9350	0.9585	0.9682

30

Few-shot Reconstruction: what to fit?

SIGGRAPH 2021

Fit variables	z	w	texture	all
PSNR	31.30	36.50	37.12	**37.19**
$PSNR_M$	13.85	18.85	37.12	**19.64**
SSIM	0.9638	0.9833	0.9841	**0.9842**
Mask IoU	0.7242	**0.9152**	0.8984	0.9012

latent code z | latent code w | neural texture | neural texture + fine-tune | reference

Extending Proxies

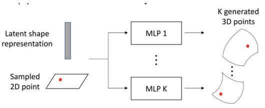
SIGGRAPH 2021

Not constrained to simple proxy geometry

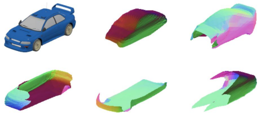
Only requirement is to have a UV map of the proxies

Free-form Proxies

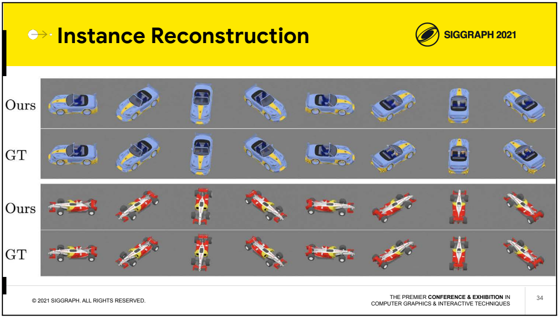
SIGGRAPH 2021

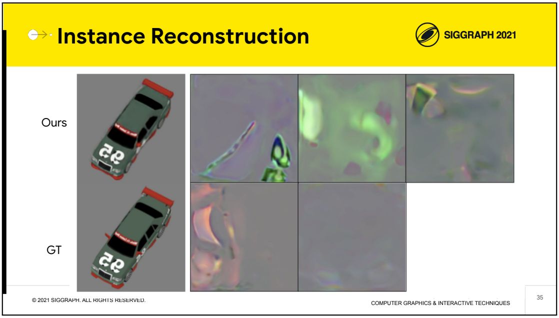


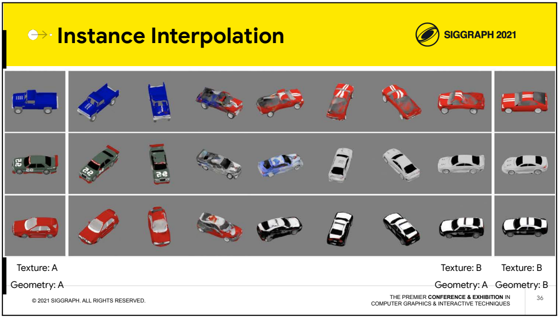
K-patch AtlasNet model



free-form proxies using AtlasNet


Instance Reconstruction
SIGGRAPH 2021
Ours
GT
Ours
GT
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
34

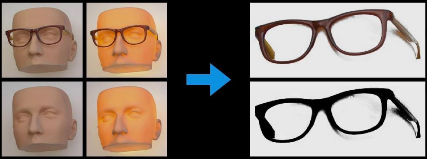

Instance Reconstruction
SIGGRAPH 2021
Ours
GT
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
35


Instance Interpolation
SIGGRAPH 2021
Texture: A
Geometry: A
Texture: B
Geometry: A
Texture: B
Geometry: B
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
36

Instance Interpolation

SIGGRAPH 2021

Eyeglasses Frames Dataset



We capture 4 images per pose: object / no object, backlight on / off

We use difference matting on backlit image to extract alpha, then solve for color in image without backlight.

Limitations

SIGGRAPH 2021

Few-shot reconstruction assumes GT pose & proxy geometry:

⇒ Train with pose and geometry estimation in the loop?

Lighting is baked into the model

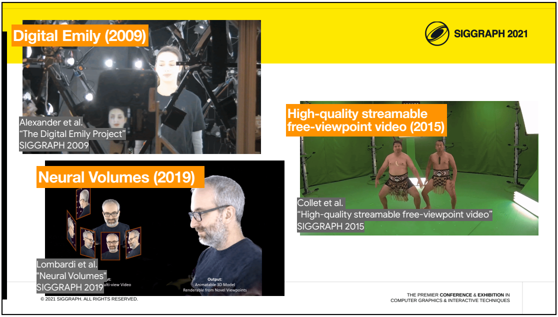
⇒ Capture in lightstage & add latent code for lighting?

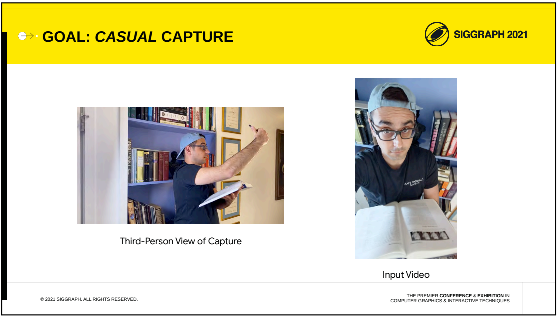
Takeaways

SIGGRAPH 2021

- **GeLaTO**: appearance representation for object category modeling
- Neural textures accurately reproduce appearance of thin structures and reflective materials
- Category model generates plausible instance interpolations and enables few-shot reconstruction

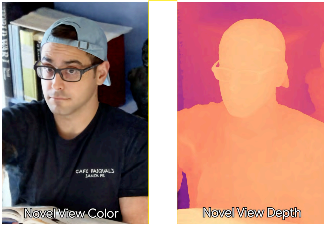

SIGGRAPH 2021
NERFIES: DEFORMABLE NEURAL RADIANCE FIELDS
Keunhong Park
Utkarsh Sinha
Jon Barron
Sofien Bouaziz
Dan Goldman
Steve Seitz
Ricardo Martin-Brualla
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


Digital Emily (2009)
Alexander et al.
"The Digital Emily Project"
SIGGRAPH 2009
SIGGRAPH 2021
High-quality streamable free-viewpoint video (2015)
Collet et al.
"High-quality streamable free-viewpoint video"
SIGGRAPH 2015
Neural Volumes (2019)
Lombardi et al.
"Neural Volumes"
SIGGRAPH 2019
Output:
Animatable 3D Model
Renderable from Novel Viewpoints
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


GOAL: CASUAL CAPTURE
SIGGRAPH 2021
Third-Person View of Capture
Input Video
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

GOAL: *FREE-VIEWPOINT* CAPTURE

SIGGRAPH 2021



THE PREMIER **CONFERENCE** & **EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

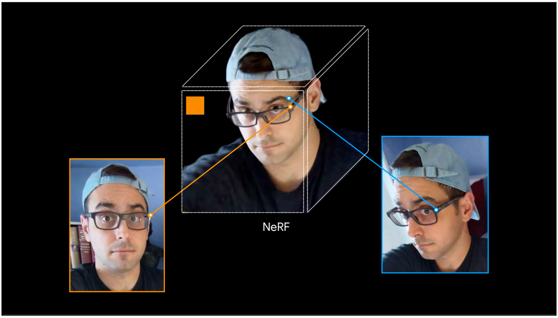
LET'S RUN NERF

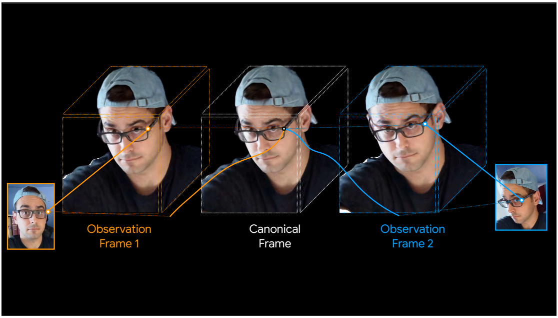
SIGGRAPH 2021

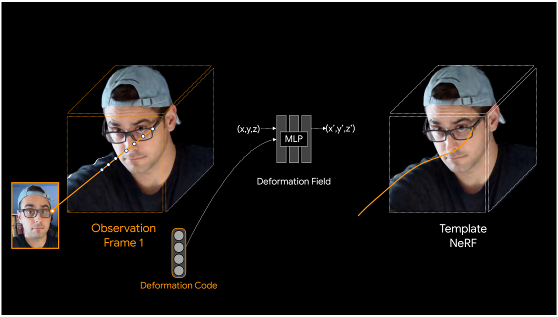
NEURAL RADIANCE FIELDS

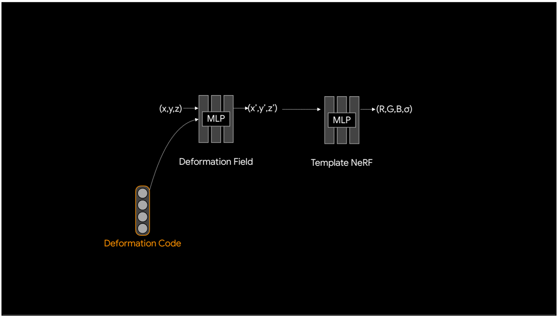
- "*NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*", Mildenhall et al, ECCV'20
- Does not work for moving subjects.

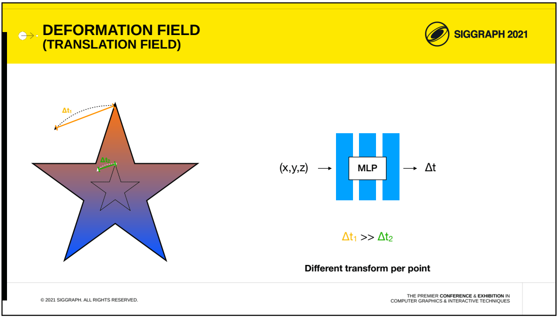
THE PREMIER **CONFERENCE** & **EXHIBITION** IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

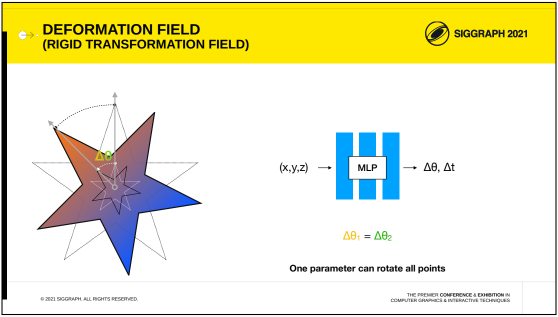


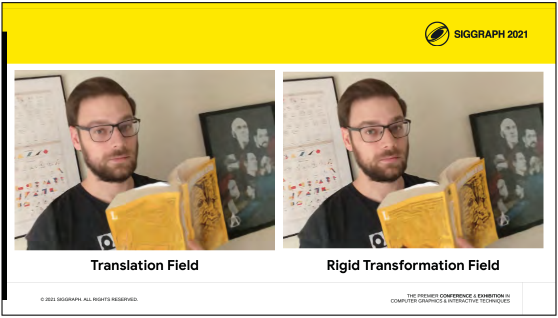

Observation
Frame 1
Canonical
Frame
Observation
Frame 2


(x,y,z)
MLP
(x',y',z')
Deformation Field
Observation
Frame 1
Deformation Code
Template
NeRF


(x,y,z)
MLP
(x',y',z')
MLP
(R,G,B,σ)
Deformation Field
Template NeRF
Deformation Code


DEFORMATION FIELD
(TRANSLATION FIELD)
SIGGRAPH 2021
Δt₁
Δt₂
(x,y,z)
MLP
Δt
Δt₁ >> Δt₂
Different transform per point
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


DEFORMATION FIELD
(RIGID TRANSFORMATION FIELD)
SIGGRAPH 2021
Δθ
(x,y,z)
MLP
Δθ, Δt
Δθ₁ = Δθ₂
One parameter can rotate all points
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


SIGGRAPH 2021
Translation Field
Rigid Transformation Field
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

UNCONSTRAINED DEFORMATIONS

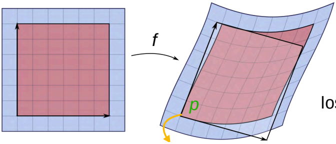
SIGGRAPH 2021

Input Video | Novel-View Color | Novel-View Depth

ELASTIC REGULARIZATION (AS-RIGID-AS-POSSIBLE)

SIGGRAPH 2021

Keep f as close to a rotation as possible
→ Penalize any ‘stretch’ (volume preservation)
→ Keep singular values of J close to 1.0



SVD: $J = U\Sigma V$

$$\text{loss} = ||\log(\Sigma) - \log(I)||_F$$

Jacobian J
Best linear approximation of f at p

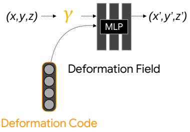
ELASTIC REGULARIZATION

SIGGRAPH 2021

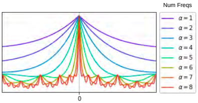
No Elastic Regularization | With Elastic Regularization

THE POSITIONAL ENCODING

SIGGRAPH 2021



$$\gamma(\mathbf{x}) = [\mathbf{x}, \cdots, \sin(2^k \pi \mathbf{x}), \cos(2^k \pi \mathbf{x}), \cdots]^T$$



Interpolating Kernels

Tancik et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”, NeurIPS 2020

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

16

A TRADEOFF

SIGGRAPH 2021

Ground Truth

4 Frequencies

8 Frequencies

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

17

A TRADEOFF

SIGGRAPH 2021

Ground Truth

4 Frequencies

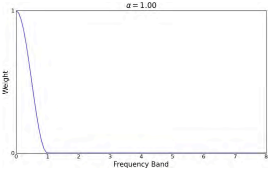
8 Frequencies

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

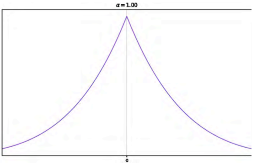
18

COARSE-TO-FINE OPTIMIZATION

SIGGRAPH 2021



Frequency Band Weights



Neural-Tangent Kernel

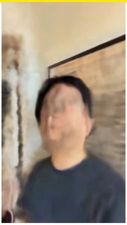
COARSE-TO-FINE OPTIMIZATION

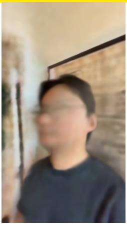
SIGGRAPH 2021

COARSE-TO-FINE OPTIMIZATION

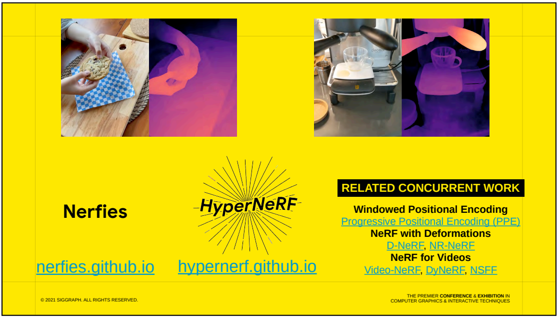
SIGGRAPH 2021

Training Progression w/o Coarse-to-Fine





Training Progression with Coarse-to-Fine


Nerfies
HyperNeRF
nerfies.github.io
hypernerf.github.io
RELATED CONCURRENT WORK
Windowed Positional Encoding
Progressive Positional Encoding (PPE)
NeRF with Deformations
D-NeRF, NR-NeRF
NeRF for Videos
Video-NeRF, DyNeRF, NSFF
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

SIGGRAPH 2021

Relightable and Editable Neural Rendering

Zexiang Xu

Adobe Research

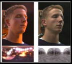
(SIGGRAPH 2021 Course: Advances in Neural Rendering)

Outline

- Relighting and appearance acquisition overview
  - Image-based methods
  - Reconstruction methods
- Volumetric neural rendering
  - Relighting (Neural Reflectance Fields)
  - Editing (Neural Texture Mapping)
- Challenges / opportunities

Appearance acquisition

Captured Images → Appearance Modeling → Rendering


[Debevec et al. 2000]


[Ren et al. 2015]


[Xu et al. 2018]


[Kang et al. 2019]

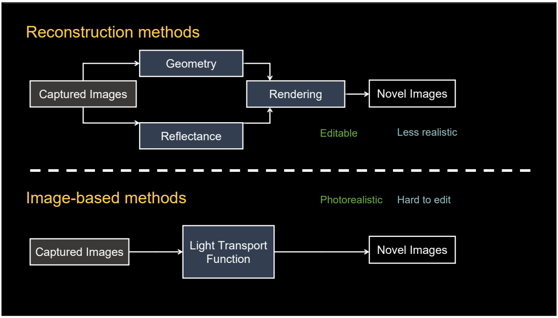

[Matusik et al. 2002]


[Xia et al. 2016]


[Li et al. 2018]


[Bi et al. 2020]

- Changing viewpoints (view synthesis)
- Changing lighting (relighting)
- Scene editing


Reconstruction methods
Captured Images
Geometry
Reflectance
Rendering
Novel Images
Editable
Less realistic
Image-based methods
Photorealistic
Hard to edit
Captured Images
Light Transport Function
Novel Images

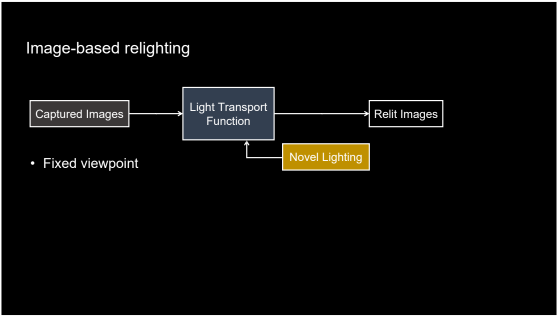
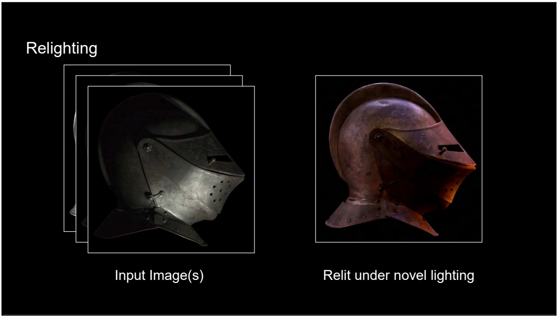
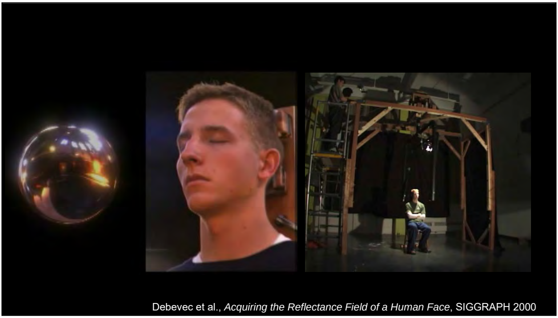


Image-based relighting
Captured Images
Light Transport Function
Relit Images
Novel Lighting
• Fixed viewpoint


Relighting
Input Image(s)
Relit under novel lighting


Debevec et al., *Acquiring the Reflectance Field of a Human Face*, SIGGRAPH 2000

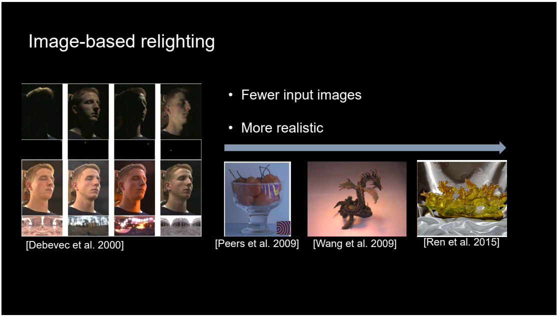
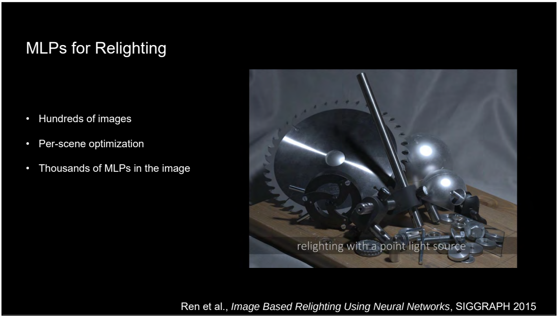
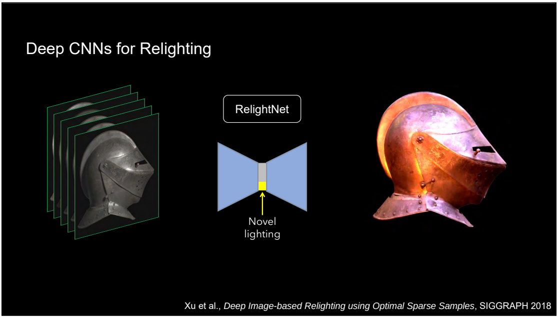
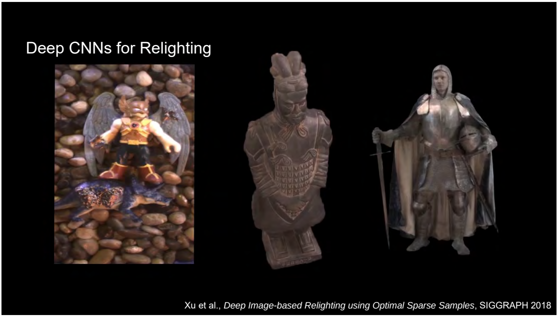
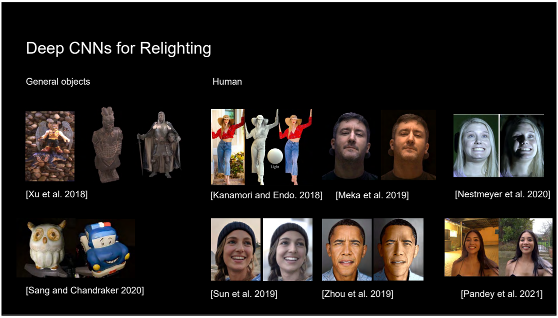


Image-based relighting
• Fewer input images
• More realistic
[Debevec et al. 2000]
[Peers et al. 2009]
[Wang et al. 2009]
[Ren et al. 2015]


MLPs for Relighting
• Hundreds of images
• Per-scene optimization
• Thousands of MLPs in the image
relighting with a point light source
Ren et al., *Image Based Relighting Using Neural Networks*, SIGGRAPH 2015

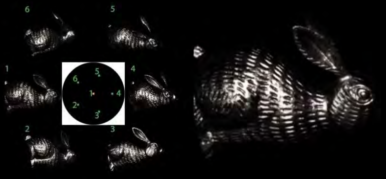

Deep CNNs for Relighting
RelightNet
Novel lighting
Xu et al., Deep Image-based Relighting using Optimal Sparse Samples, SIGGRAPH 2018


Deep CNNs for Relighting
Xu et al., Deep Image-based Relighting using Optimal Sparse Samples, SIGGRAPH 2018


Deep CNNs for Relighting
General objects
Human
[Xu et al. 2018]
[Kanamori and Endo. 2018]
[Meka et al. 2019]
[Nestmeyer et al. 2020]
[Sang and Chandraker 2020]
[Sun et al. 2019]
[Zhou et al. 2019]
[Pandey et al. 2021]

Can we change the viewpoint?

Relighting + view synthesis

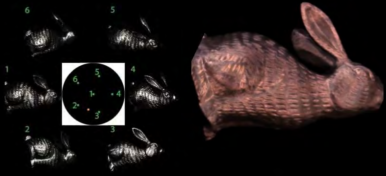


Captured images

Novel view synthesis

Xu et al., *Deep View Synthesis from Sparse Photometric Images*, SIGGRAPH 2019

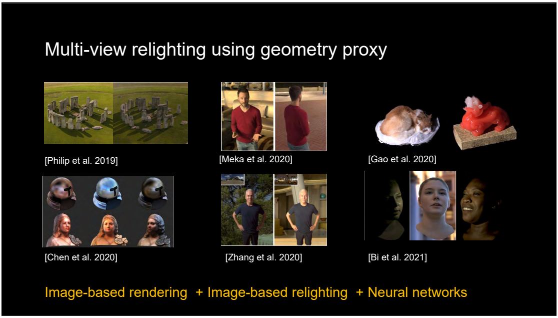
Relighting + view synthesis

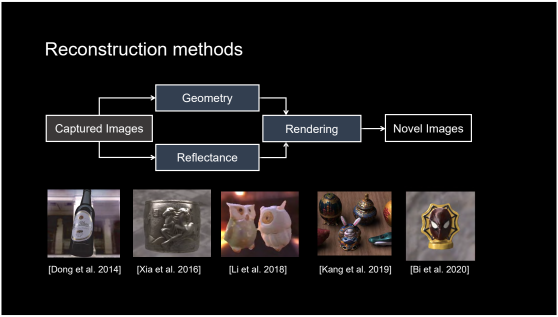


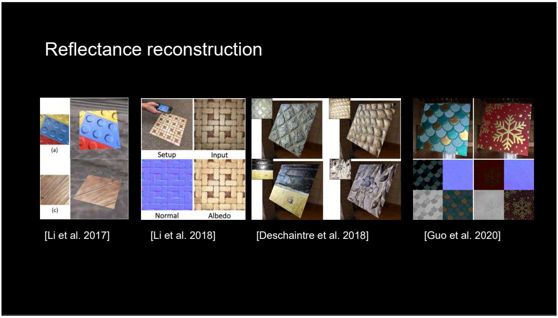
Captured images

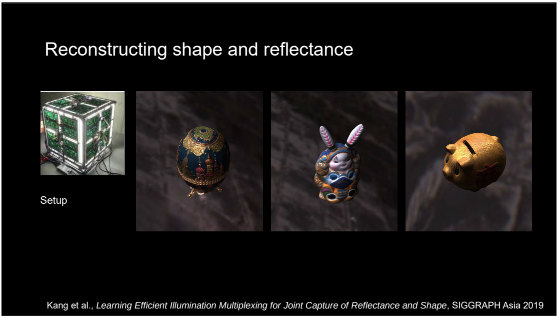
Novel view synthesis
+ relighting

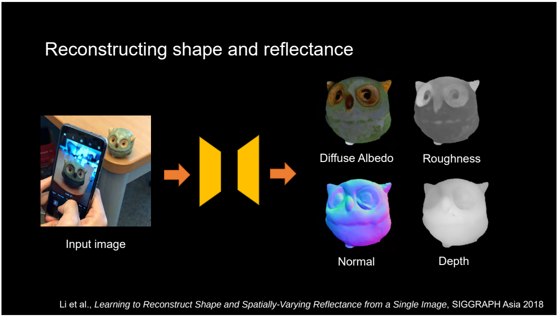
Xu et al., *Deep View Synthesis from Sparse Photometric Images*, SIGGRAPH 2019

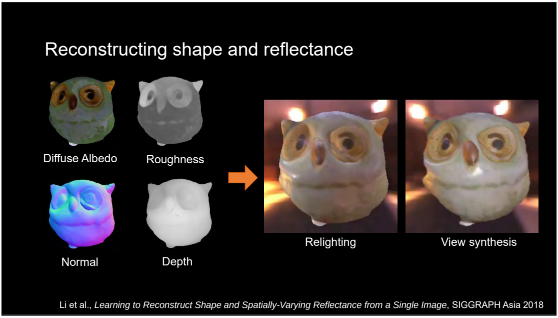

Multi-view relighting using geometry proxy
[Philip et al. 2019]
[Meka et al. 2020]
[Gao et al. 2020]
[Chen et al. 2020]
[Zhang et al. 2020]
[Bi et al. 2021]
Image-based rendering + Image-based relighting + Neural networks

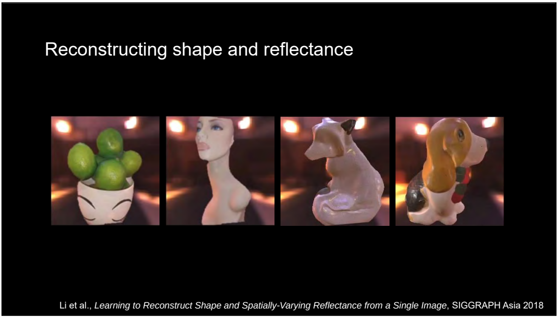

Reconstruction methods
Geometry
Captured Images
Rendering
Novel Images
Reflectance
[Dong et al. 2014]
[Xia et al. 2016]
[Li et al. 2018]
[Kang et al. 2019]
[Bi et al. 2020]

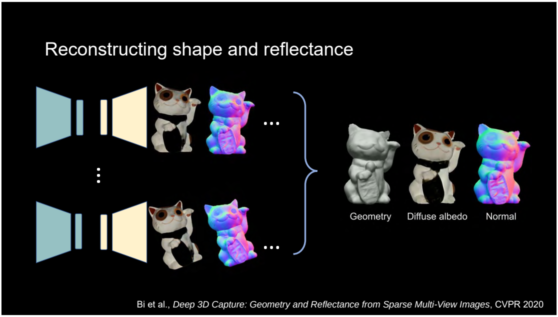

Reflectance reconstruction
(a)
(c)
Setup
Input
Normal
Albedo
[Li et al. 2017]
[Li et al. 2018]
[Deschaintre et al. 2018]
[Guo et al. 2020]

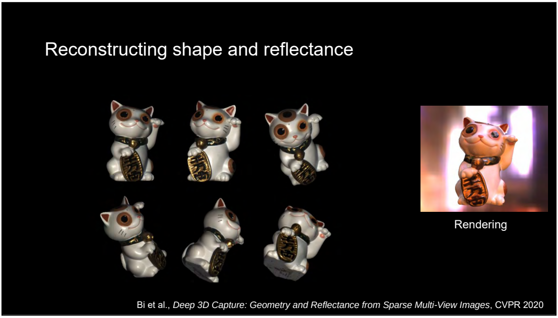

Reconstructing shape and reflectance
Setup
Kang et al., *Learning Efficient Illumination Multiplexing for Joint Capture of Reflectance and Shape*, SIGGRAPH Asia 2019


Reconstructing shape and reflectance
Input image
Diffuse Albedo
Roughness
Normal
Depth
Li et al., *Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image*, SIGGRAPH Asia 2018


Reconstructing shape and reflectance
Diffuse Albedo
Roughness
Normal
Depth
Relighting
View synthesis
Li et al., *Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image*, SIGGRAPH Asia 2018

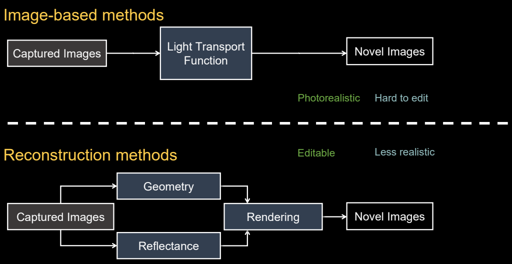

Reconstructing shape and reflectance
Li et al., *Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image*, SIGGRAPH Asia 2018


Reconstructing shape and reflectance
Geometry
Diffuse albedo
Normal
Bi et al., *Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images*, CVPR 2020


Reconstructing shape and reflectance
Rendering
Bi et al., *Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images*, CVPR 2020

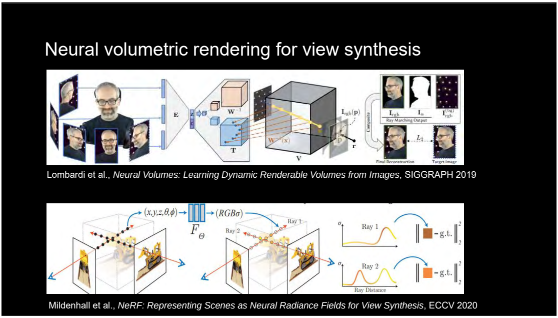
Reconstructing shape and reflectance

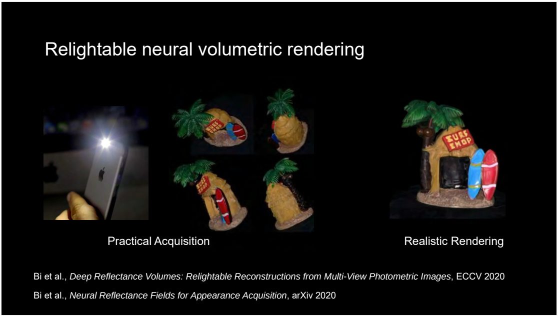
Bi et al., *Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images*, CVPR 2020

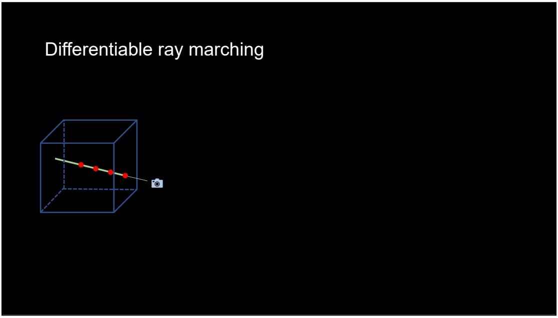


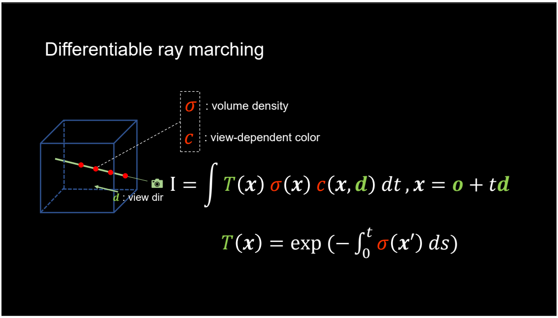
Reconstruction methods

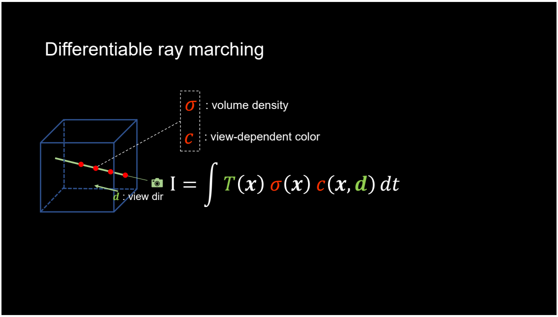
- Joint optimization of geometry and reflectance
- Meshes are hard to differentiate in rendering
- Differentiable volume rendering (ray marching)

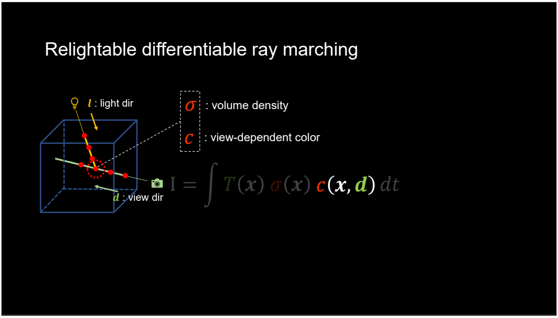

Neural volumetric rendering for view synthesis
Lombardi et al., *Neural Volumes: Learning Dynamic Renderable Volumes from Images*, SIGGRAPH 2019
Mildenhall et al., *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, ECCV 2020

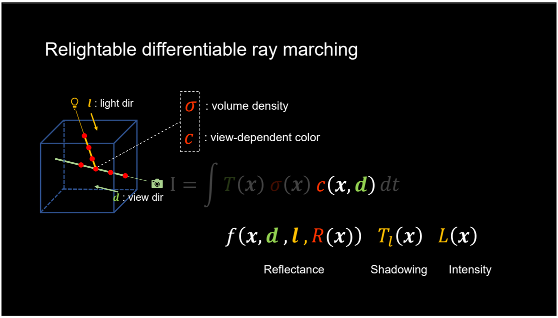

Relightable neural volumetric rendering
Practical Acquisition
Realistic Rendering
Bi et al., *Deep Reflectance Volumes: Relightable Reconstructions from Multi-View Photometric Images*, ECCV 2020
Bi et al., *Neural Reflectance Fields for Appearance Acquisition*, arXiv 2020

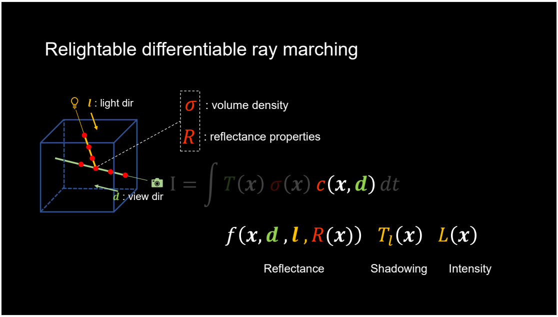

Differentiable ray marching

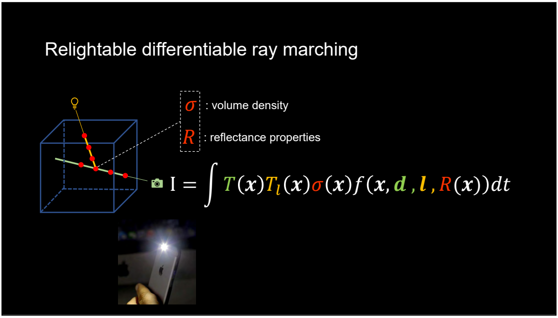

Differentiable ray marching
σ : volume density
c : view-dependent color
$\boldsymbol{d}$: view dir
$\mathrm{I} = \int T(\boldsymbol{x})\, \sigma(\boldsymbol{x})\, c(\boldsymbol{x}, \boldsymbol{d})\, dt\,, \boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{d}$
$T(\boldsymbol{x}) = \exp\left(-\int_0^t \sigma(\boldsymbol{x}')\, ds\right)$

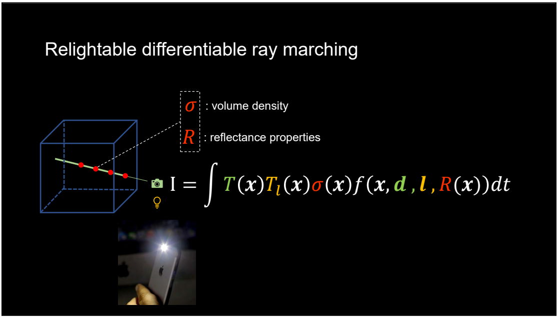

Differentiable ray marching
σ : volume density
c : view-dependent color
$\boldsymbol{d}$: view dir
$\mathrm{I} = \int T(\boldsymbol{x})\, \sigma(\boldsymbol{x})\, c(\boldsymbol{x}, \boldsymbol{d})\, dt$

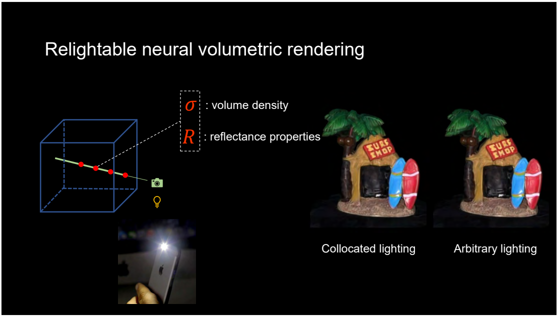

Relightable differentiable ray marching
$\boldsymbol{l}$: light dir
σ : volume density
c : view-dependent color
$\boldsymbol{d}$: view dir
$\mathrm{I} = \int T(\boldsymbol{x})\, \sigma(\boldsymbol{x})\, c(\boldsymbol{x}, \boldsymbol{d})\, dt$

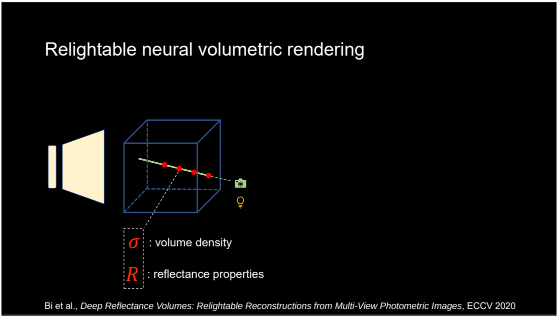

Relightable differentiable ray marching
l : light dir
σ : volume density
c : view-dependent color
d : view dir
$I = \int T(x)\,\sigma(x)\,c(x, d)\,dt$
$f(x, d, l, R(x))$ $T_l(x)$ $L(x)$
Reflectance
Shadowing
Intensity

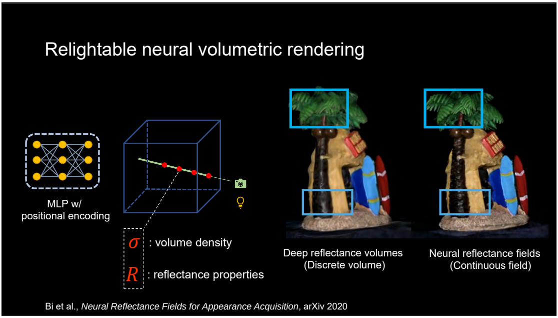

Relightable differentiable ray marching
l : light dir
σ : volume density
R : reflectance properties
d : view dir
$I = \int T(x)\,\sigma(x)\,c(x, d)\,dt$
$f(x, d, l, R(x))$ $T_l(x)$ $L(x)$
Reflectance
Shadowing
Intensity


Relightable differentiable ray marching
σ : volume density
R : reflectance properties
$I = \int T(x) T_l(x) \sigma(x) f(x, d, l, R(x)) dt$

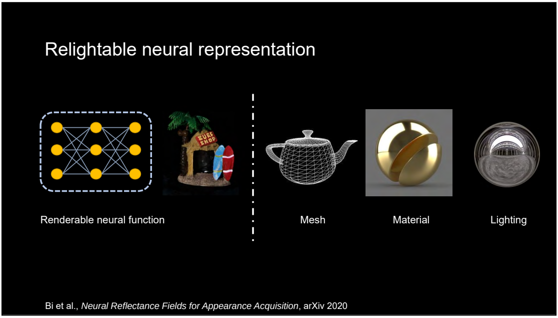

Relightable differentiable ray marching
σ : volume density
R : reflectance properties
$I = \int T(\boldsymbol{x}) T_l(\boldsymbol{x}) \sigma(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{l}, R(\boldsymbol{x})) dt$

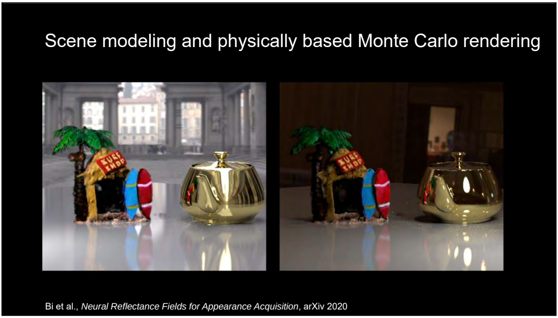

Relightable neural volumetric rendering
σ : volume density
R : reflectance properties
Collocated lighting
Arbitrary lighting

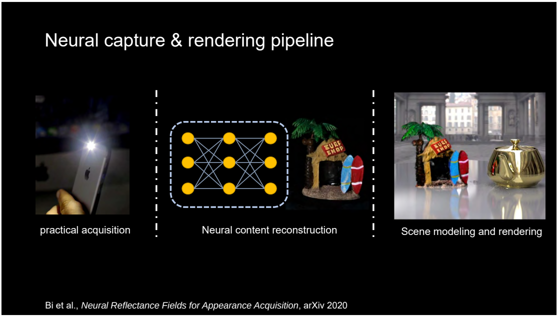

Relightable neural volumetric rendering
σ : volume density
R : reflectance properties
Bi et al., *Deep Reflectance Volumes: Relightable Reconstructions from Multi-View Photometric Images*, ECCV 2020

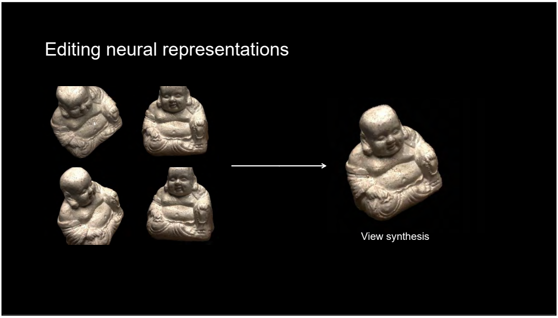

Relightable neural volumetric rendering
MLP w/
positional encoding
σ : volume density
R : reflectance properties
Deep reflectance volumes
(Discrete volume)
Neural reflectance fields
(Continuous field)
Bi et al., Neural Reflectance Fields for Appearance Acquisition, arXiv 2020

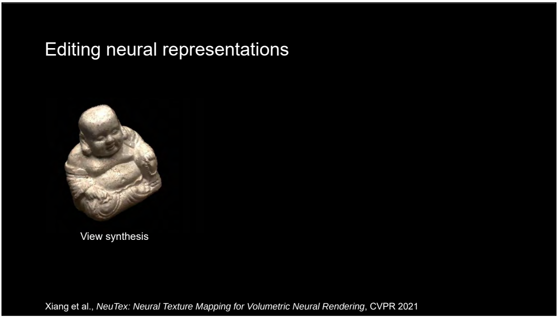

Relightable neural volumetric rendering
Bi et al., Neural Reflectance Fields for Appearance Acquisition, arXiv 2020

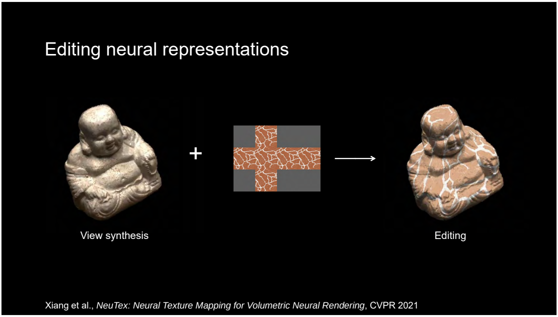

Relightable neural representation
Renderable neural function
Mesh
Material
Lighting
Bi et al., Neural Reflectance Fields for Appearance Acquisition, arXiv 2020

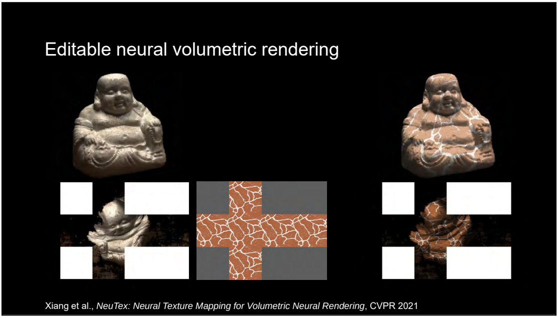

Scene modeling and physically based Monte Carlo rendering
Bi et al., *Neural Reflectance Fields for Appearance Acquisition*, arXiv 2020


Neural capture & rendering pipeline
practical acquisition
Neural content reconstruction
Scene modeling and rendering
Bi et al., *Neural Reflectance Fields for Appearance Acquisition*, arXiv 2020


Editing neural representations
View synthesis


Editing neural representations
View synthesis
Xiang et al., *NeuTex: Neural Texture Mapping for Volumetric Neural Rendering*, CVPR 2021


Editing neural representations
+
View synthesis
Editing
Xiang et al., *NeuTex: Neural Texture Mapping for Volumetric Neural Rendering*, CVPR 2021


Editable neural volumetric rendering
Xiang et al., *NeuTex: Neural Texture Mapping for Volumetric Neural Rendering*, CVPR 2021

Editable neural volumetric rendering

Xiang et al., *NeuTex: Neural Texture Mapping for Volumetric Neural Rendering*, CVPR 2021

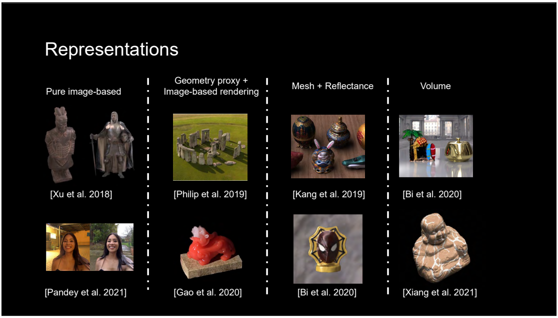
Relightable and editable volumetric rendering

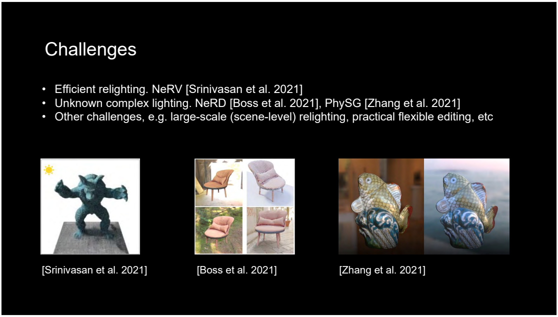
Bi et al., *Deep Reflectance Volumes: Relightable Reconstructions from Multi-View Photometric Images*, ECCV 2020
Bi et al., *Neural Reflectance Fields for Appearance Acquisition*, arXiv 2020

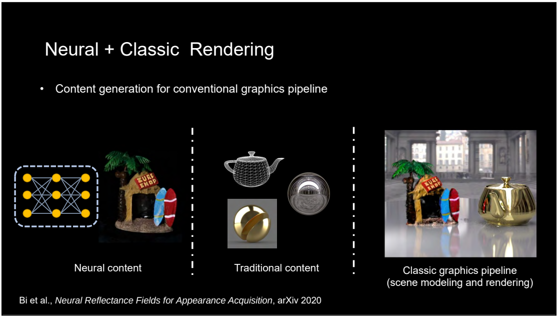
Xiang et al., *NeuTex: Neural Texture Mapping for Volumetric Neural Rendering*, CVPR 2021

Conclusion, challenges, opportunities

- Representations
  - Pure image-based modeling
  - Image-based rendering (geometry proxy) + relighting
  - Mesh + BRDFs
  - Volumetric representation


Representations
Pure image-based
Geometry proxy + Image-based rendering
Mesh + Reflectance
Volume
[Xu et al. 2018]
[Philip et al. 2019]
[Kang et al. 2019]
[Bi et al. 2020]
[Pandey et al. 2021]
[Gao et al. 2020]
[Bi et al. 2020]
[Xiang et al. 2021]


Challenges
• Efficient relighting. NeRV [Srinivasan et al. 2021]
• Unknown complex lighting. NeRD [Boss et al. 2021], PhySG [Zhang et al. 2021]
• Other challenges, e.g. large-scale (scene-level) relighting, practical flexible editing, etc
[Srinivasan et al. 2021]
[Boss et al. 2021]
[Zhang et al. 2021]


Neural + Classic Rendering
• Content generation for conventional graphics pipeline
Neural content
Traditional content
Classic graphics pipeline
(scene modeling and rendering)
Bi et al., *Neural Reflectance Fields for Appearance Acquisition*, arXiv 2020

Neural + Classic Rendering

- Content generation for conventional graphics pipeline
- Improve conventional graphics, e.g. material modeling [Kuznetsov et al. 2021]

Neural + Classic Rendering

Kuznetsov et al., *NeuMIP: Multi-Resolution Neural Materials*, SIGGRAPH 2021

Conclusion, challenges, opportunities

- Representations
  - Pure image-based modeling
  - Image-based rendering (geometry proxy) + relighting
  - Mesh + BRDFs
  - Volumetric representation
- Challenges
  - Efficient relighting. NeRV [Srinivasan et al. 2021]
  - Unknown complex lighting. NeRD [Boss et al. 2021], PhySG [Zhang et al. 2021]
  - Large-scale (scene-level) relighting, practical flexible editing
- Neural + Classic Rendering
  - Content generation for conventional graphics pipeline
  - Improve conventional graphics, material modeling [Kuznetsov et al. 2021]

Total Relighting:
Learning to Relight Portraits for Background Replacement

Rohit Pandey*, **Sergio Orts-Escolano***, Chloe LeGendre*, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello

Google Research

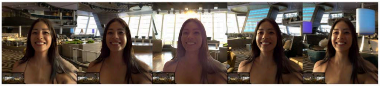
Motivation

Neural rendering for relighting

How to change the illumination in renderings, photographs with the help of machine learning

Novel approaches that combine classical computer graphics pipelines and learnable components

Generating photo-realistic imagery

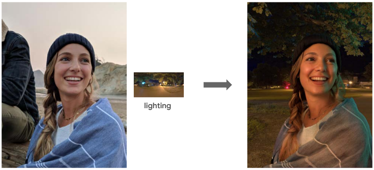


Objective

Given an arbitrary in-the-wild tone-mapped portrait image, we want to relight the subject and realistically composite them into a new background.

Problem statement

Inputs:
In-the-wild, LDR RGB portrait
High resolution (16k), HDR panoramic lighting environment



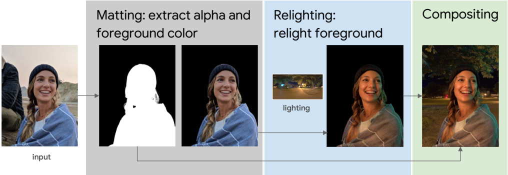
Previous work

Input image and estimated lighting

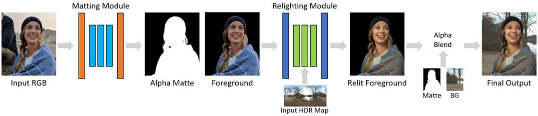
Rendered images under three novel illuminations

Sun et al. “Single Image Portrait Relighting”, SIGGRAPH 2019

Core components: Matting, Relighting, Compositing

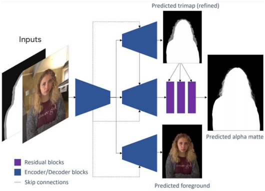


Core components: Matting, Relighting and Compositing



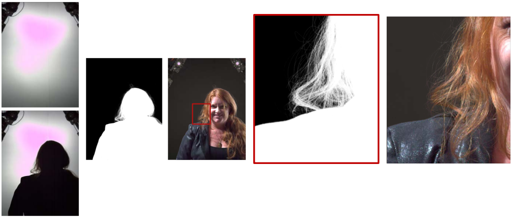
Deep Matting Module - Design

Convolutional neural network for foreground and alpha prediction from portraits.



Light Stage Dataset - Matting Ground Truth

Ratio Matting



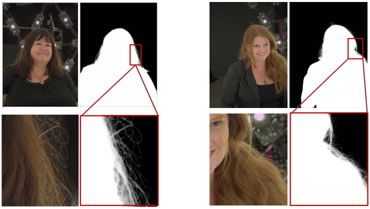
Light Stage Dataset - Matting Ground Truth

Ratio Matting

Estimated mattes are aligned with the ‘tracking’ frame (fully lit image) using dense flow

Light Stage Dataset - Matting Ground Truth

Deep Background Matting



Light Stage Dataset - Matting Ground Truth

Deep Background Matting

Multiple Light Stage Viewpoints

Estimated Alpha

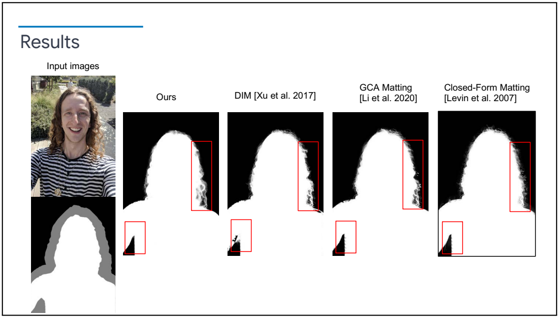
Deep Matting Module - Losses

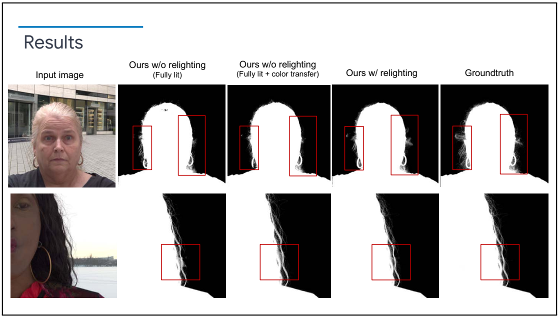
All compared with ground truth:

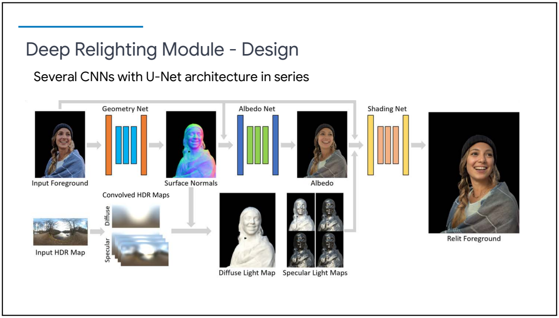
- Trimap loss
- Alpha loss (only in unknown regions)
- Foreground loss (only for pixels where $\alpha_{gt} > 0$)
- Compositing loss (using "over" operator)
- Pyramid Laplacian loss (multi-scale loss on alpha mattes)

Results

Results


Results
Input images
Ours
DIM [Xu et al. 2017]
GCA Matting
[Li et al. 2020]
Closed-Form Matting
[Levin et al. 2007]

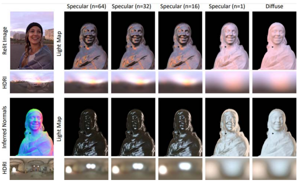

Results
Input image
Ours w/o relighting
(Fully lit)
Ours w/o relighting
(Fully lit + color transfer)
Ours w/ relighting
Groundtruth


Deep Relighting Module - Design
Several CNNs with U-Net architecture in series
Geometry Net
Albedo Net
Shading Net
Input Foreground
Surface Normals
Albedo
Relit Foreground
Convolved HDR Maps
Diffuse
Specular
Input HDR Map
Diffuse Light Map
Specular Light Maps

Deep Relighting Module - Lighting Representation

Light Map Representation:
Borrowing from real-time graphics

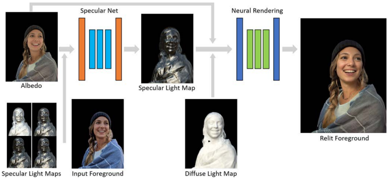
Lighting can be injected into the network along channels, with spatially-aligned pixel inputs, desirable for U-Nets.



Deep Relighting Module - Lighting Representation

Neural rendering network essentially takes Phong shaded reflection components as inputs, and learns complex illumination effects by treating the rest as a residual learning problem.

Deep Relighting Module - Shading Network



Light Stage Dataset

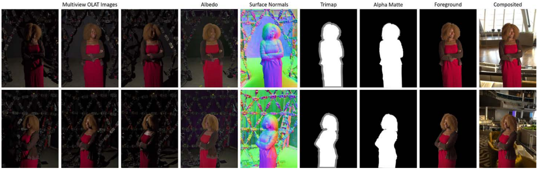
Dataset spans:
- Identity (70)
- Viewpoint
- Lighting
- Expression
- Accessorizations

Light Stage Dataset

Multi-view OLAT images

Multi-view composited images

Light Stage Dataset - Relighting Ground Truth



Deep Relighting Module - Losses

All compared with ground truth:

- Geometry loss
- Albedo loss
- Relit image loss
- Albedo VGG loss
- Relit image VGG loss

- Specular-weighted relit image loss
  - Encourage network to produce specularities
- Albedo adversarial loss (face region)
- Relit image adversarial loss (face region)

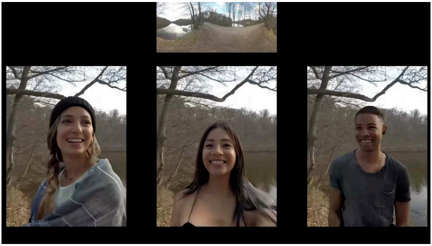
Results

Input Image

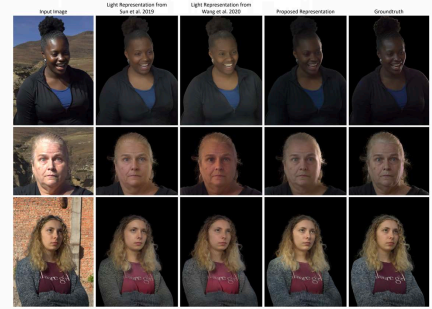
Results

Input Image

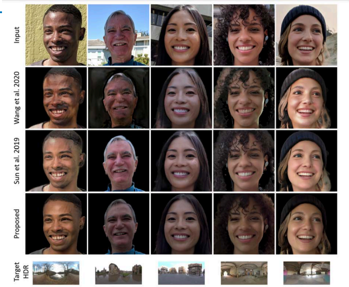
Results



Results



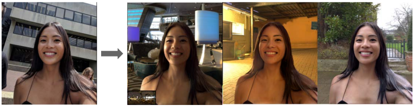
Results



Video Results

Extension - relighting using Deep Light

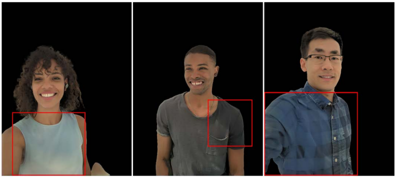
Application: Data Augmentation



...

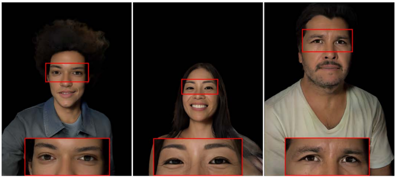
Limitations

Imprecise Albedo on Clothing



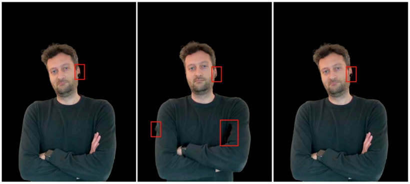
Limitations

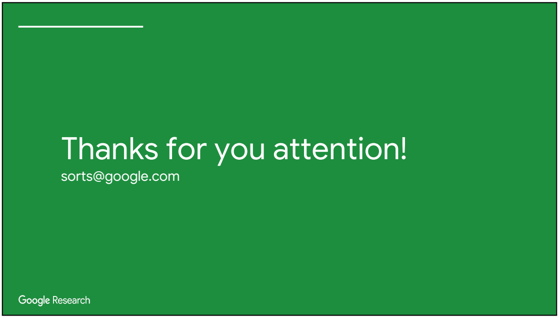
Missing Eyes Specularities



Limitations

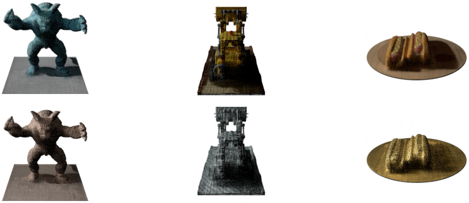
Temporal Instability of Matte Estimates




Thanks for you attention!
sorts@google.com
Google Research

Relightable Neural Radiance Fields

Pratul Srinivasan
https://pratulsrinivasan.github.io/
Google Research

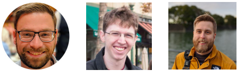


About Me

PhD at UC Berkeley
Graduated December 2020
Advisors: Ren Ng and Ravi Ramamoorthi

Google Research

Research Scientist at Google Research
Group led by David Salesin
Manager: Jon Barron
w/ Ben Mildenhall and Peter Hedman

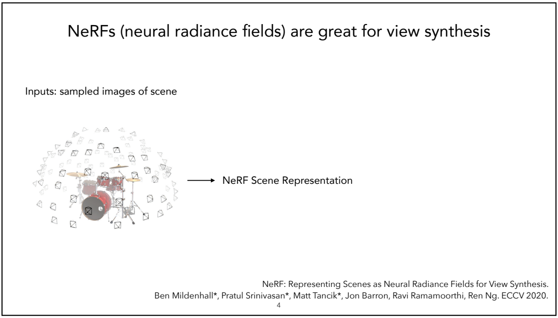


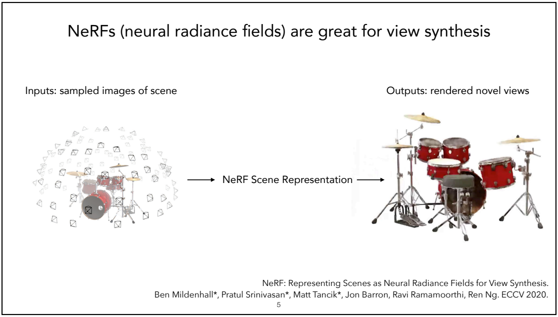
NeRFs (neural radiance fields) are great for view synthesis

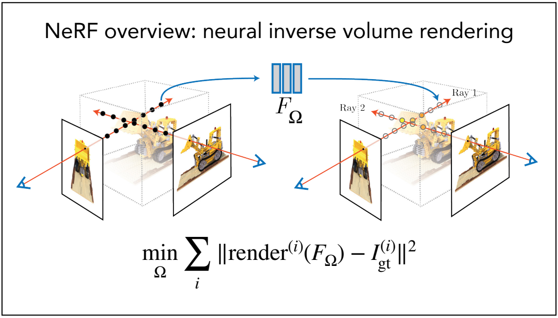
Inputs: sampled images of scene

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.
Ben Mildenhall*, Pratul Srinivasan*, Matt Tancik*, Jon Barron, Ravi Ramamoorthi, Ren Ng. ECCV 2020.

3

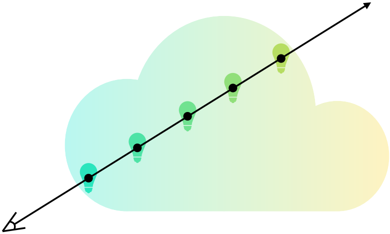

NeRFs (neural radiance fields) are great for view synthesis
Inputs: sampled images of scene
NeRF Scene Representation
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.
Ben Mildenhall*, Pratul Srinivasan*, Matt Tancik*, Jon Barron, Ravi Ramamoorthi, Ren Ng. ECCV 2020.
4


NeRFs (neural radiance fields) are great for view synthesis
Inputs: sampled images of scene
Outputs: rendered novel views
NeRF Scene Representation
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.
Ben Mildenhall*, Pratul Srinivasan*, Matt Tancik*, Jon Barron, Ravi Ramamoorthi, Ren Ng. ECCV 2020.
5

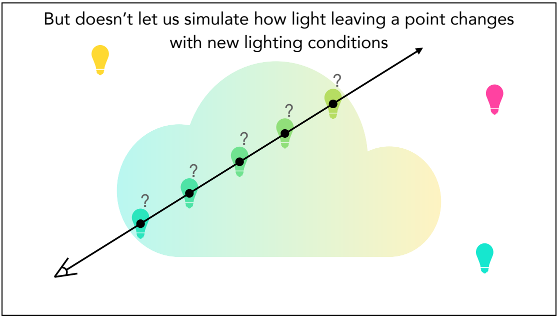

NeRF overview: neural inverse volume rendering
F_Ω
Ray 1
Ray 2
$\min_{\Omega} \sum_i \|\mathrm{render}^{(i)}(F_\Omega) - I_{gt}^{(i)}\|^2$

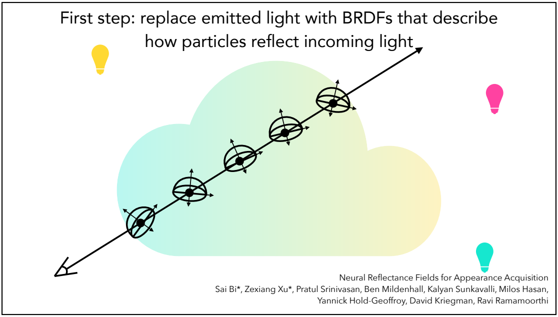
Can we use a similar neural inverse volume rendering approach to recover relightable 3D representations of objects?

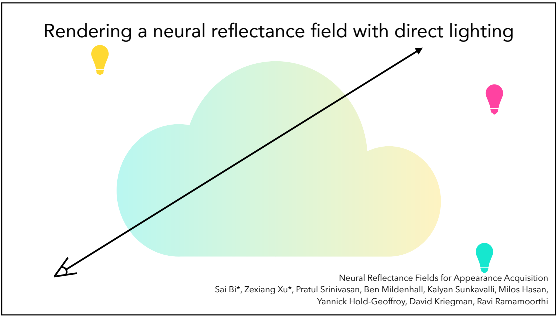
NeRF represents a volume of particles that emit light

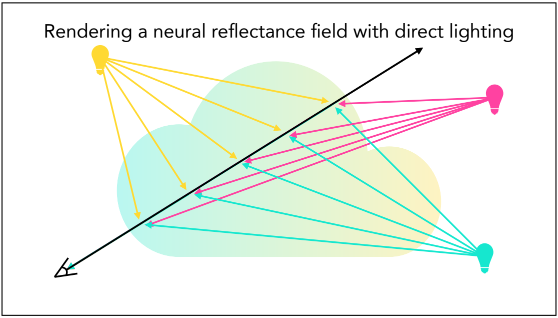


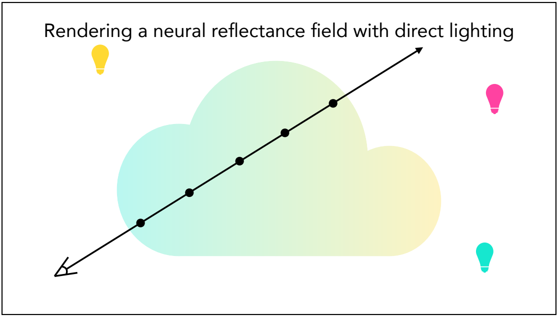
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.
Ben Mildenhall*, Pratul Srinivasan*, Matt Tancik*, Jon Barron, Ravi Ramamoorthi, Ren Ng. ECCV 2020.

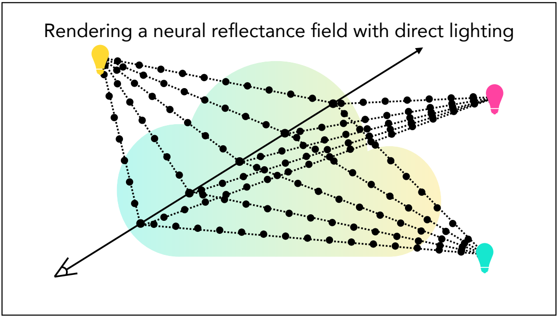

But doesn't let us simulate how light leaving a point changes with new lighting conditions
?
?
?
?
?


First step: replace emitted light with BRDFs that describe how particles reflect incoming light
Neural Reflectance Fields for Appearance Acquisition
Sai Bi*, Zexiang Xu*, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan,
Yannick Hold-Geoffroy, David Kriegman, Ravi Ramamoorthi

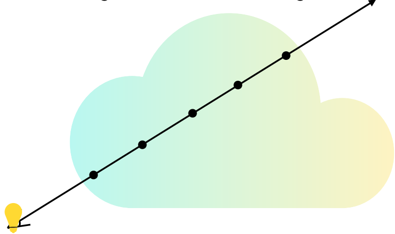

Rendering a neural reflectance field with direct lighting
Neural Reflectance Fields for Appearance Acquisition
Sai Bi*, Zexiang Xu*, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan,
Yannick Hold-Geoffroy, David Kriegman, Ravi Ramamoorthi


Rendering a neural reflectance field with direct lighting


Rendering a neural reflectance field with direct lighting


Rendering a neural reflectance field with direct lighting

Just direct lighting with multiple sources is too slow at training time —> use colocated light and camera

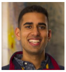


Can we simulate more complex light transport in a way that is still tractable during training?

NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis
CVPR 2021
pratulsrinivasan.github.io/nerv/



Pratul Srinivasan



Google Research

Boyang Deng

Google Research

Xiuming Zhang

MIT

Matt Tancik

UC Berkeley

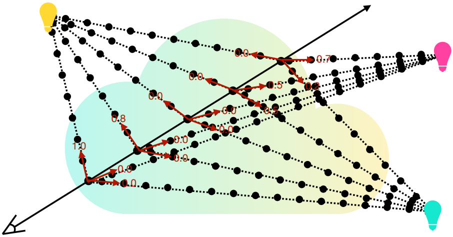
Ben Mildenhall

Google Research

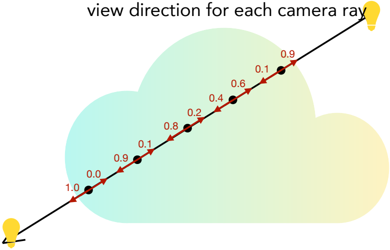
Jon Barron

Google Research

Idea: can we approximate visibility computation during training?



We already observe these values during training along +/- view direction for each camera ray



What about indirect illumination?

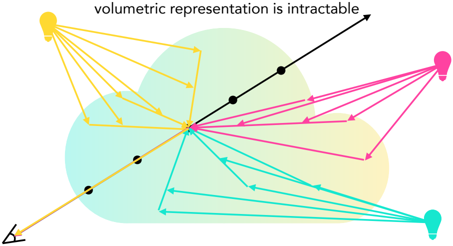
What about indirect illumination?

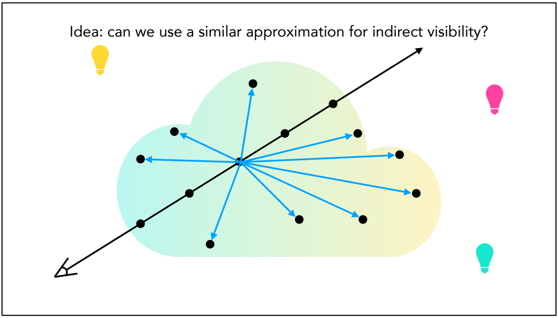
Credit: Eric Tabellion, Dreamworks

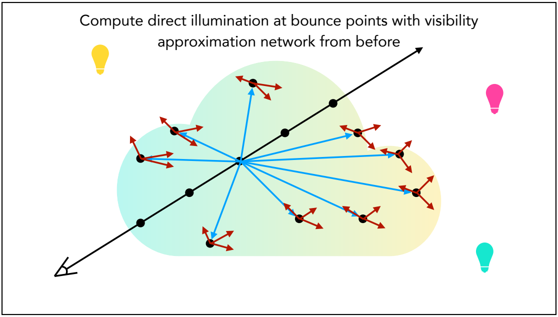
What about indirect illumination?

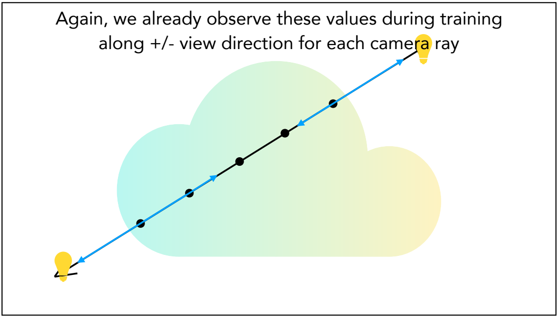
Credit: Eric Tabellion, Dreamworks

Brute-force simulation of indirect illumination inside a neural volumetric representation is intractable




Idea: can we use a similar approximation for indirect visibility?

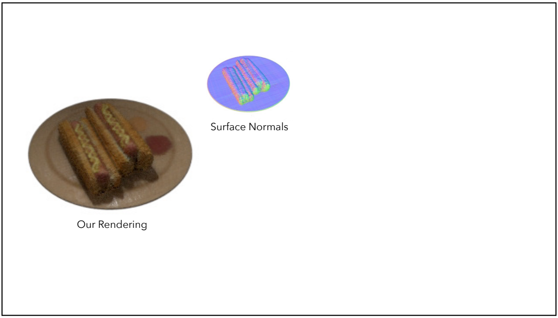

Compute direct illumination at bounce points with visibility
approximation network from before


Again, we already observe these values during training
along +/- view direction for each camera ray

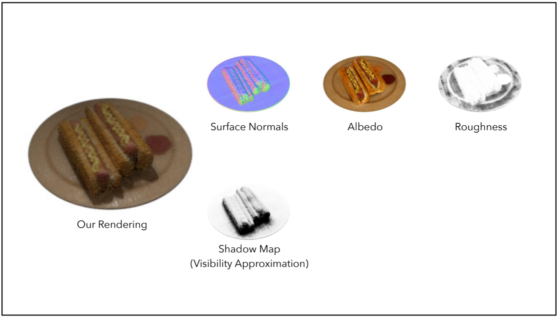
Experiments

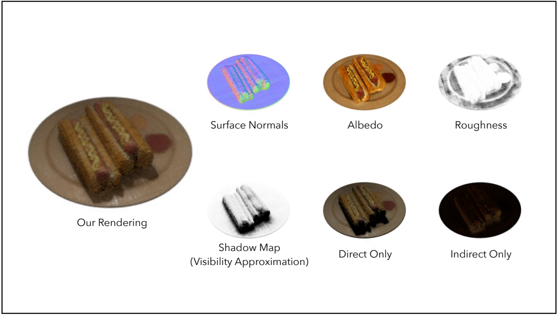
Input Images

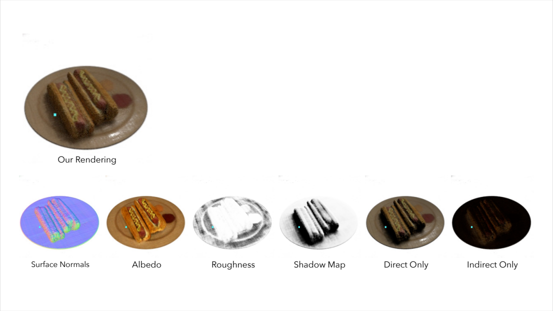
Our Rendering

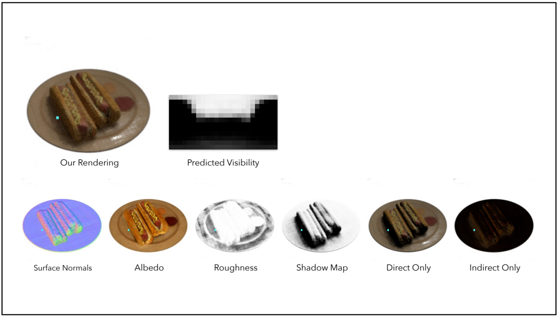

Surface Normals
Our Rendering

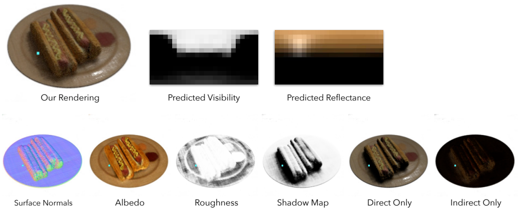

Surface Normals
Albedo
Roughness
Our Rendering

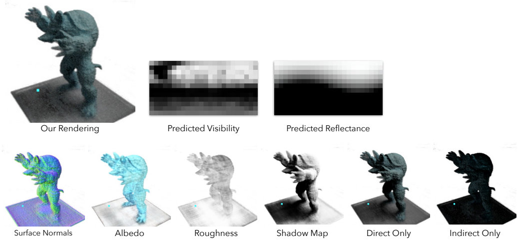

Surface Normals
Albedo
Roughness
Our Rendering
Shadow Map
(Visibility Approximation)


Surface Normals
Albedo
Roughness
Our Rendering
Shadow Map
(Visibility Approximation)
Direct Only
Indirect Only

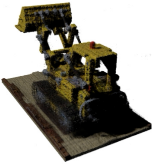

Our Rendering
Surface Normals
Albedo
Roughness
Shadow Map
Direct Only
Indirect Only


Our Rendering
Predicted Visibility
Surface Normals
Albedo
Roughness
Shadow Map
Direct Only
Indirect Only

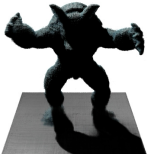




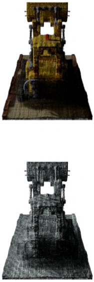
Our renderings, with and without indirect illumination



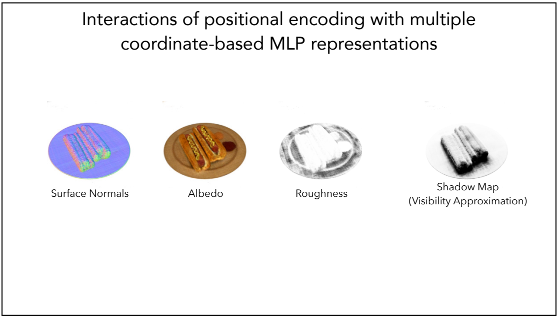
Our renderings, with and without indirect illumination

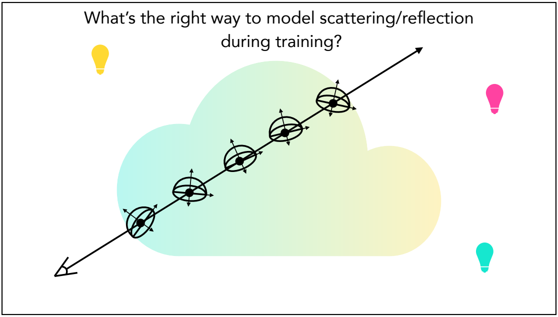


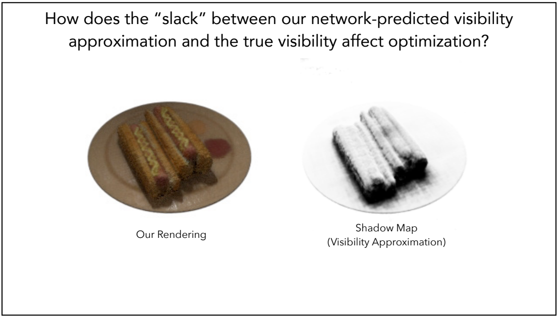
Our renderings, with and without indirect illumination



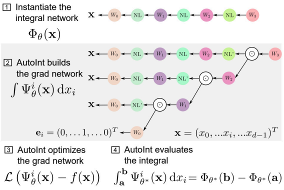
Open research questions for relightable NeRF-like models


Interactions of positional encoding with multiple coordinate-based MLP representations
Surface Normals
Albedo
Roughness
Shadow Map
(Visibility Approximation)


What's the right way to model scattering/reflection during training?


How does the "slack" between our network-predicted visibility approximation and the true visibility affect optimization?
Our Rendering
Shadow Map
(Visibility Approximation)

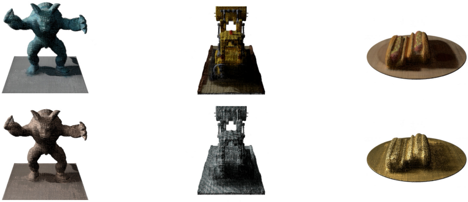
Automatic Integration for ensuring that estimated visibility is always consistent with current geometry during training?



AutoInt: Automatic Integration for Fast Neural Volume Rendering.
David B. Lindell*, Julien N. P. Marte*, Gordon Wetzstein, CVPR 2021.

Relightable Neural Radiance Fields

Pratul Srinivasan
https://pratulsrinivasan.github.io/
Google Research



Compositional Scene Representations

SIGGRAPH 2021 Course: Advances in Neural Rendering

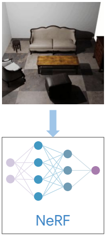
Michelle Guo
Stanford University

Recently, NeRF has demonstrated unprecedented fidelity on view synthesis

Mildenhall et al. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.* ECCV 2020

2

NeRF works great for static scenes



3

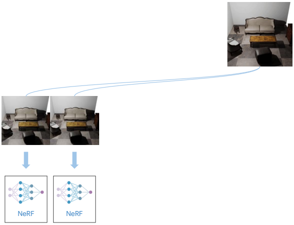
But how does NeRF work for dynamic scenes?

4

Limitation: NeRF requires retraining per frame

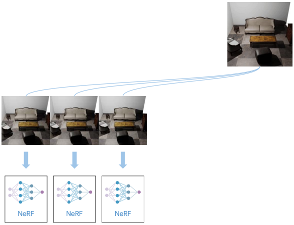
5

Limitation: NeRF requires retraining per frame



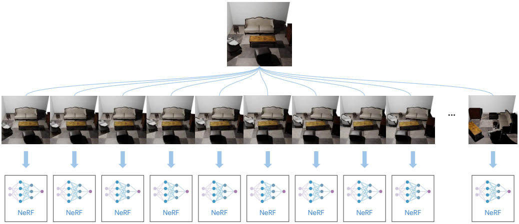
6

Limitation: NeRF requires retraining per frame



7

Limitation: NeRF requires retraining per frame

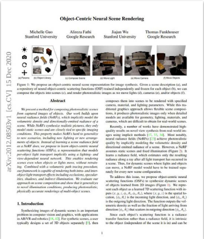


8

Find me on arXiv

Object-Centric Neural Scene Rendering

Move Camera Move Lighting Move Objects



Object-Centric Neural Scene Rendering

Michelle Guo, Stanford University; Alireza Fathi, Google Research; Jiajun Wu, Stanford University; Thomas Funkhouser, Google Research

arXiv:2012.08503v1 [cs.CV] 15 Dec 2020

Figure 1: We propose an object-centric neural scene representation for image synthesis. Given a scene description (a), and a repository of neural object-centric scattering functions (OSF) trained independently and frozen for each object (b), we can compose the objects into scenes (c), and render photorealistic images as we move lights (d), cameras (e), and/or objects (f).

Abstract

We present a method for composing photorealistic scenes from captured images of objects. Our work builds upon neural radiance fields (NeRFs), which implicitly model the volumetric density and directionally-emitted radiance of a scene. While NeRFs synthesize realistic pictures, they only model static scenes and are closely tied to specific imaging conditions. This property makes NeRFs hard to generalize to new scenarios, including new lighting or new arrangements of objects. Instead of learning a scene radiance field as a NeRF does, we propose to learn object-centric neural scattering functions (OSFs), a representation that models per-object light transport implicitly using a lighting- and view-dependent neural network. This enables rendering scenes even when objects or lights move, without retraining. Combined with a volumetric path tracing procedure, our framework is capable of rendering both intra- and inter-object light transport effects including occlusions, specularities, shadows, and indirect illumination. We evaluate our approach on scene composition and show that it generalizes to novel illumination conditions, producing photorealistic, physically accurate renderings of multi-object scenes.

1. Introduction

Synthesizing images of dynamic scenes is an important problem in computer vision and graphics, with applications in AR/VR and robotics [18, 42]. For synthetic scenes, a user typically designs a set of 3D objects separately [5], then composes them into scenes to be rendered with specified camera, material, and lighting parameters. While this traditional graphics approach allows flexible scene compositions, it produces photorealistic images only when detailed models are available for geometry, lighting, materials, and cameras, which are difficult to obtain for real-world scenes.

Recently, a number of works have demonstrated high-quality results on novel view synthesis from real-world images using implicit methods [17, 33, 34]. Most notably, neural radiance fields (NeRFs) [23] achieve photorealistic quality by implicitly modeling the volumetric density and directional emitted radiance of a scene. However, a NeRF assumes static scenes and fixed illumination (Figure 2). It learns a radiance field, which estimates only the resulting radiance along a ray after all light transport has occurred in a scene. Thus, for dynamic scenes where lights and objects can move, a NeRF model would have to be trained separately for every new scene configuration.

To address this issue, we propose object-centric neural scattering functions (OSFs) to synthesize dynamic scenes of objects learned from 2D images (Figure 1). We represent each object as a learned 7D scattering function with inputs $(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o)$, where (x, y, z) is the spatial location, (ϕ_i, θ_i) is the incoming light direction, and (ϕ_o, θ_o) is the outgoing light direction. The function outputs the volumetric density as well as the fraction of light arriving from direction (ϕ_i, θ_i) that scatters in outgoing direction (ϕ_o, θ_o).

Since each object's scattering function is a radiance transfer function rather than a radiance field, it is intrinsic to the object (independent of the scene it is in) and can be

1

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

9

Train each object once, compose infinite number of scenes

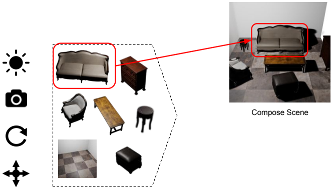
Scene Parameters

OSF Object Library
(no retraining required!)

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

10

Train each object once, compose infinite number of scenes



Compose Scene

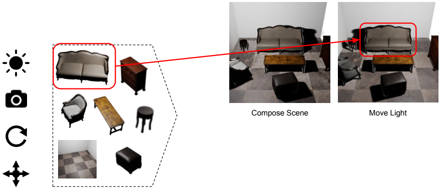
Scene Parameters

OSF Object Library
(no retraining required!)

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

11

Train each object once, compose infinite number of scenes



Compose Scene

Move Light

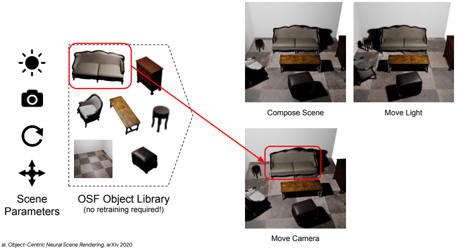
Scene Parameters

OSF Object Library
(no retraining required!)

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

12

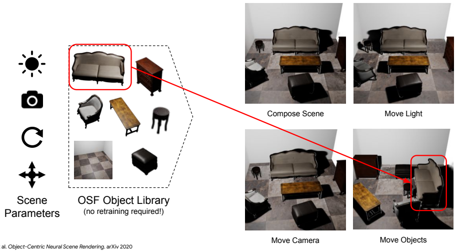
Train each object once, compose infinite number of scenes



Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

13

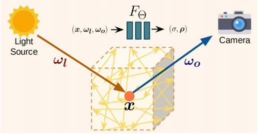
Train each object once, compose infinite number of scenes



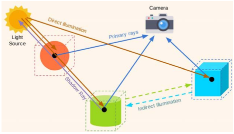
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

14

This Talk: Learning object vs. scene light transport



Object-Level Light Transport



Scene-Level Light Transport

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

15

Rendering Equation

$\tau(t) = \exp\left(-\int_{t_n}^{t} \sigma((u))du\right)$

$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

16

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

Rendering Equation

$\tau(t) = \exp\left(-\int_{t_n}^{t} \sigma((u))du\right)$

$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

$$L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) = \int_{\mathcal{S}} L(\boldsymbol{x}, \boldsymbol{\omega_l})f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o})d\boldsymbol{\omega_l}$$

17

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

Rendering a single object

$\tau(t) = \exp\left(-\int_{t_n}^{t} \sigma((u))du\right)$

$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

$$L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) = L(\boldsymbol{x}, \boldsymbol{\omega_l})f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o})$$

18

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

Rendering a single object

$\tau(t) = \exp\left(-\int_{t_n}^{t} \sigma((u))du\right)$

$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

$$\begin{aligned} L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) &= L(\boldsymbol{x}, \boldsymbol{\omega_l}) f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o}) \\ &= f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o}) \end{aligned}$$

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

19

7D Object-Centric Neural Scene Scattering Function (OSF)

$$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$$

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

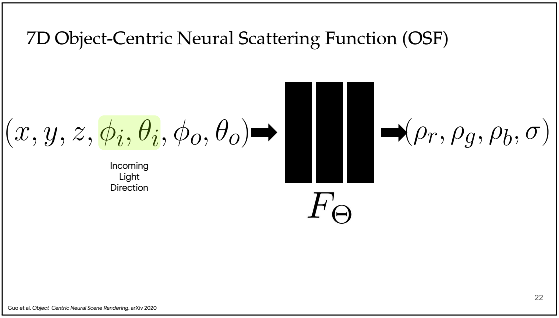
20

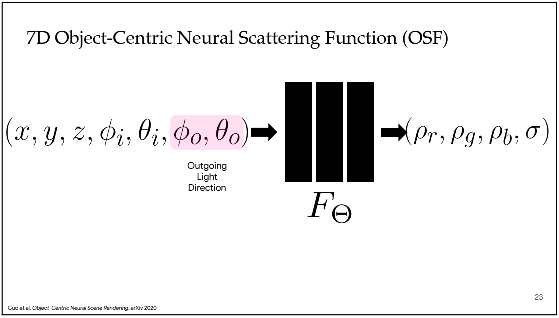
7D Object-Centric Neural Scattering Function (OSF)

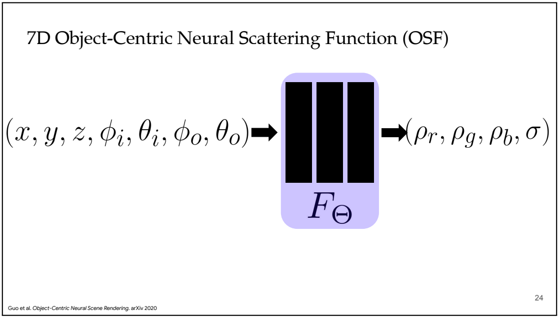
$$(\underbrace{x, y, z}_{\text{Spatial Location}}, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$$

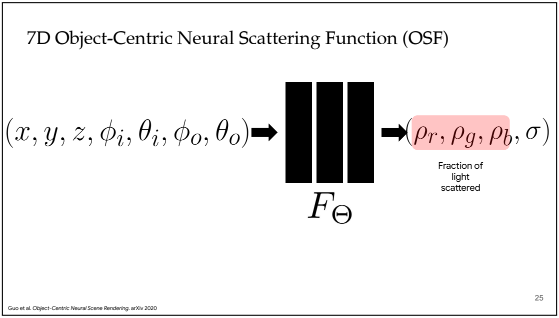
Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

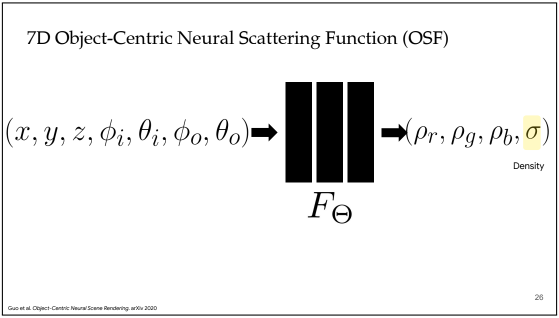
21

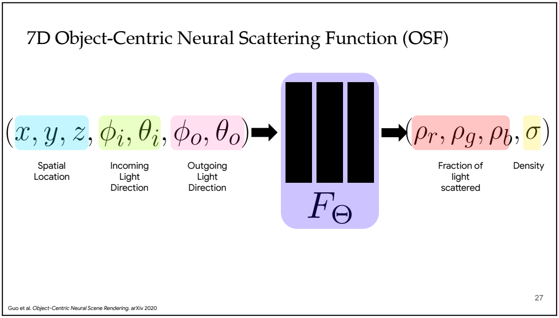

7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Incoming Light Direction
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
22


7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Outgoing Light Direction
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
23

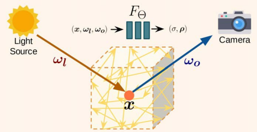

7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
24


7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Fraction of light scattered
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020
25

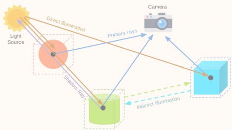

7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Density
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020
26


7D Object-Centric Neural Scattering Function (OSF)
$(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o) \rightarrow F_\Theta \rightarrow (\rho_r, \rho_g, \rho_b, \sigma)$
Spatial Location
Incoming Light Direction
Outgoing Light Direction
Fraction of light scattered
Density
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020
27

Learning object vs. scene light transport



Object-Level Light Transport

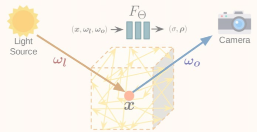


Scene-Level Light Transport

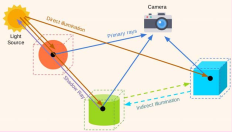
28

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

Learning object vs. scene light transport



Object-Level Light Transport



Scene-Level Light Transport

29

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

Rendering a single object

$$\tau(t) = \exp\left(-\int_{t_n}^{t} \sigma((u))du\right)$$

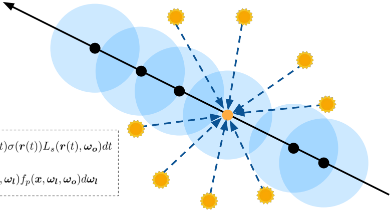
$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

$$\begin{aligned} L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) &= L(\boldsymbol{x}, \boldsymbol{\omega_l}) f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o}) \\ &= f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o}) \end{aligned}$$

30

Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

OSF integrates over sphere of all incoming light directions



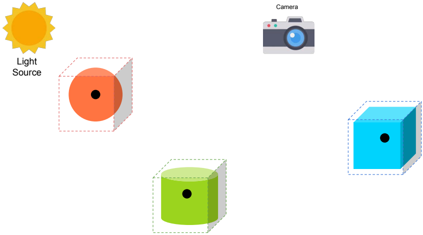
$$L(\boldsymbol{x}_0, \boldsymbol{\omega_o}) = \int_{t_n}^{t_f} \tau(t)\sigma(\boldsymbol{r}(t))L_s(\boldsymbol{r}(t), \boldsymbol{\omega_o})dt$$

$$L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) = \int_{\mathcal{S}} L(\boldsymbol{x}, \boldsymbol{\omega_l}) f_p(\boldsymbol{x}, \boldsymbol{\omega_l}, \boldsymbol{\omega_o}) d\boldsymbol{\omega_l}$$

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

31

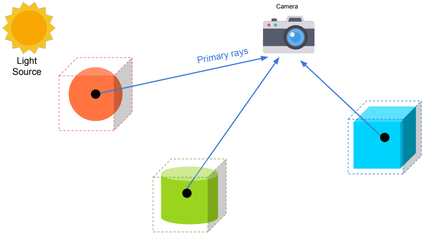
Volumetric path tracing



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

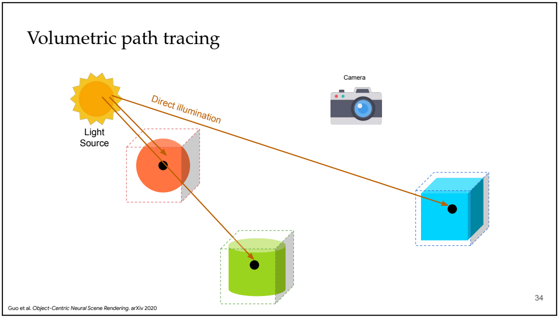
32

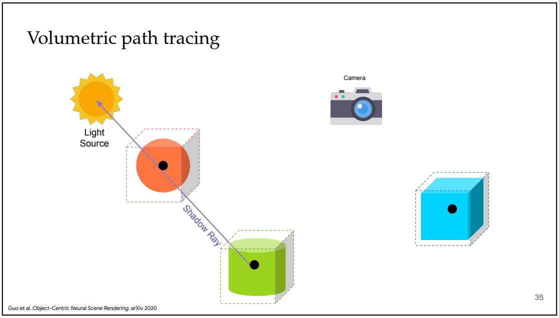
Volumetric path tracing

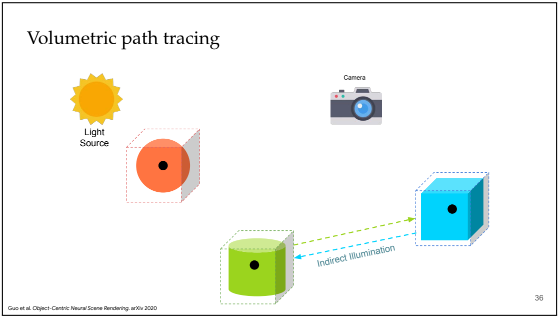


Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

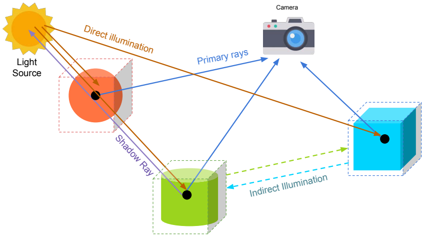
33


Volumetric path tracing
Camera
Light Source
Direct illumination
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
34


Volumetric path tracing
Camera
Light Source
Shadow Ray
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
35


Volumetric path tracing
Camera
Light Source
Indirect Illumination
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
36

Volumetric path tracing



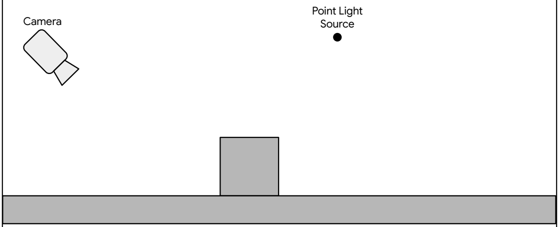
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

37

Example: Shadow Rays

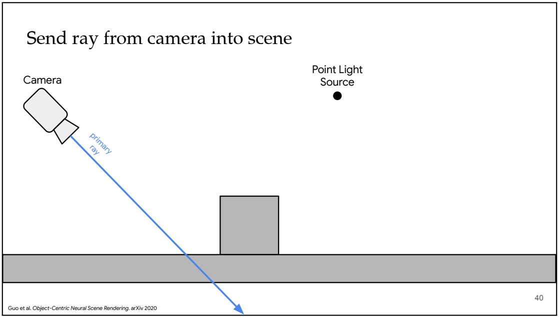
38

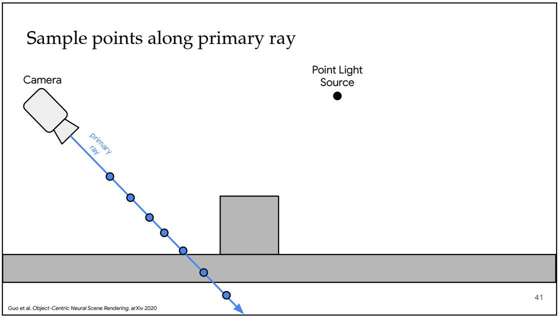
Given a scene specification

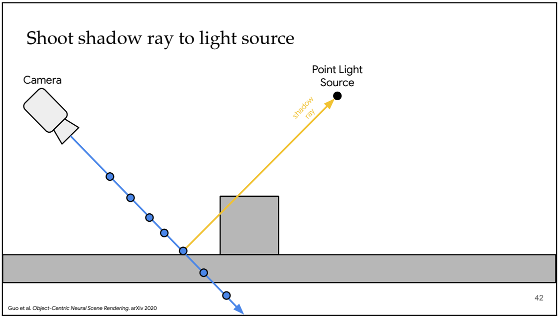


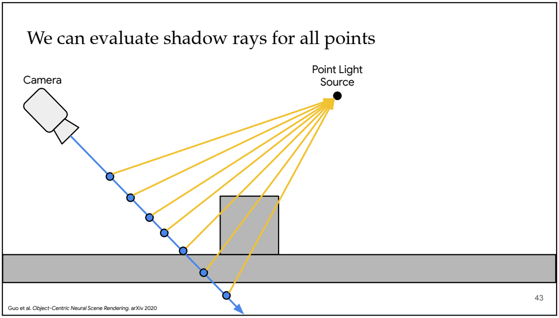
Guo et al. *Object-Centric Neural Scene Rendering.* arXiv 2020

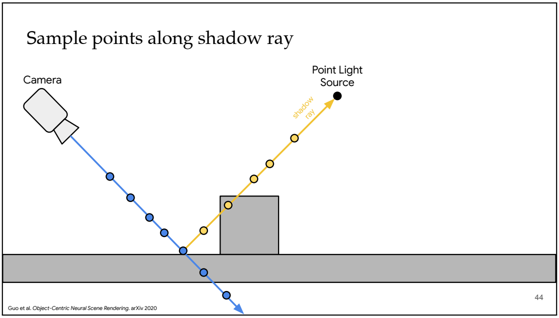
39

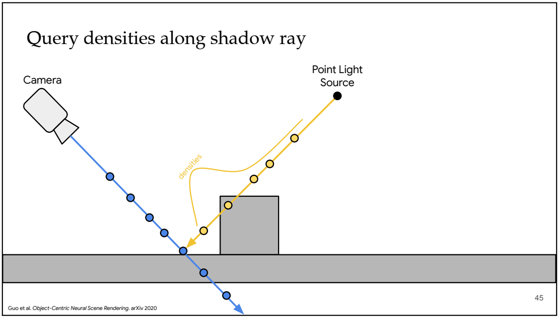

Send ray from camera into scene
Camera
Point Light
Source
primary
ray
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
40

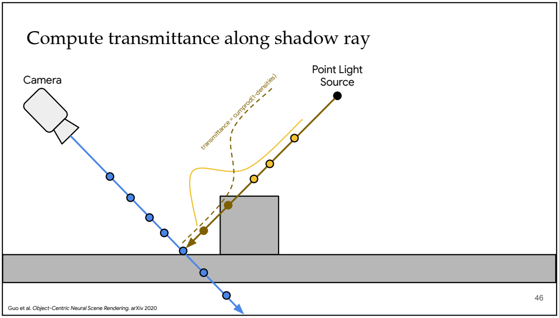

Sample points along primary ray
Camera
Point Light
Source
primary
ray
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
41

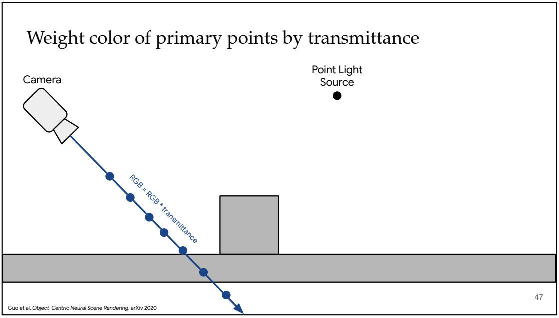

Shoot shadow ray to light source
Camera
Point Light
Source
shadow
ray
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
42

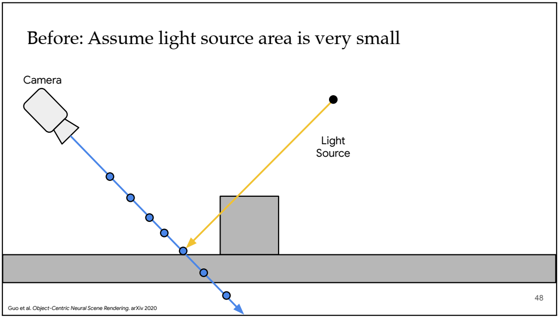

We can evaluate shadow rays for all points
Camera
Point Light Source
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
43


Sample points along shadow ray
Camera
Point Light Source
shadow ray
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
44

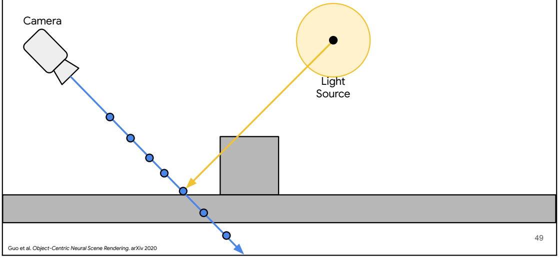

Query densities along shadow ray
Camera
Point Light Source
densities
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
45


Compute transmittance along shadow ray
Camera
Point Light
Source
transmittance = cumprod(1-densities)
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
46


Weight color of primary points by transmittance
Camera
Point Light
Source
RGB = RGB * transmittance
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
47


Before: Assume light source area is very small
Camera
Light
Source
Guo et al. Object-Centric Neural Scene Rendering. arXiv 2020
48

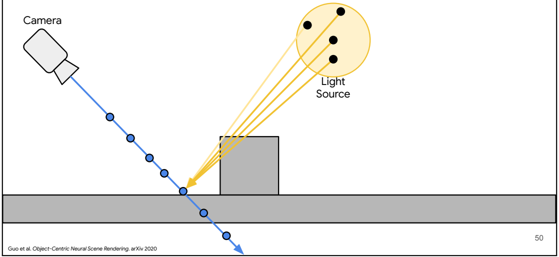
Now: Assume light source is larger circular area



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

49

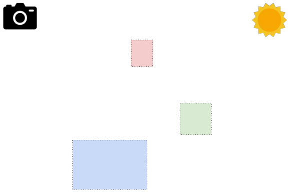
Average over multiple rays from light source



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

50

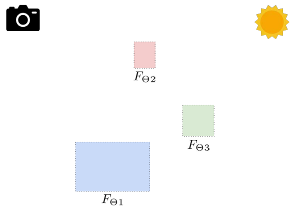
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

51

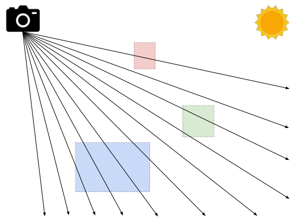
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

52

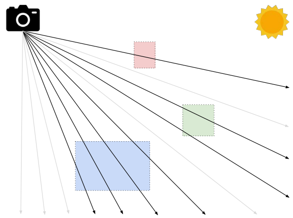
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

53

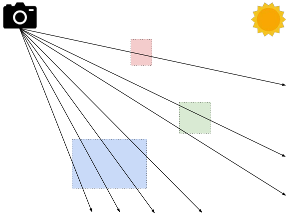
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

54

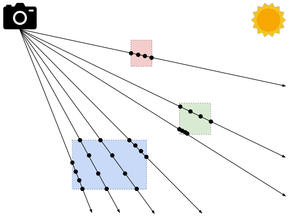
We adopt a sparse ray-box intersection approach



55

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

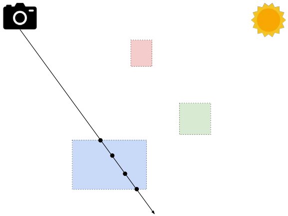
We adopt a sparse ray-box intersection approach



56

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

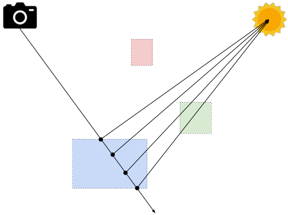
We adopt a sparse ray-box intersection approach



57

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

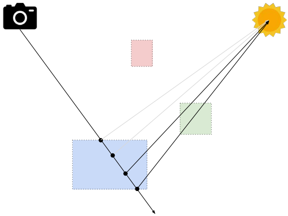
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

58

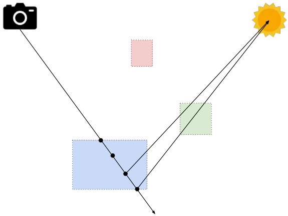
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

59

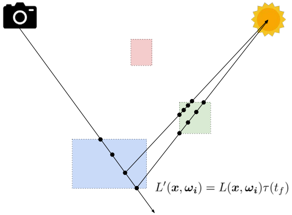
We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

60

We adopt a sparse ray-box intersection approach



Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

61

Results

62

Generalization to novel illumination

63

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

Scene composition: Moving Objects

Ours | o-NeRF + S | o-NeRF | Ground Truth

64

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

Scene composition: Moving Camera

Ours | o-NeRF + S | o-NeRF | Ground Truth

65

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

Scene composition: Moving Light

Ours | o-NeRF + S | o-NeRF | Ground Truth

66

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

Real-World Results

67

Learning OSFs on real-world objects

68

Dataset from "Neural Reflectance Fields for Appearance Acquisition" (Bi et al. 2020)

Composing real-world objects into scenes

Move Objects

Move Camera

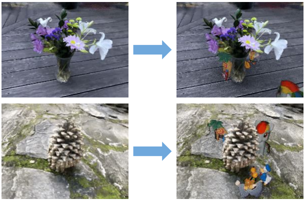
Move Light

69

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

Inserting real-world objects into outdoor scenes

NeRF (Mildenhall et al.) OSF (Ours)



70

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020

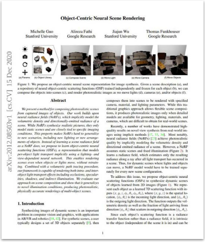
Thank You


Michelle Guo


Alireza Fathi


Jiajun Wu


Tom Funkhouser



arXiv:2012.08503v1 [cs.CV] 15 Dec 2020

Object-Centric Neural Scene Rendering

Michelle Guo, Stanford University
Alireza Fathi, Google Research
Jiajun Wu, Stanford University
Thomas Funkhouser, Google Research

(a) Params (b) Object Library (c) Compose Scene (d) Move Light (e) Move Camera (f) Move Objects

Figure 1: We propose an object-centric neural scene representation for image synthesis. Given a scene description (a), and a repository of neural object-centric scattering functions (OSF) trained independently and frozen for each object (b), we can compose the objects into scenes (c), and render photorealistic images as we move lights (d), cameras (e), and/or objects (f).

Abstract

We present a method for composing photorealistic scenes from captured images of objects. Our work builds upon neural radiance fields (NeRFs), which implicitly model the volumetric density and directionally-emitted radiance of a scene. While NeRFs synthesize realistic pictures, they only model static scenes and are closely tied to specific imaging conditions. This property makes NeRFs hard to generalize to new scenarios, including new lighting or new arrangements of objects. Instead of learning a scene radiance field as a NeRF does, we propose to learn object-centric neural scattering functions (OSFs), a representation that models per-object light transport implicitly using a lighting- and view-dependent neural network. This enables rendering scenes even when objects or lights move, without retraining. Combined with a volumetric path tracing procedure, our framework is capable of rendering both intra- and inter-object light transport effects including occlusions, specularities, shadows, and indirect illumination. We evaluate our approach on scene composition and show that it generalizes to novel illumination conditions, producing photorealistic, physically accurate renderings of multi-object scenes.

1. Introduction

Synthesizing images of dynamic scenes is an important problem in computer vision and graphics, with applications in AR/VR and robotics [18, 42]. For synthetic scenes, a user typically designs a set of 3D objects separately [5], then composes them into scenes to be rendered with specified camera, material, and lighting parameters. While this traditional graphics approach allows flexible scene compositions, it produces photorealistic images only when detailed models are available for geometry, lighting, materials, and cameras, which are difficult to obtain for real-world scenes.

Recently, a number of works have demonstrated high-quality results on novel view synthesis from real-world images using implicit methods [17, 33, 34]. Most notably, neural radiance fields (NeRFs) [23] achieve photorealistic quality by implicitly modeling the volumetric density and directional emitted radiance of a scene. However, a NeRF assumes static scenes and fixed illumination (Figure 2). It learns a radiance field, which estimates only the resulting radiance along a ray after all light transport has occurred in a scene. Thus, for dynamic scenes where lights and objects can move, a NeRF model would have to be trained separately for every new scene configuration.

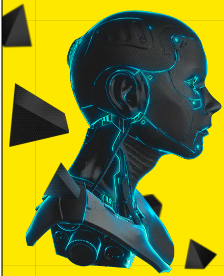
To address this issue, we propose object-centric neural scattering functions (OSFs) to synthesize dynamic scenes of objects learned from 2D images (Figure 1). We represent each object as a learned 7D scattering function with inputs $(x, y, z, \phi_i, \theta_i, \phi_o, \theta_o)$, where (x, y, z) is the spatial location, (ϕ_i, θ_i) is the incoming light direction, and (ϕ_o, θ_o) is the outgoing light direction. The function outputs the volumetric density as well as the fraction of light arriving from direction (ϕ_i, θ_i) that scatters in outgoing direction (ϕ_o, θ_o).

Since each object's scattering function is a radiance transfer function rather than a radiance field, it is intrinsic to the object (independent of the scene it is in) and can be

1

71

Guo et al. *Object-Centric Neural Scene Rendering*. arXiv 2020



SIGGRAPH 2021

- COURSE ON ADVANCES IN NEURAL RENDERING -

AN OVERVIEW OF NEURAL RADIANCE FIELDS FOR GENERAL DYNAMIC SCENES

BY EDGAR TRETSCHK
MAX PLANCK INSTITUTE FOR INFORMATICS

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

PROBLEM

SIGGRAPH 2021

[NR-NeRF]

Input: Monocular RGB video of a general dynamic scene

4D Reconstruction

Geometry
+ Appearance
+ Deformation

[NR-NeRF]

Goal: Novel view synthesis in space and time

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

2

WHAT'S DIFFICULT ABOUT THE PROBLEM?

SIGGRAPH 2021

- Need to accumulate geometry and appearance information across time, but ambiguities:

Static scenes: Geometry, Appearance

Geometry | Appearance | Geometry Changes (motion, deformations) | Appearance Changes (light, material)

Dynamic scenes: Geometry, Appearance, Geometry Changes (motion, deformations), Appearance Changes (light, material)

- Deformation modelling:
  - Priors: Only single view per time step → How does currently hidden geometry deform?
  - Parametrization: E.g., discontinuities for topology changes

THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

3

RELATED WORK

SIGGRAPH 2021

- On general dynamic reconstruction:
  - Use depth, e.g. DynamicFusion [Newcombe et al. 2015]
  - NRSfM: Only hard surface, assumes correspondences via optical flow
  - Template-based tracking from RGB, Direct, Dense, Deformable [Yu et al. 2015]
  - Image-based warping, e.g. [Yoon et al. 2020]
- Neural representations:
  - Use multi-view, e.g. Neural Volumes [Lombardi et al. 2019], [Bansal et al. 2020]
  - Only hard surfaces, e.g. Differentiable Volumetric Rendering [Niemeyer et al. 2020]
  - 3D supervision, e.g. Occupancy Flow [Niemeyer et al. 2019]
  - Image-based without full 3D representation, e.g. X-Fields [Bemana et al. 2020]

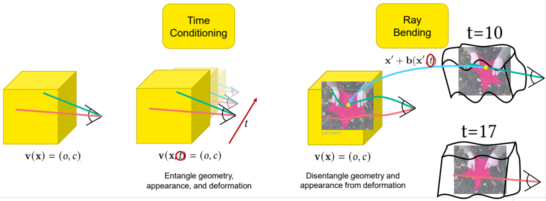
THE NERF WAVE

SIGGRAPH 2021

- A number of concurrent works (by predominant input video type used):
  - Real monocular video:
    - Nerfies (Park et al. 2020)
    - NeRFlow (Du et al. 2020)
    - NR-NeRF (Tretschk et al. 2020)
    - NSFF (Li et al. 2021)
    - Video-NeRF (Xian et al. 2020)
  - Real multi-view video:
    - DyNeRF (Li et al. 2021)
  - Synthetic video:
    - D-NeRF (Pumarola et al. 2021)
    - NeRFlow (Du et al. 2020)

HOW CAN WE PARAMETRIZE DEFORMATIONS?

SIGGRAPH 2021



HOW CAN WE PARAMETRIZE DEFORMATIONS?

SIGGRAPH 2021

TIME CONDITIONING | **RAY BENDING**

Correspondences/ Temporal Consistency

- "Soft"
  - Via auxiliary 3D scene flow
    - Correspondences can drift over time
  - Can reconstruct frames individually, incorrect correspondences not too problematic
  - → Handles larger motion empirically

- "Hard"
  - Via ray bending into canonical model
    - Correspondences cannot drift over time
  - Difficult to recover from wrong estimation of large motion early during training
    - D-NeRF: Use curriculum
  - → Handles smaller motion empirically

Appearance Changes (light, material)

- Straightforward
- Not straightforward
  - Canonical model is time-independent

Input

Topology Changes

Result

- Straightforward
- Not straightforward
  - Ray bending needs to allow for discontinuities

A CLOSER LOOK AT TIME CONDITIONING

SIGGRAPH 2021

How can we share and regularize geometry and appearance across time?

NERFLOW | **NSFF** | **VIDEO-NERF**

Static

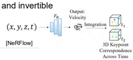
NERFLOW:
- Losses:
  - Discourage motion in empty space
  - Rigidity across space and time

NSFF:
- Second, static NeRF

VIDEO-NERF:
- Losses:
  - Non-surface points hidden in input frame remain static

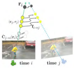
Dynamic

NERFLOW:
- Losses:
  - Geometry, appearance, keypoint positions consistent in 3D across time
- Scene flow MLP:
  - Neural ODE → Continuous in time and invertible



[NeRFlow]

NSFF:
- Losses:
  - 2D rendering consistent across time
  - Small, smooth, sparse scene flow
- Scene flow MLP:
  - Offsets → Discrete in time and not invertible



[NSFF]

A CLOSER LOOK AT RAY BENDING

SIGGRAPH 2021

Ray bending ($\mathbf{x} = \mathbf{x}' + \mathbf{b}(\mathbf{x}', t)$) is always parametrized by an MLP, not classically. Why?

ADVANTAGES

- Flexible and expressive
- Continuous in space and time, and differentiable
  - → Easily realize continuous, natural deformation priors without any need to approximate them on a discrete representation like a mesh
    - Nerfies: Elasticity via singular values of Jacobian of $\mathbf{b}(\mathbf{x}', t)$
    - NR-NeRF: Rigidity via divergence of $\mathbf{b}(\mathbf{x}', t)$
  - → Easily determine instantaneous ray direction of bent ray necessary for view-dependent effects
- Inherent smoothness → Inpaints information on its own
- Volumetric → Can handle smoke or fluids
- Spatially and temporally consistent novel-view renderings
- Natural coarse-to-fine strategy by phasing in higher frequencies of positionally encoded input in the course of training

DISADVANTAGES

- Completely opaque parametrization → Difficult to edit and manipulate [NR-NeRF]
- Entangled, no localization in space or time → Need to prevent loss of already-learned information during training
- Inherently continuous → Topology changes are difficult
- Non-trivial to invert

Without rigidity prior

[NR-NeRF]

With rigidity prior

WHAT ABOUT NOVEL VIEWS?

SIGGRAPH 2021

- Goal: Spatiotemporal novel-view camera trajectory significantly different from input camera trajectory
- But: At time *t*, only geometry visible in the input frame is constrained
- → Need to impose priors on hidden geometry to prevent artifacts

HIDDEN STATIC PARTS

- NSFF: Use a separate, static NeRF
- Nerfies: Precompute 3D background points and encourage them to remain static
- NR-NeRF: Keep points that are static across input frames static

HIDDEN DYNAMIC PARTS

- NeRFlow: Temporal smoothness
- NR-NeRF: Local spatial rigidity

Input NR-NeRF Two methods without constraints on hidden dynamic parts

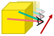
[NR-NeRF]

CONCLUSION

SIGGRAPH 2021

- Two classes of deformation models with complementary advantages:

Time Conditioning
+ larger motion
+ topology change
+ material and light changes



?

Ray Bending
+ hard correspondences across time / temporal consistency

Every time step is a keyframe → Only one keyframe

- Some future directions:
  - Modelling physics (light, material, deformations, etc.)
  - Editability

[NSFF]

Motion sculptures

[NR-NeRF]

Foreground removal

[NR-NeRF]

Motion exaggeration

SIGGRAPH 2021

THANK YOU!

MAIN REFERENCES

SIGGRAPH 2021

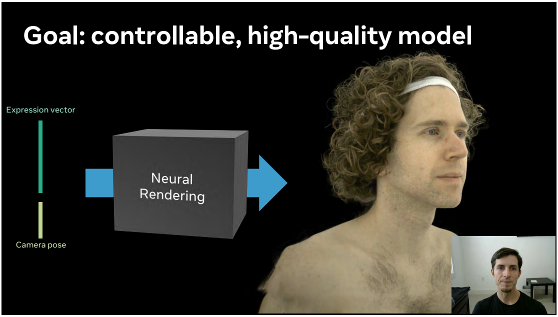
- [D-NeRF] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. Computer Vision and Pattern Recognition (CVPR).
- [DyNeRF] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. 2021. Neural 3D Video Synthesis. arXiv.
- [NeRF] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. European Conference on Computer Vision (ECCV).
- [Nerfies] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2020. Deformable Neural RadianceFields. arXiv.
- [NeRFlow] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. 2020. Neural Radiance Flow for 4D View Synthesis and Video Processing. arXiv.
- [NR-NeRF] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhoefer, Christoph Lassner, and Christian Theobalt. 2020. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. arXiv.
- [NSFF] Zhengqi Li, Simon Niklaus· Noah Snavely, and Oliver Wang. 2021. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. Computer Vision and Pattern Recognition (CVPR).
- [Video-NeRF] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2020. Space-time Neural Irradiance Fields for Free-Viewpoint Video. arXiv.

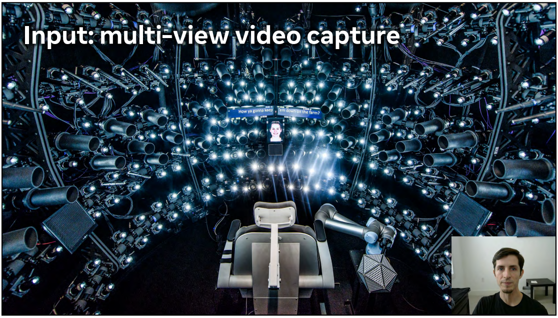
RELATED-WORK REFERENCES

SIGGRAPH 2021

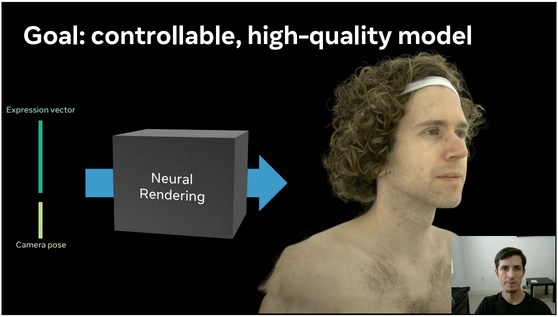
- Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. Computer Vision and Pattern Recognition (CVPR).
- Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. 2015. Direct, Dense,and Deformable: Template-Based Non-Rigid 3D Reconstruction from RGB Video. International Conference on Computer Vision (ICCV).
- Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel View Synthesis of Dynamic Scenes with Globally Coherent Depths from a Monocular Camera. Computer Vision and Pattern Recognition (CVPR).
- Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. SIGGRAPH.
- Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 2020. 4D Visualization of Dynamic Events from Unconstrained Multi-View Videos. Computer Vision and Pattern Recognition (CVPR).
- Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. Computer Vision and Pattern Recognition (CVPR).
- Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2019. Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. International Conference on Computer Vision (ICCV).
- Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. X-Fields: Implicit Neural View-, Light- and Time-Image Interpolation. SIGGRAPH Asia.


SIGGRAPH 2021
VIRTUAL 9-13 AUGUST
Efficient Neural Rendering of Dynamic Humans and Scenes
Stephen Lombardi
FACEBOOK REALITY LABS
RESEARCH

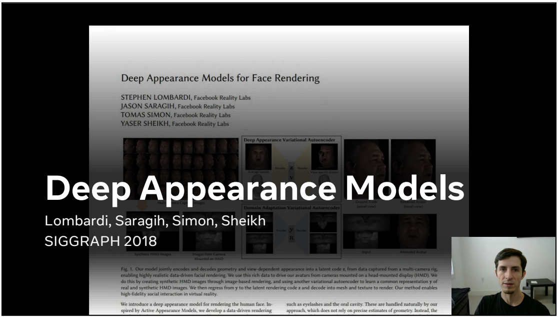

Goal: controllable, high-quality model
Expression vector
Camera pose
Neural Rendering

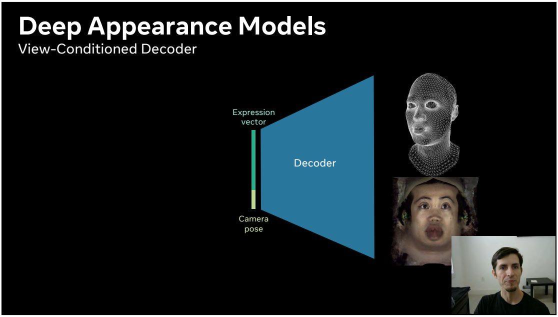

Input: multi-view video capture

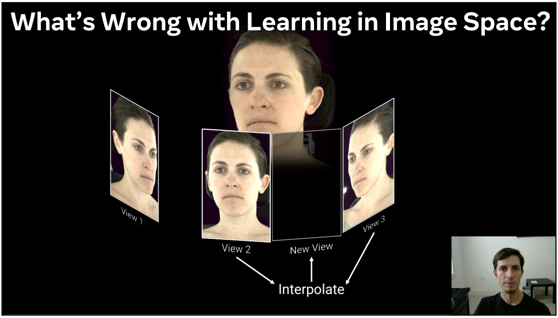

Preprocessing: facial mesh tracking

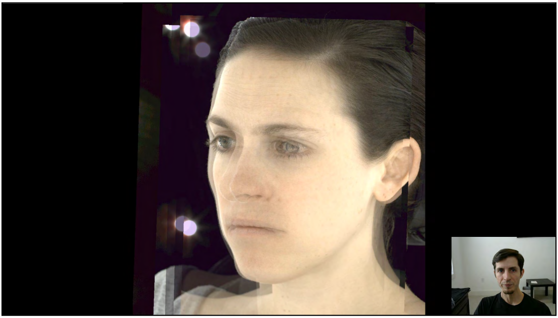

Goal: controllable, high-quality model
Expression vector
Camera pose
Neural Rendering

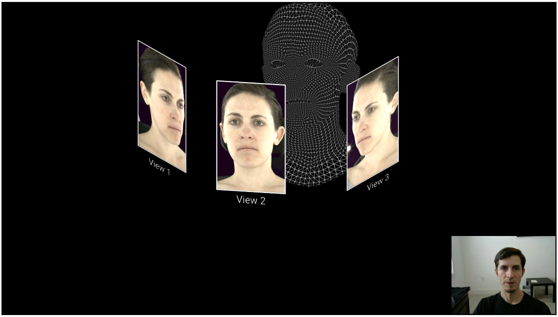

Deep Appearance Models
Improved completeness
Higher efficiency and quality

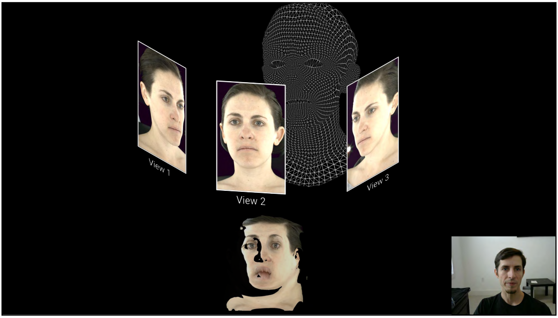

Deep Appearance Models for Face Rendering
STEPHEN LOMBARDI, Facebook Reality Labs
JASON SARAGIH, Facebook Reality Labs
TOMAS SIMON, Facebook Reality Labs
YASER SHEIKH, Facebook Reality Labs
Deep Appearance Variational Autoencoder
Domain Adaptation Variational Autoencoder
Deep Appearance Models
Lombardi, Saragih, Simon, Sheikh
SIGGRAPH 2018
Fig. 1. Our model jointly encodes and decodes geometry and view-dependent appearance into a latent code z, from data captured from a multi-camera rig, enabling highly realistic data-driven facial rendering. We use this rich data to drive our avatars from cameras mounted on a head-mounted display (HMD). We do this by creating synthetic HMD images through image-based rendering, and using another variational autoencoder to learn a common representation y of real and synthetic HMD images. We then regress from y to the latent rendering code z and decode into mesh and texture to render. Our method enables high-fidelity social interaction in virtual reality.
We introduce a deep appearance model for rendering the human face. In-spired by Active Appearance Models, we develop a data-driven rendering
such as eyelashes and the oral cavity. These are handled naturally by our approach, which does not rely on precise estimates of geometry. Instead, the

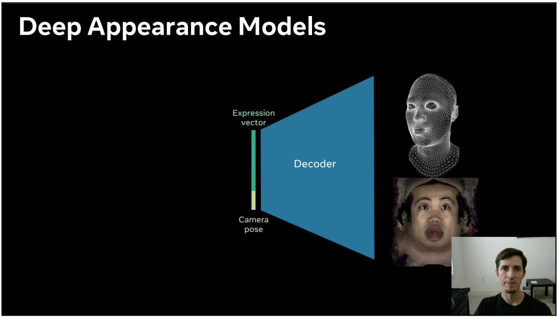

Deep Appearance Models
View-Conditioned Decoder
Expression vector
Decoder
Camera pose

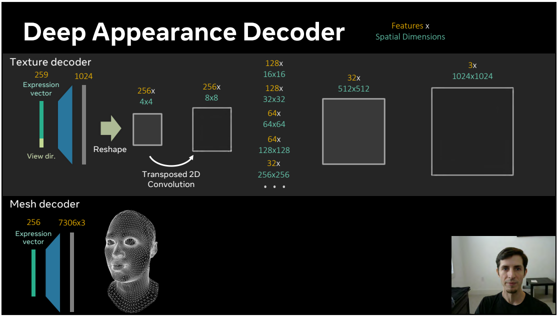

What's Wrong with Learning in Image Space?
View 1
View 2
New View
View 3
Interpolate

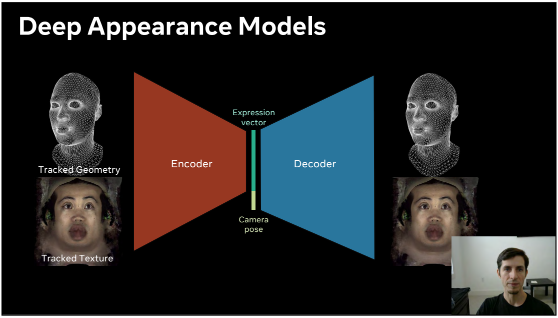


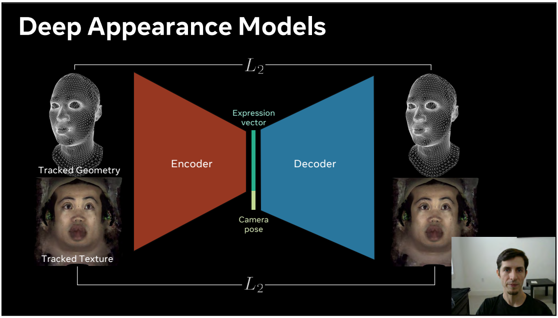

View 1
View 2
View 3

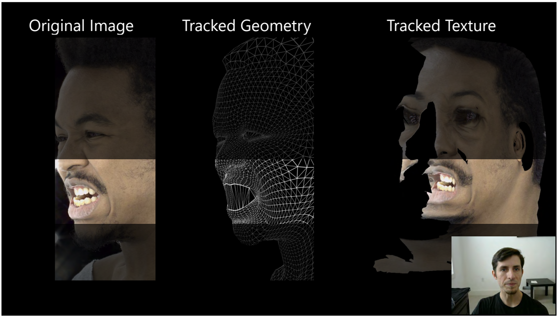

View 1
View 2
View 3

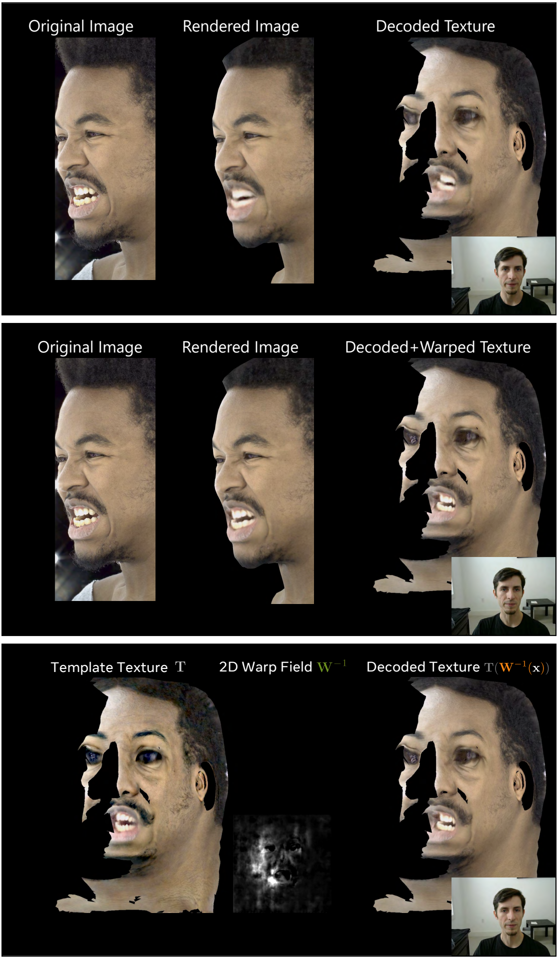

Deep Appearance Models
Expression vector
Decoder
Camera pose

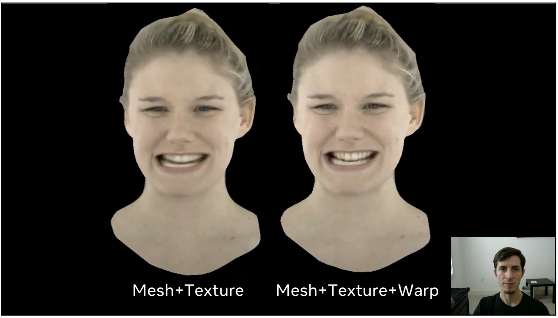

Deep Appearance Decoder
Features x
Spatial Dimensions
Texture decoder
259
Expression vector
View dir.
1024
Reshape
256x
4x4
256x
8x8
Transposed 2D Convolution
128x
16x16
128x
32x32
64x
64x64
64x
128x128
32x
256x256
32x
512x512
3x
1024x1024
Mesh decoder
256
Expression vector
7306x3


Deep Appearance Models
Tracked Geometry
Tracked Texture
Encoder
Expression vector
Decoder
Camera pose

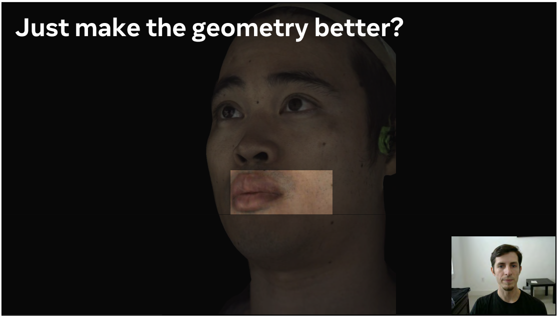

Deep Appearance Models
L_2
Tracked Geometry
Encoder
Expression vector
Decoder
Camera pose
Tracked Texture
L_2

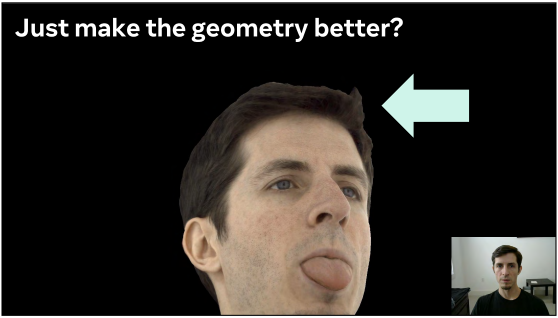

Original Image
Tracked Geometry
Tracked Texture

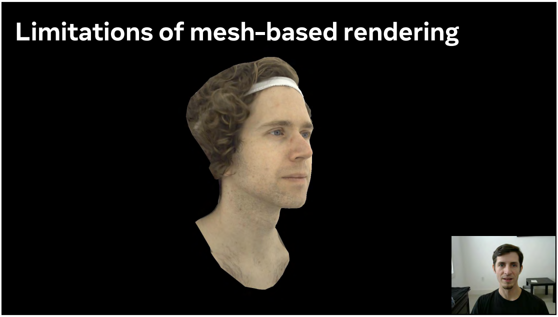

Original Image
Rendered Image
Decoded Texture
Original Image
Rendered Image
Decoded+Warped Texture
Template Texture T
2D Warp Field $\mathbf{W}^{-1}$
Decoded Texture $\mathrm{T}(\mathbf{W}^{-1}(\mathbf{x}))$

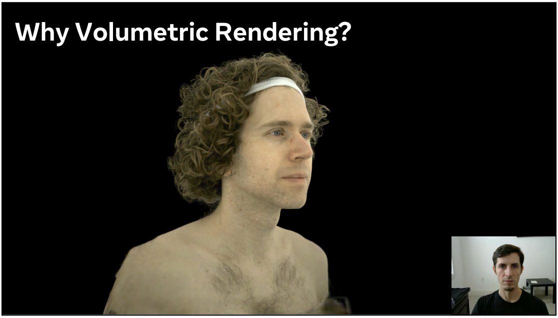

Mesh+Texture
Mesh+Texture+Warp

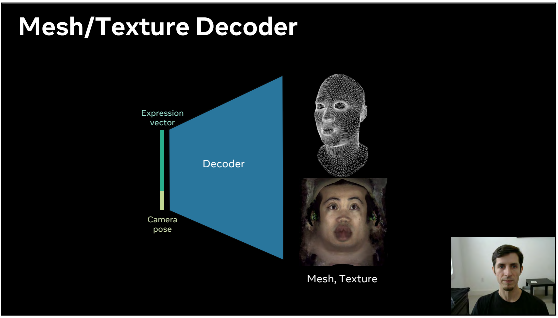

Without Warp
With Warp

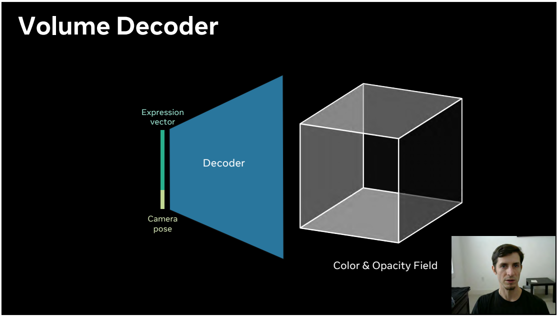

Just make the geometry better?

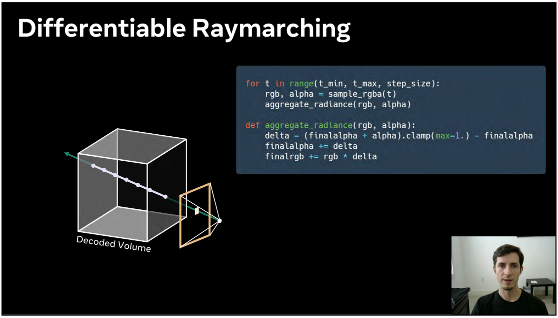

Just make the geometry better?

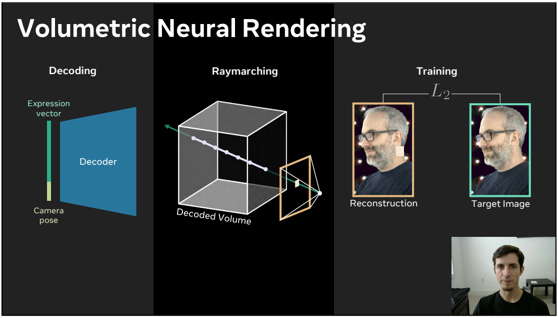

Limitations of mesh-based rendering

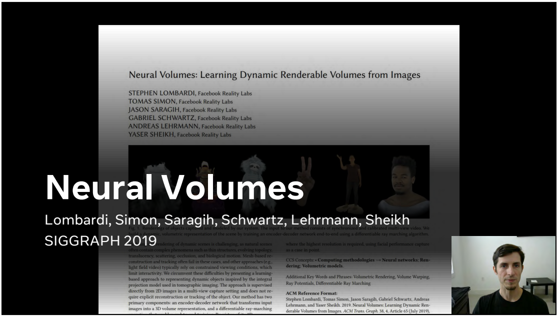

Why Volumetric Rendering?

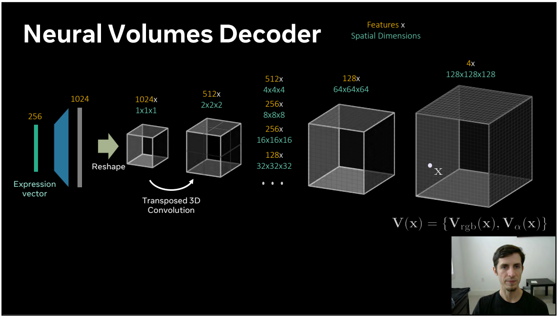

Mesh/Texture Decoder
Expression vector
Decoder
Camera pose
Mesh, Texture

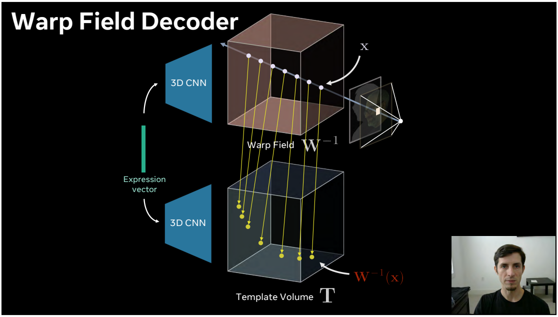

Volume Decoder
Expression vector
Decoder
Camera pose
Color & Opacity Field

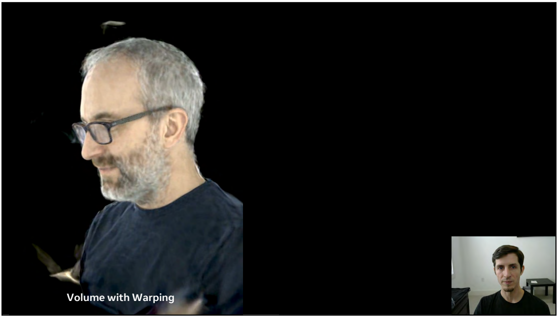

Differentiable Raymarching
for t in range(t_min, t_max, step_size):
    rgb, alpha = sample_rgba(t)
    aggregate_radiance(rgb, alpha)
def aggregate_radiance(rgb, alpha):
    delta = (finalalpha + alpha).clamp(max=1.) - finalalpha
    finalalpha += delta
    finalrgb += rgb * delta
Decoded Volume

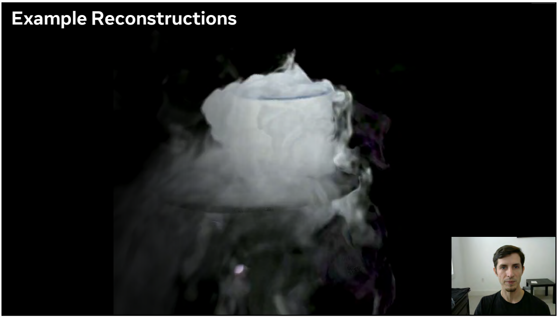

Volumetric Neural Rendering
Decoding
Expression vector
Decoder
Camera pose
Raymarching
Decoded Volume
Training
L_2
Reconstruction
Target Image

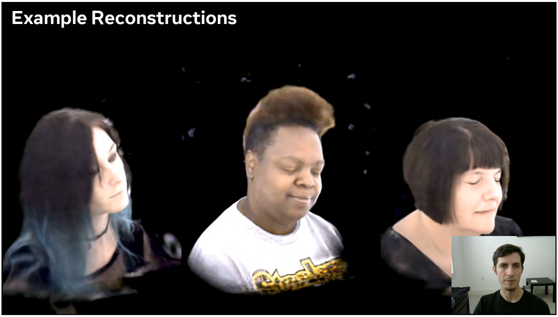

Neural Volumes: Learning Dynamic Renderable Volumes from Images
STEPHEN LOMBARDI, Facebook Reality Labs
TOMAS SIMON, Facebook Reality Labs
JASON SARAGIH, Facebook Reality Labs
GABRIEL SCHWARTZ, Facebook Reality Labs
ANDREAS LEHRMANN, Facebook Reality Labs
YASER SHEIKH, Facebook Reality Labs
Neural Volumes
Lombardi, Simon, Saragih, Schwartz, Lehrmann, Sheikh
SIGGRAPH 2019


Neural Volumes Decoder
Features x
Spatial Dimensions
256
1024
Expression vector
Reshape
1024x
1x1x1
512x
2x2x2
Transposed 3D Convolution
512x
4x4x4
256x
8x8x8
256x
16x16x16
128x
32x32x32
...
128x
64x64x64
4x
128x128x128
x
$V(x) = \{V_{rgb}(x), V_{\alpha}(x)\}$

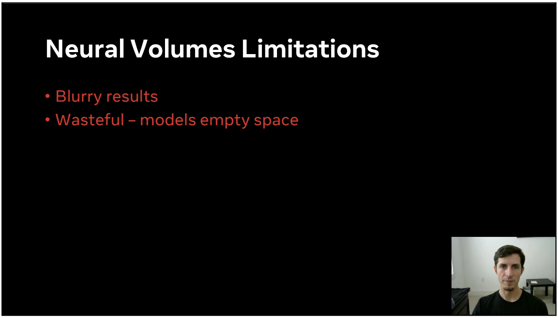

Warp Field Decoder
3D CNN
x
Warp Field W^{-1}
Expression vector
3D CNN
$W^{-1}(x)$
Template Volume T


Volume with Warping

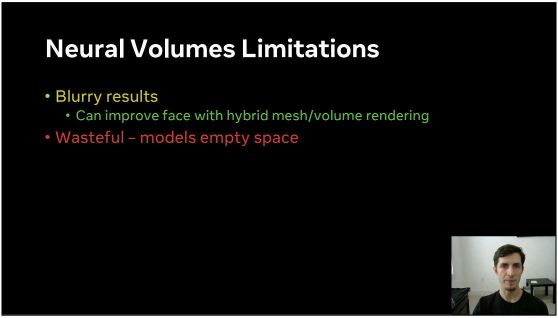

Example Reconstructions

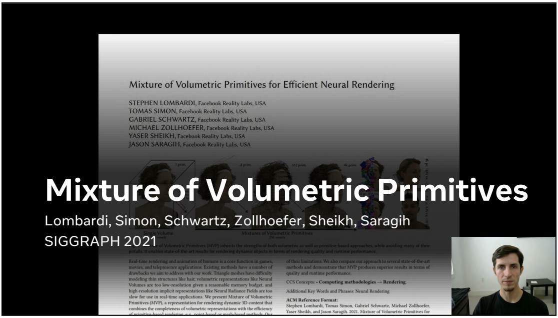

Example Reconstructions

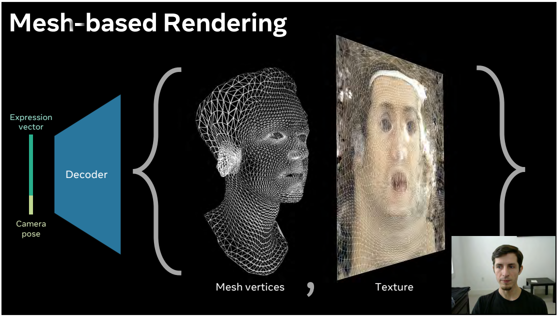

Example Reconstructions

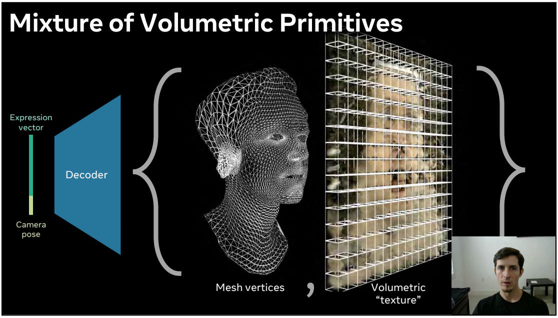

Neural Volumes Limitations
• Blurry results
• Wasteful – models empty space

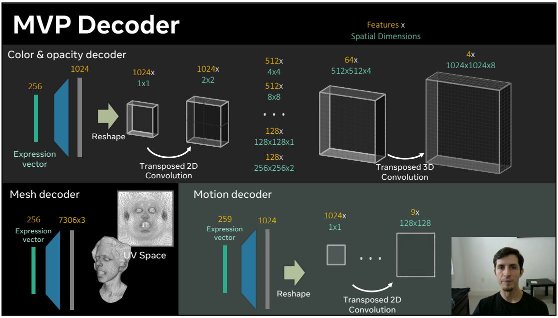

Hybrid Rendering with 3D Meshes

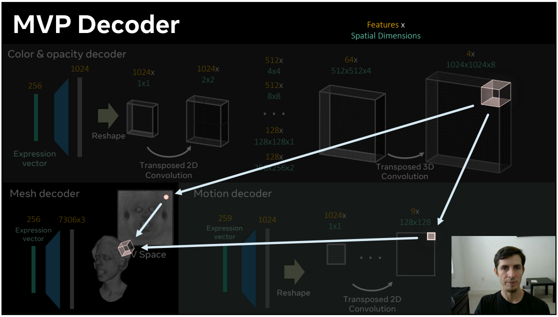

Neural Volumes Limitations
• Blurry results
• Can improve face with hybrid mesh/volume rendering
• Wasteful – models empty space

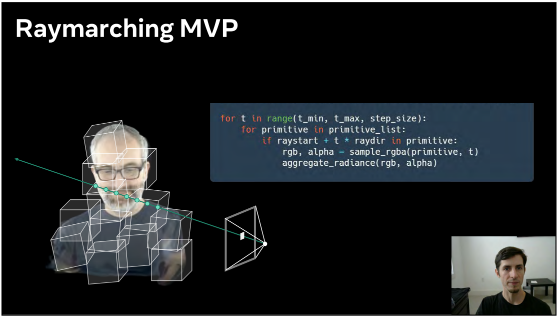

Mixture of Volumetric Primitives for Efficient Neural Rendering
STEPHEN LOMBARDI, Facebook Reality Labs, USA
TOMAS SIMON, Facebook Reality Labs, USA
GABRIEL SCHWARTZ, Facebook Reality Labs, USA
MICHAEL ZOLLHOEFER, Facebook Reality Labs, USA
YASER SHEIKH, Facebook Reality Labs, USA
JASON SARAGIH, Facebook Reality Labs, USA
Mixture of Volumetric Primitives
Lombardi, Simon, Schwartz, Zollhoefer, Sheikh, Saragih
SIGGRAPH 2021

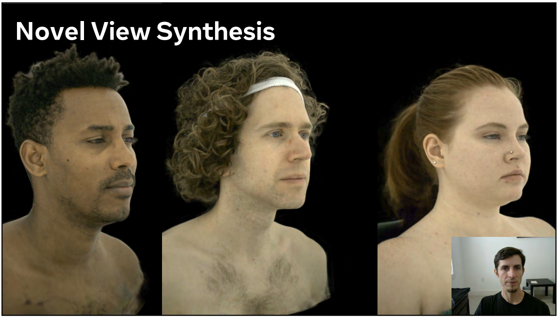

Mesh-based Rendering
Expression vector
Camera pose
Decoder
Mesh vertices
,
Texture


Mixture of Volumetric Primitives
Expression vector
Camera pose
Decoder
Mesh vertices
,
Volumetric “texture”

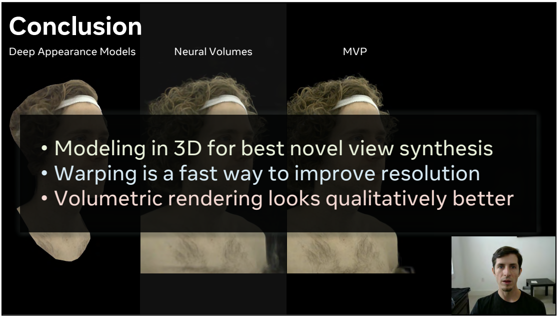

MVP Decoder
Features x
Spatial Dimensions
Color & opacity decoder
256
1024
Expression
vector
Reshape
1024x
1x1
1024x
2x2
Transposed 2D
Convolution
512x
4x4
512x
8x8
. . .
128x
128x128x1
128x
256x256x2
64x
512x512x4
4x
1024x1024x8
Transposed 3D
Convolution
Mesh decoder
256
Expression
vector
7306x3
UV Space
Motion decoder
259
Expression
vector
1024
Reshape
1024x
1x1
. . .
9x
128x128
Transposed 2D
Convolution

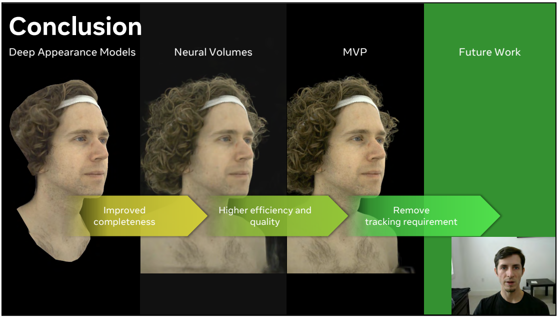

MVP Decoder
Features x
Spatial Dimensions
Color & opacity decoder
256
1024
Expression
vector
Reshape
1024x
1x1
1024x
2x2
Transposed 2D
Convolution
512x
4x4
512x
8x8
. . .
128x
128x128x1
64x
512x512x4
4x
1024x1024x8
Transposed 3D
Convolution
Mesh decoder
256
Expression
vector
7306x3
Motion decoder
259
Expression
vector
1024
Reshape
1024x
1x1
. . .
9x
128x128
Transposed 2D
Convolution


Raymarching MVP
for t in range(t_min, t_max, step_size):
for primitive in primitive_list:
if raystart + t * raydir in primitive:
rgb, alpha = sample_rgba(primitive, t)
aggregate_radiance(rgb, alpha)


Novel View Synthesis


Results


Conclusion
Deep Appearance Models
Neural Volumes
MVP
• Modeling in 3D for best novel view synthesis
• Warping is a fast way to improve resolution
• Volumetric rendering looks qualitatively better


Conclusion
Deep Appearance Models
Neural Volumes
MVP
Future Work
Improved completeness
Higher efficiency and quality
Remove tracking requirement

Neural Rendering for Dynamic Performance Capture

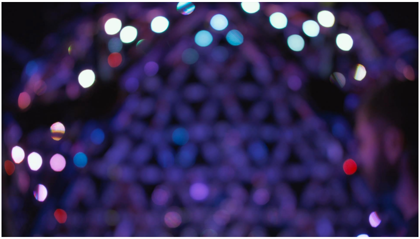
Rohit K. Pandey
rohitpandey@google.com

Google

Outline

- Dynamic performance capture
- LookinGood (*SIGGRAPH Asia 2018*)
- SimplyLookGood a.k.a. Volumetric Capture of Humans with a Single RGBD Camera via Semi-Parametric Learning (*CVPR 2019*)
- LookinGood++
- Deep Relightable Textures a.k.a. Deep Relightables (*SIGGRAPH Asia 2020*)
- In the wild applications
- Future directions

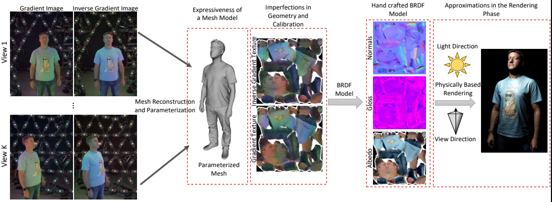
Dynamic Performance Capture



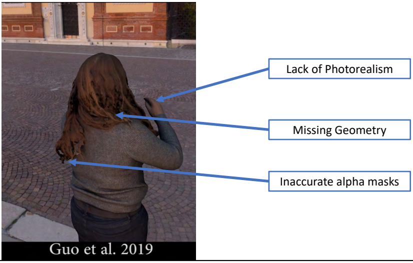
Guo et al. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting – SIGGRAPH Asia 2019

Traditional Capture and Reconstruction Pipeline

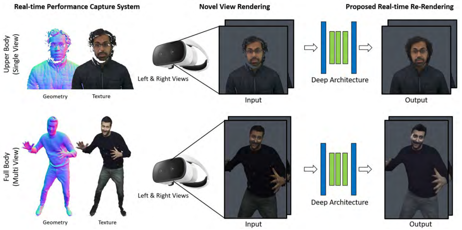
Guo et al. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting – SIGGRAPH Asia 2019



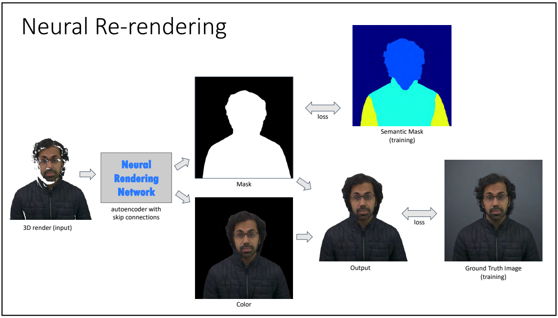
Caveats of the Geometric Pipeline

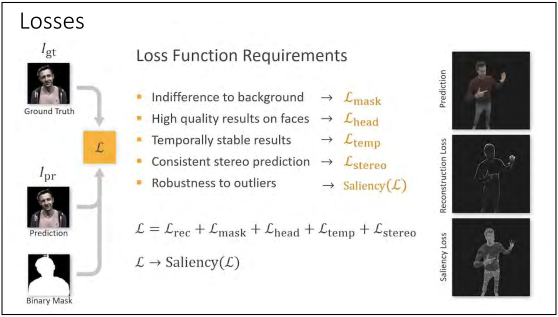


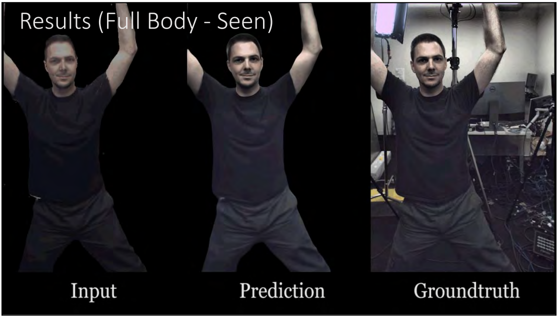
LookinGood

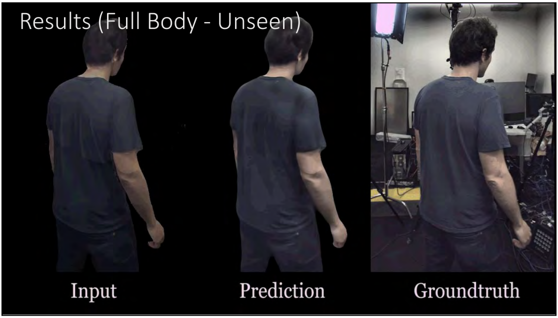


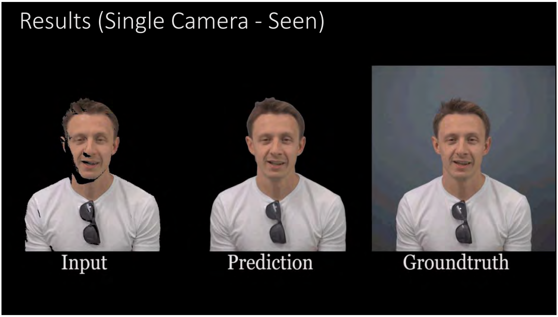
Martin-Brualla, Ricardo, et al. "LookinGood: enhancing performance capture with real-time neural re-rendering." SIGGRAPH Asia 2018

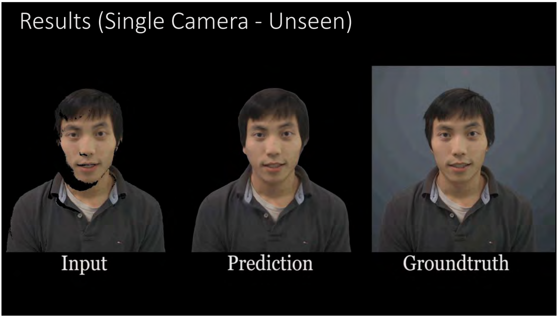

Neural Re-rendering
3D render (input)
Neural Rendering Network
autoencoder with skip connections
Mask
loss
Semantic Mask (training)
Color
Output
loss
Ground Truth Image (training)


Losses
I_{gt}
Ground Truth
$\mathcal{L}$
I_{pr}
Prediction
Binary Mask
Loss Function Requirements
Indifference to background → $\mathcal{L}_{\mathrm{mask}}$
High quality results on faces → $\mathcal{L}_{\mathrm{head}}$
Temporally stable results → $\mathcal{L}_{\mathrm{temp}}$
Consistent stereo prediction → $\mathcal{L}_{\mathrm{stereo}}$
Robustness to outliers → Saliency($\mathcal{L}$)
$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{mask}} + \mathcal{L}_{\mathrm{head}} + \mathcal{L}_{\mathrm{temp}} + \mathcal{L}_{\mathrm{stereo}}$
$\mathcal{L} \rightarrow \mathrm{Saliency}(\mathcal{L})$
Prediction
Reconstruction Loss
Saliency Loss


Results (Full Body - Seen)
Input
Prediction
Groundtruth


Results (Full Body - Unseen)
Input
Prediction
Groundtruth


Results (Single Camera - Seen)
Input
Prediction
Groundtruth


Results (Single Camera - Unseen)
Input
Prediction
Groundtruth

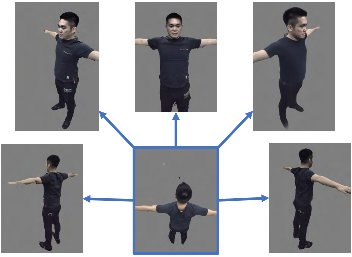
SimplyLookinGood

a.k.a. Volumetric Capture of Humans with a Single RGBD Camera via Semi-Parametric Learning

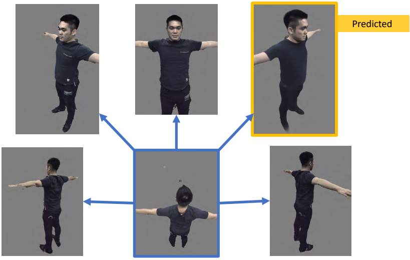
- Goal
  - Drop infrastructure and make volumetric capture more accessible to users
- Approach
  - Reduce multi-view setup to a single camera
  - Use a previously seen calibration sequence to infer parts of the user not visible in the current frame

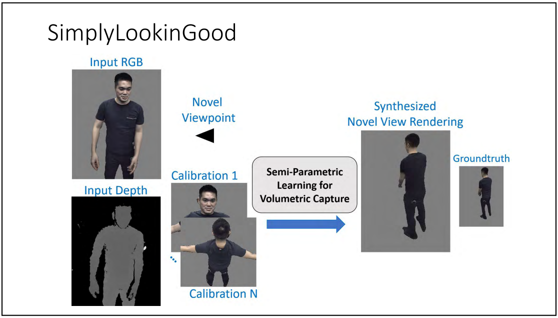
Pandey, Rohit, et al. "Volumetric Capture of Humans with a Single RGBD Camera via Semi-Parametric Learning." *CVPR 2019*

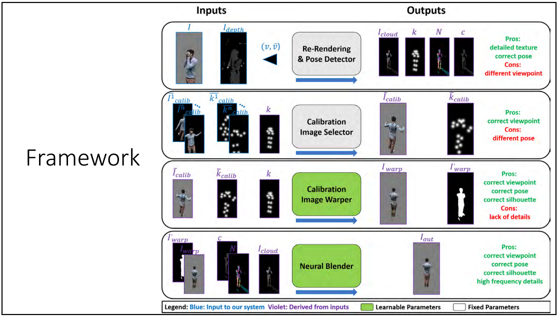
Single Camera Volumetric Capture

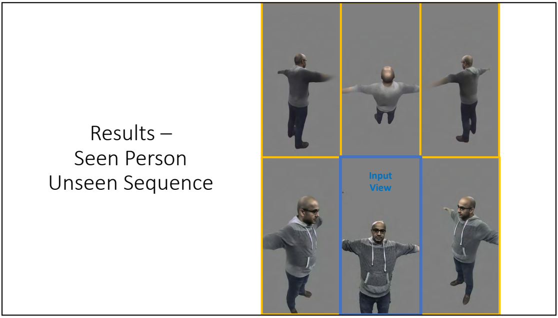


Single Camera Volumetric Capture

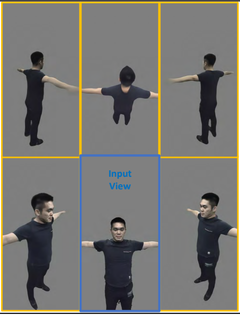



SimplyLookinGood
Input RGB
Novel Viewpoint
Synthesized Novel View Rendering
Groundtruth
Calibration 1
Semi-Parametric Learning for Volumetric Capture
Input Depth
Calibration N

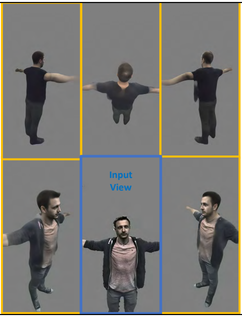

Framework
Inputs
Outputs
Re-Rendering & Pose Detector
Pros: detailed texture correct pose
Cons: different viewpoint
Calibration Image Selector
Pros: correct viewpoint
Cons: different pose
Calibration Image Warper
Pros: correct viewpoint correct pose correct silhouette
Cons: lack of details
Neural Blender
Pros: correct viewpoint correct pose correct silhouette high frequency details
Legend: Blue: Input to our system Violet: Derived from inputs
Learnable Parameters
Fixed Parameters


Results – Seen Person Unseen Sequence
Input View

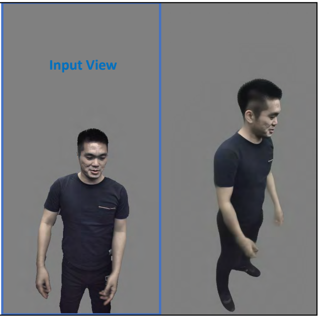
Results –
Seen Person
Unseen Sequence



Results –
Unseen Person
Unseen Sequence



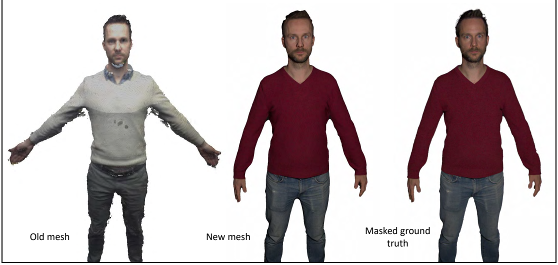
Viewpoint
Generalization



The Switch to Better Data

- Old Dataset
  - 1MP Camera resolution
  - 8 cameras
  - ~15 subjects
  - Real-time reconstruction method (Motion2Fusion, Dou et al. 2017)
- New Dataset
  - 12 MP Camera resolution
  - ~60 cameras
  - ~70 subjects
  - Offline reconstruction method (The Relightables, Guo et al. 2019)

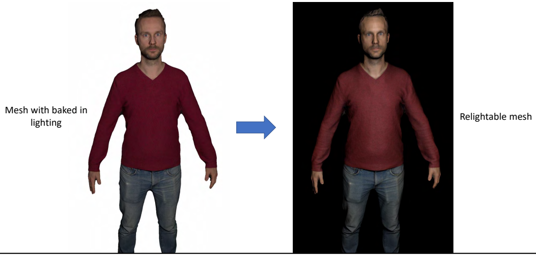
Christoph Then and Now



LookinGood++

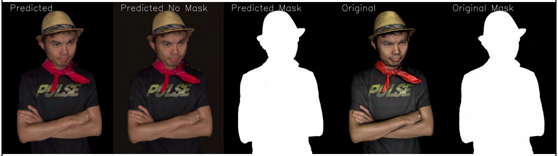
- LookinGood on better input meshes and better GT
  - Works but relighting is not possible
- LookinGood + relighting
  - Requires mesh rendered in target lighting condition
  - Relightable meshes need to come from gradient images in order to support dynamic capture
    - An OLAT sequence takes ~6 seconds to capture and the person cannot move
  - Non photorealistic rendering compared to baked in lighting
    - Handcrafted BRDF (cosine lobe assumption)
    - Approximated rendering (no subsurface scattering, global illumination etc.)

Loss of Photorealism in Relightable Mesh



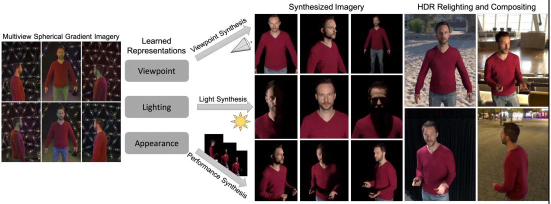
Seen Person Results

Unseen Person Results

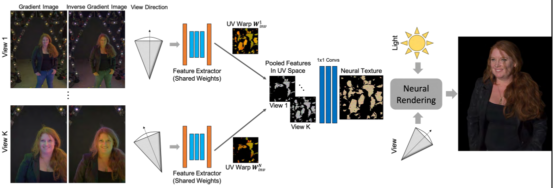


Generalization is difficult!

Deep Relightables



Deep Relightables Pipeline



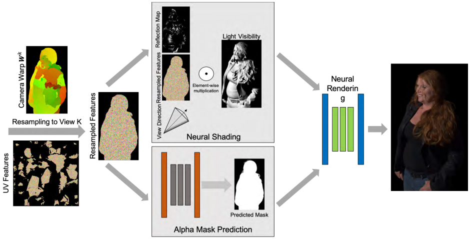
Feature Extractor

- Gradient image input
  - Albedo information
  - Normals information
  - View dependent effects information
  - Enables moving performers
- View Direction input
  - Enables the features to reason about view dependent effects
- Receptive field of features
  - Enables the features to reason about pixels that the geometry did not capture

Regressed Neural Texture

- Neural UV features
  - Regressed from input images and thus support generalization to unseen people and poses
  - Supports different resolutions at train and test time
- Feature pooling and 1x1 Convolutions
  - Enable feature extraction from a varying number of viewpoints
    - Factors in camera visibility to weigh front facing pixels higher
    - Independent of the order of input viewpoints
    - Enables training with few views but evaluating with all available views
    - Enables running feature extraction only once during eval for novel view synthesis
  - Supports changing UV parametrization
    - Warps are precomputed based on geometry
    - 1x1 convs do not see the spatial extent of the UV texture
    - Enables moving performers

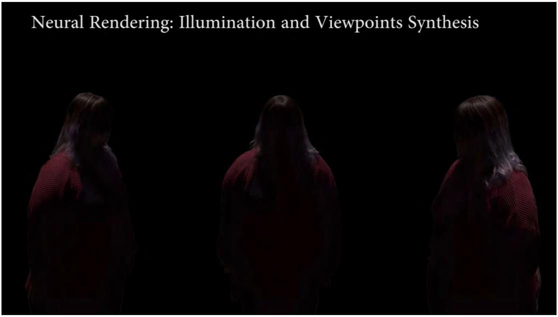
Neural Re-Renderer

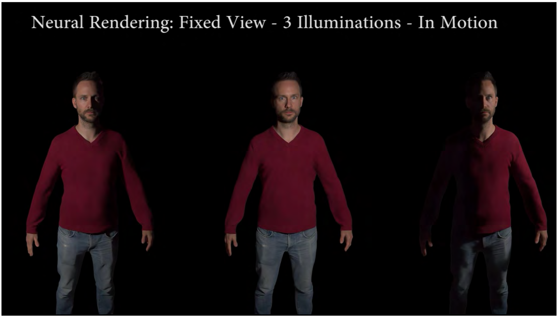


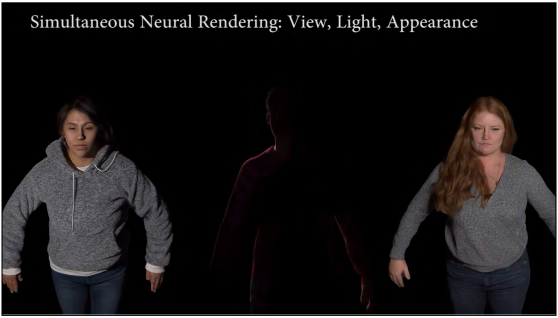
Re-renderer Details

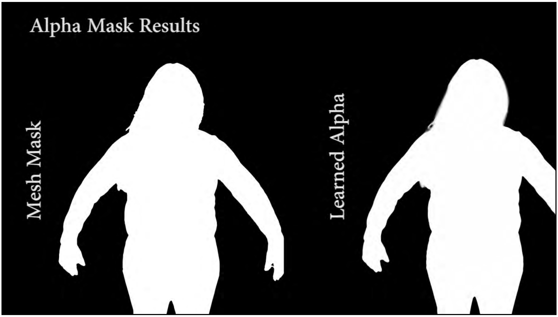
- View direction input
  - Enables reasoning of view dependent effects
- Light visibility input
  - Improves the quality and generalization of self occlusions
- Reflection map input
  - Improves the quality and generalization of view dependent effects
- RGB output at 1x1 conv level
  - Improves the generalization of RGB colors
- Alpha mask prediction
  - Enables high quality compositing with HDRI backgrounds
- Perceptual losses on final re-rendered RGB output
  - With optional adversarial finetuning


Neural Rendering: View Synthesis
PULSE

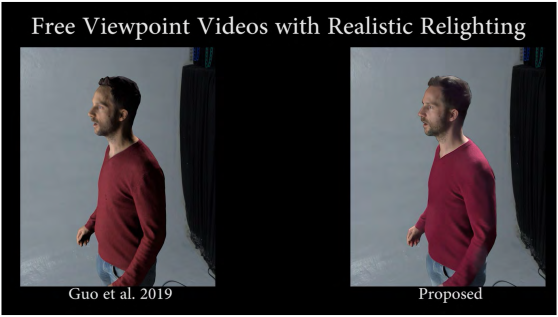

Neural Rendering: Illumination and Viewpoints Synthesis


Neural Rendering: Fixed View - 3 Illuminations - In Motion


Simultaneous Neural Rendering: View, Light, Appearance


Alpha Mask Results
Mesh Mask
Learned Alpha


View Synthesis Comparisons
Martin-Brualla et al. 2018
Thies et al. 2019
Proposed



Neural Rendered Outputs as GT Data

- Data augmentation via,
  - More viewpoints from free view synthesis
  - Arbitrary lighting conditions
  - Arbitrary backgrounds via alpha compositing

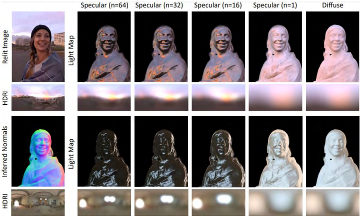
- Better than state-of-the-art geometric pipelines in terms of,
  - Photorealism
  - Geometry
  - Alpha mattes

In The Wild Applications using Neural Rendered Data

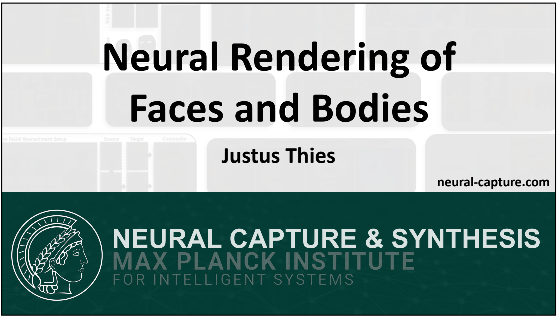
- Relighting
- Light estimation
- Alpha matting
- Depth estimation
- Pose estimation

Extensions and Future Directions

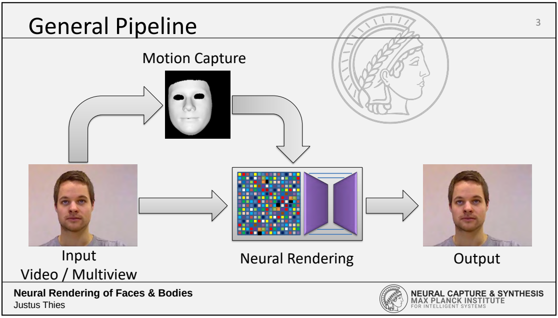
- Predicting relit images directly instead of OLATs
- Depth and correspondence prediction
- Pose and appearance control

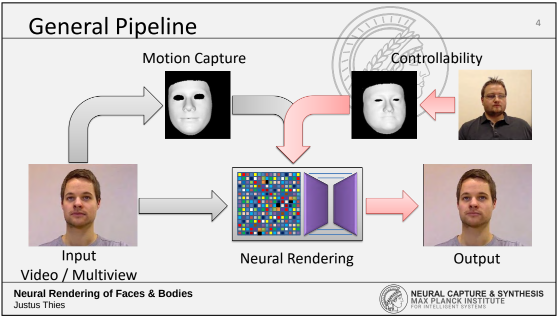


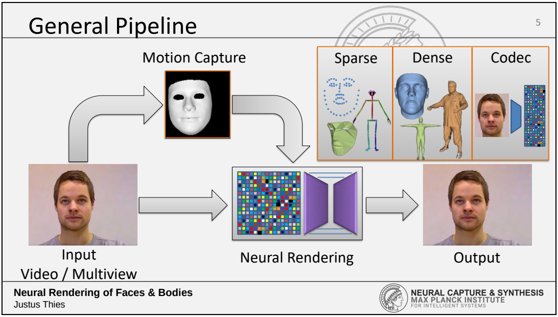
Pandey, Rohit, et al. "Total Relighting: Learning to Relight Portraits for Background Replacement" SIGGRAPH 2021

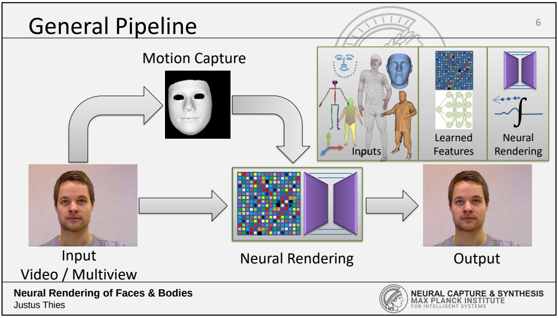

Neural Rendering of
Faces and Bodies
Justus Thies
neural-capture.com
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

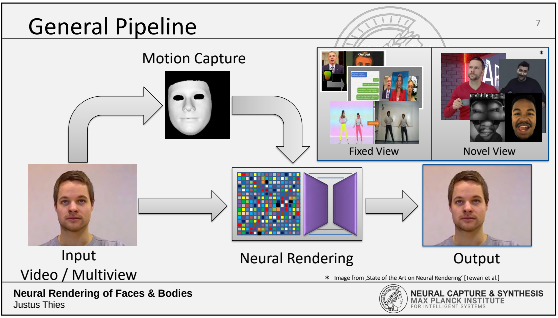

Faces
Bodies
Audio Input
Live Facial Reenactment Setup
Input RGB-D Sequence
Dynamic Surface Function Network
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


General Pipeline
3
Motion Capture
Input
Video / Multiview
Neural Rendering
Output
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

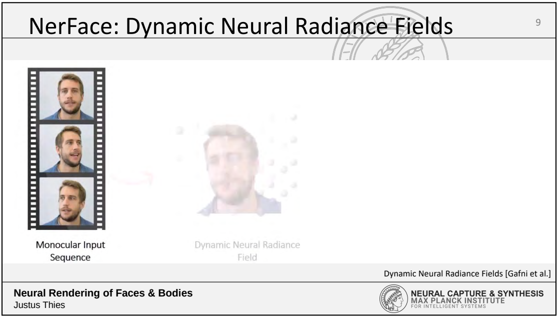

General Pipeline
4
Motion Capture
Controllability
Input
Video / Multiview
Neural Rendering
Output
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

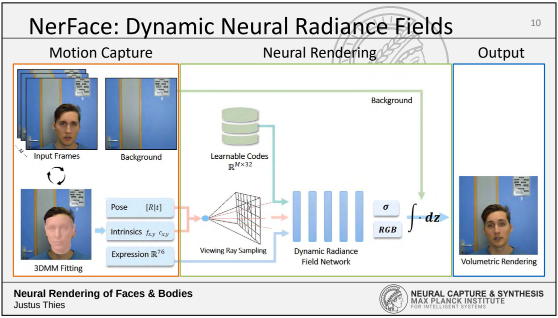

General Pipeline
5
Motion Capture
Sparse
Dense
Codec
Input
Video / Multiview
Neural Rendering
Output
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

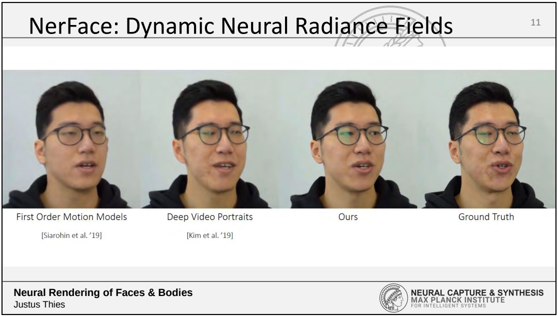

General Pipeline
6
Motion Capture
Inputs
Learned
Features
Neural
Rendering
Input
Video / Multiview
Neural Rendering
Output
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

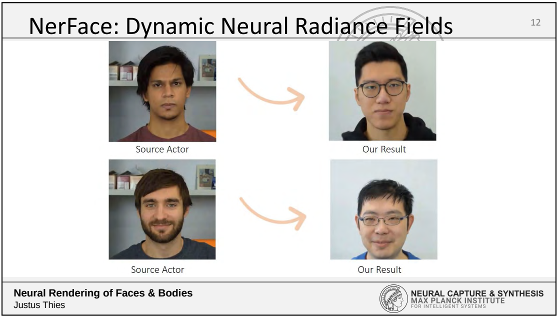

General Pipeline
7
Motion Capture
Fixed View
Novel View
Input
Video / Multiview
Neural Rendering
Output
* Image from ‚State of the Art on Neural Rendering' [Tewari et al.]
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


Faces
Bodies
Live Facial Reenactment Setup
Dynamic Surface Function Network
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


NerFace: Dynamic Neural Radiance Fields
9
Monocular Input
Sequence
Dynamic Neural Radiance
Field
Dynamic Neural Radiance Fields [Gafni et al.]
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

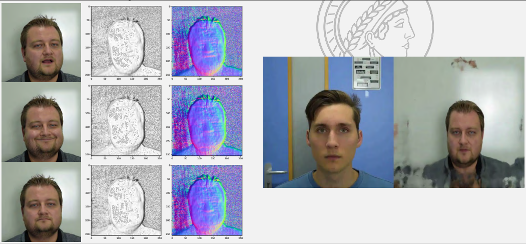

NerFace: Dynamic Neural Radiance Fields
10
Motion Capture
Neural Rendering
Output
Input Frames
Background
Background
Learnable Codes
$\mathbb{R}^{M\times 32}$
3DMM Fitting
Pose $[R|t]$
Intrinsics $f_{x,y}$ $c_{x,y}$
Expression $\mathbb{R}^{76}$
Viewing Ray Sampling
Dynamic Radiance
Field Network
σ
RGB
$\int \cdot dz$
Volumetric Rendering
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


NerFace: Dynamic Neural Radiance Fields
11
First Order Motion Models
[Siarohin et al. '19]
Deep Video Portraits
[Kim et al. '19]
Ours
Ground Truth
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


NerFace: Dynamic Neural Radiance Fields
12
Source Actor
Our Result
Source Actor
Our Result
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

NerFace: Dynamic Neural Radiance Fields

13



Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

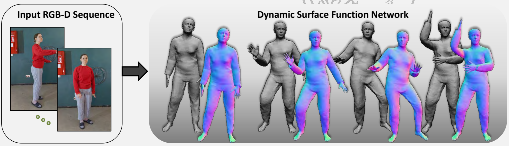
Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Dynamic Surface Function Networks

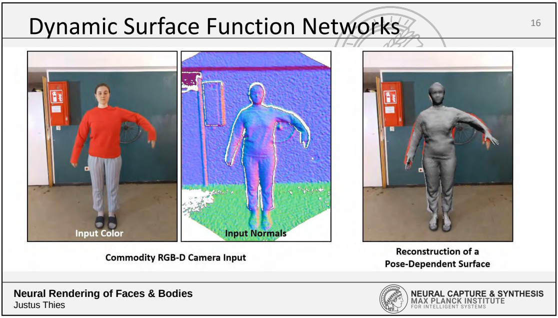
15

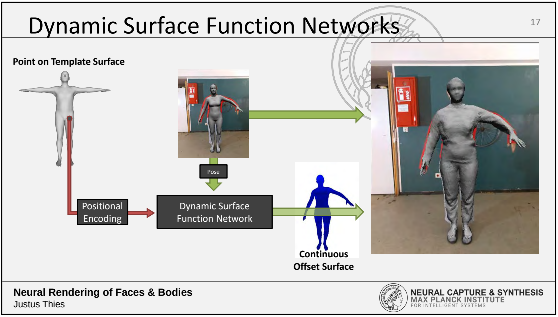
Dynamic Surface Function Networks [Burov et al.]

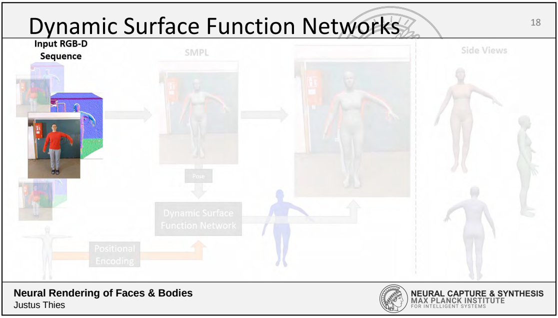


Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

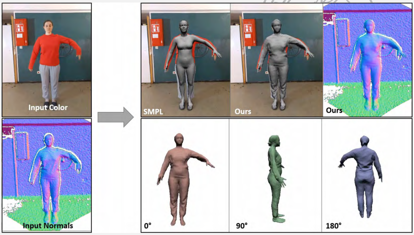

Dynamic Surface Function Networks
16
Input Color
Input Normals
Commodity RGB-D Camera Input
Reconstruction of a
Pose-Dependent Surface
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


Dynamic Surface Function Networks
17
Point on Template Surface
Pose
Positional
Encoding
Dynamic Surface
Function Network
Continuous
Offset Surface
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


Dynamic Surface Function Networks
18
Input RGB-D
Sequence
SMPL
Side Views
Pose
Dynamic Surface
Function Network
Positional
Encoding
Neural Rendering of Faces & Bodies
Justus Thies
NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Dynamic Surface Function Networks

19

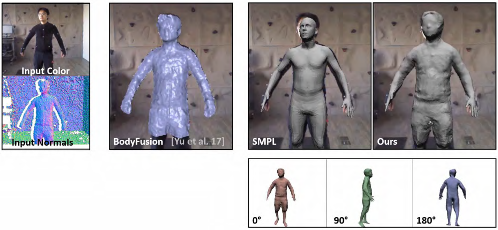


Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Dynamic Surface Function Networks

20

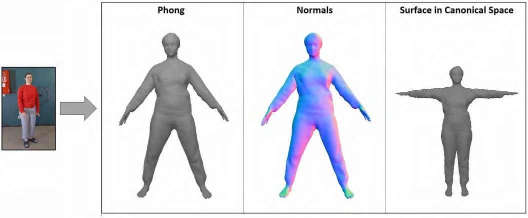


Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

Dynamic Surface Function Networks

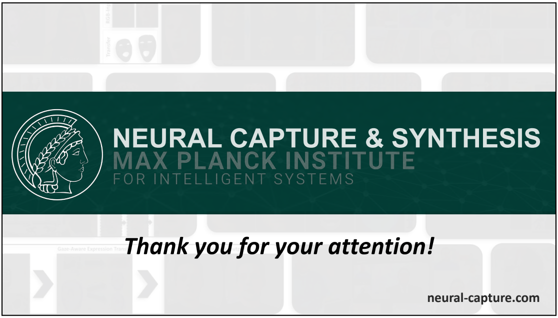
21



‚Jumping Jacks‘ Motion - MoSh [Loper et al. 14]

Neural Rendering of Faces & Bodies
Justus Thies

NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS


NEURAL CAPTURE & SYNTHESIS
MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS
Thank you for your attention!
neural-capture.com

5/26/2021

Neural Rendering and Video-based Animation of Human Actors

Christian Theobalt

Visual Computing and AI Department

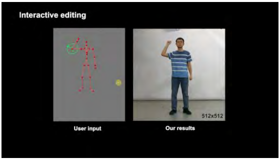
MPI for Informatics

Saarbruecken, Germany

max planck institut informatik

Christian Theobalt

Neural Rendering of Human Actors – New Possibilities



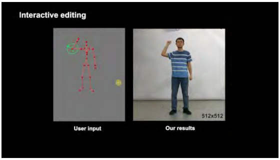
Liu et al., Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation, **IEEE TVCG 2021**

max planck institut informatik

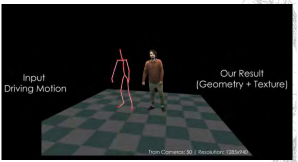
Christian Theobalt

5/26/2021

Neural Rendering of Human Actors – New Possibilities



Liu et al., Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation, **IEEE TVCG 2021**



Habermann et al., Real-time Deep Dynamic Characters, **SIGGRAPH 2021**

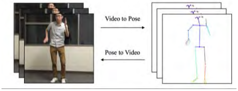
max planck institut informatik

Christian Theobalt

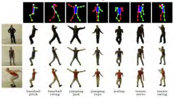
Related Work

- Single view and **sparse** image-based methods

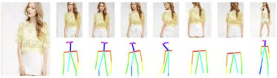
Chan et al. 2019 (ICCV)



Esser et al. 2018 (CVPR)



Pumarola et al. 2018 (CVPR)



(Si et al. 2018 (CVPR), Ma et al. 2018 (CVPR, Aberman et al. 2019 (EG), Siarohin 2019 (NeurIPS)…)

max planck institut informatik

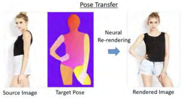
Christian Theobalt

5/26/2021

Related Work

- Single view and **dense** image-based methods

Liu et al. 2019 (ToG)



Sarkar et al. 2020 (ECCV)

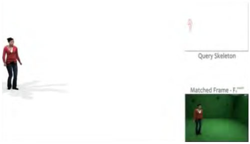
(Martin-Brualla et al. 2018 (ToG), Wang et al. 2018 (NeurIPS), Li et al. 2019 (CVPR), Liu et al. 2020 (TVCG), Sarkar et al. 2021 (arXiv)....)

max planck institut informatik

Christian Theobalt

Related Work

- Multi-view non learning-based methods



Xu et al. 2011 (SIGGRAPH)

Casas et al. 2014 (CGF)

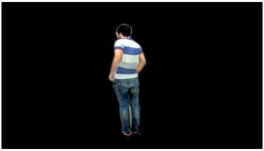
(Collet et al. 2015 (ToG), Li et al. 2014 (MVA), Zitnick et al. 2004 (ToG), Stoll et al. 2010 (ToG), Volino et al. 2014 (BMVC)...)

max planck institut informatik

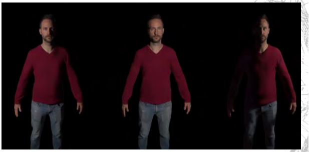
Christian Theobalt

Related Work

- Multi-view learning based methods



Shysheya et al. 2019 (CVPR)



Meka et al. 2020 (SIGASIA)

(Lombardi et al. 2019 (ToG), Sitzmann et al. 2019 (CVPR), Pumarola et al. 2020 (CVPR), Mildenhall et al. 2020 (ECCV), Habermann et al. 2021 (SIGGRAPH)....)

mpii max planck institut informatik

Christian Theobalt

High-Fidelity Neural Human Motion Transfer From Monocular Video

CVPR 2021 (Oral)

Moritz Kappel[1], Vladislav Golyanik[2], Mohamed Elgharib[2], Jann-Ole Henningson[1], Hans-Peter Seidel[2], Susana Castillo[1], Christian Theobalt[2], Marcus Magnor[1]

[1] ICG, TU Braunschweig [2] MPI for Informatics, SIC

Technische Universität Braunschweig

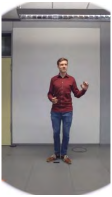
mpii max planck institut informatik

mpii max planck institut informatik

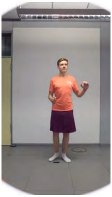
Christian Theobalt

5/26/2021

Problem Statement



Source Actor



Our Result

Christian Theobalt

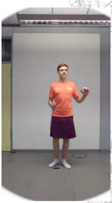
Problem Statement

Source Actor

2D Shape

Clothing Structure

Target Appearance

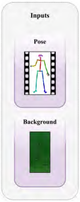


Our Result

Christian Theobalt

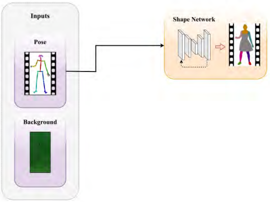
5/26/2021

Method Overview



Christian Theobalt

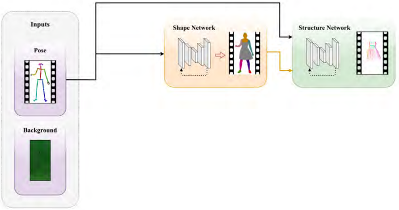
Method Overview



Christian Theobalt

5/26/2021

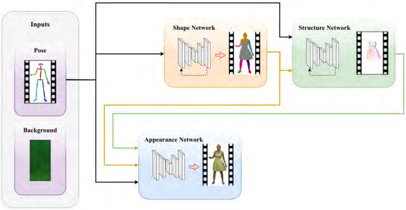
Method Overview



max planck institut informatik

Christian Theobalt

Method Overview

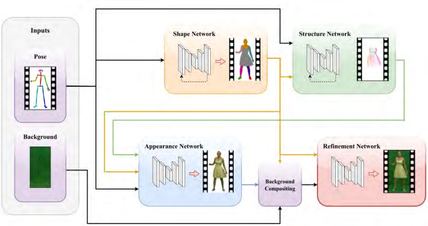


max planck institut informatik

Christian Theobalt

5/26/2021

Method Overview



max planck institut informatik

Christian Theobalt

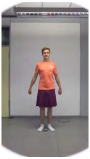
Dataset

max planck institut informatik

Christian Theobalt

5/26/2021

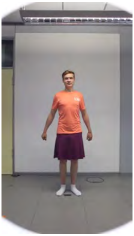
Dataset



max planck institut informatik

Christian Theobalt

Dataset



max planck institut informatik

Christian Theobalt

5/26/2021

Dataset

max planck institut informatik

Christian Theobalt

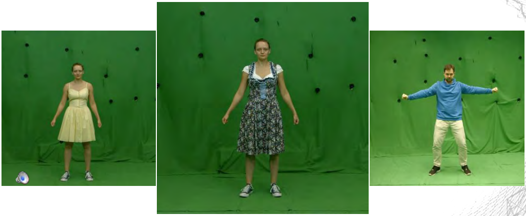
Dataset

max planck institut informatik

Christian Theobalt

5/26/2021

Dataset



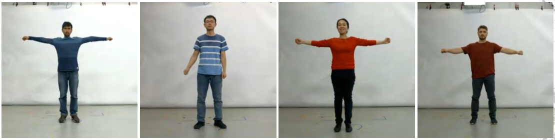
Christian Theobalt

Dataset

Christian Theobalt

5/26/2021

Dataset: Liu et al. [1]



[1] Liu, Lingjie and Xu, Weipeng and Habermann, Marc and Zollhöfer, Michael and Bernard, Florian and Kim, Hyeongwoo and Wang, Wenping and Theobalt, Christian, Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation, TVCG 2020

max planck institut informatik

Christian Theobalt

Results

Driving Motion **2D Shape** **Clothing Structure** **Target Appearance** **Final Output**

max planck institut informatik

Christian Theobalt

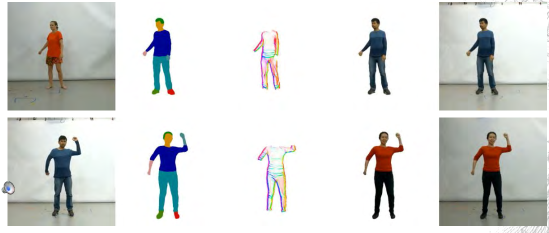
5/26/2021

Results

max planck institut informatik

Christian Theobalt

Results



max planck institut informatik

Christian Theobalt

Comparison to Related Work

Driving Motion | **Pix2PixHD [1]** | **EDN [2]** | **RecycleGAN [3]** | **Ours**

[1] Ting-Chun Wang and Ming-Yu Liu and Jun-Yan Zhu and Andrew Tao and Jan Kautz and Bryan Catanzaro, High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, CVPR 2018
[2] Chan, Caroline and Ginosar, Shiry and Zhou, Tinghui and Efros, Alexei A, Everybody Dance Now ICCV 2019
[3] Bansal, Aayush and Ma, Shugao and Ramanan, Deva and Sheikh, Yaser, Recycle-gan: Unsupervised video retargeting, ECCV 2018

Comparison to Related Work

Pix2PixHD [Wang et al. CVPR'18] | **Ours**

5/26/2021

Comparison to Related Work

RecycleGAN [Bansal et al. ECCV'18] Ours

max planck institut informatik

Christian Theobalt

Comparison to Related Work

Driving Motion EDN [1] Liu et al. [2] Ours

[1] Chan et al., ICCV 2019 [2] Liu et al., TVCG 2020

max planck institut informatik

Christian Theobalt

5/26/2021

Comparison to Related Work

Driving Motion **EDN [1]** **Liu et al. [2]** **Ours**

[1] Chan et al., ICCV 2019 [2] Liu et al., TVCG 2020

max planck institut informatik

Christian Theobalt

Application - Garment Style Transfer

Driving Motion + Shape + Structure **Output** **Original Style**

max planck institut informatik

Christian Theobalt

5/26/2021

Application - Garment Style Transfer

Driving Motion + Shape + Structure **Output** **Original Style**

Christian Theobalt

Application - Wrinkle Intensity Manipulation

0.5 x Structure **Original Structure** **2.0 x Structure**

Christian Theobalt

5/26/2021

More Results

Driving Motion

Final Result (0.33x speed)

Christian Theobalt

Real-time Deep Dynamic Characters

SIGGRAPH 2021

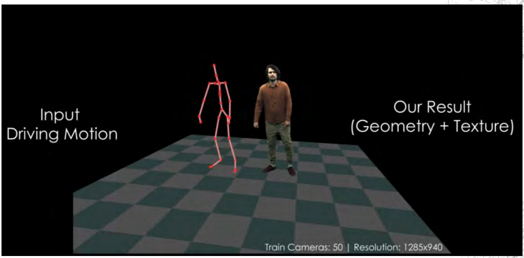
Marc Habermann[1] Lingjie Liu[1] Weipeng Xu[2] Michael Zollhoefer[2]
Gerard Pons-Moll[1] Christian Theobalt[1]

MPI for Informatics, SIC Facebook Reality Labs

Christian Theobalt

5/26/2021

Introduction



mpii max planck institut informatik

Christian Theobalt

Introduction

Input driving motion

mpii max planck institut informatik

Christian Theobalt

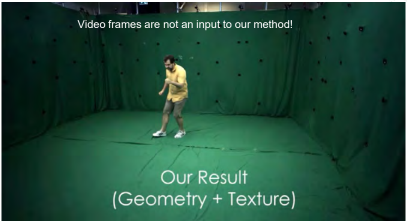
5/26/2021

Introduction

Our predicted geometry

Christian Theobalt

Introduction



Christian Theobalt

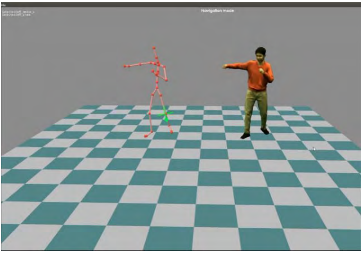
5/26/2021

Applications - Video Synthesis

max planck institut informatik

Christian Theobalt

Applications - Interactive Character Editing

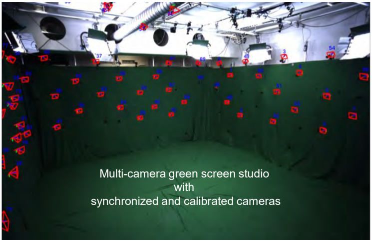


max planck institut informatik

Christian Theobalt

5/26/2021

Data Acquisition



max planck institut informatik

Christian Theobalt

Data Acquisition

Multi-view images

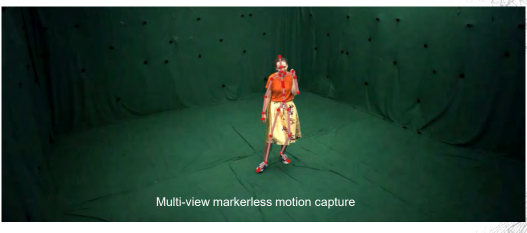
Foreground masks

max planck institut informatik

Christian Theobalt

5/26/2021

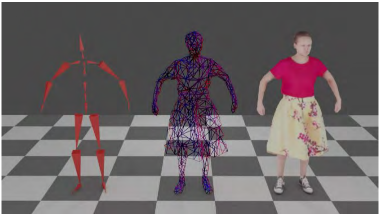
Data Acquisition



Per-frame skeletal pose $\mathcal{S}_f = \{\boldsymbol{\theta}_f, \boldsymbol{\alpha}_f, \mathbf{z}_f\}$

Christian Theobalt

Character Representation



Skeleton Embedded graph Textured template mesh

Christian Theobalt

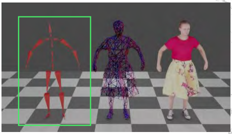
5/26/2021

Character Representation

Embedded Graph / Skeleton / Template Mesh

$\mathcal{S} = \{\boldsymbol{\theta} \in \mathbb{R}^{57}, \boldsymbol{\alpha}, \mathbf{z}\}$ $\mathcal{G} = \{\mathbf{G}, \mathbf{A}, \mathbf{T} \in \mathbb{R}^{K\times 3}\}$ $\mathbf{V} \in \mathbb{R}^{N\times 3}$

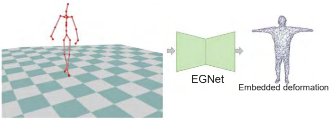
Joint angles · Undeformed graph node position · Rest vertex positions · Root rotation · Node rotation parameters (Euler angles) · Root translation · Node translation parameters



max planck institut informatik

Christian Theobalt

Method Overview

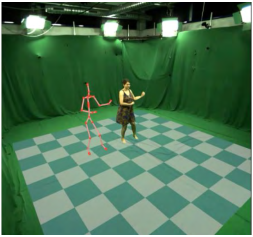


max planck institut informatik

Christian Theobalt

57

Results



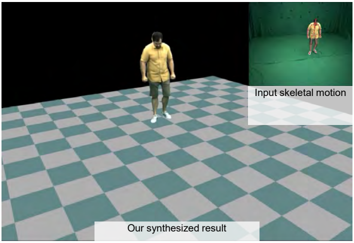
Christian Theobalt

Applications

Christian Theobalt

5/26/2021

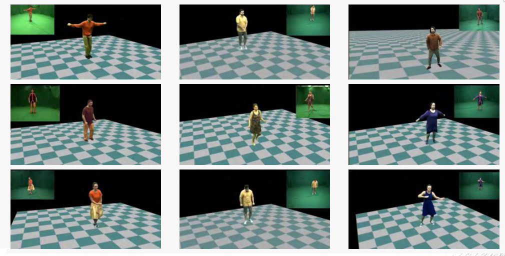
More Results



max planck institut informatik

Christian Theobalt

More Results



max planck institut informatik

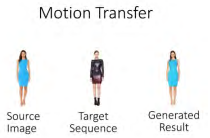
Christian Theobalt

76

5/26/2021

Thank you

- New Visual Computing and AI Department
  - https://www.mpi-inf.mpg.de/departments/visual-computing-and-artificial-intelligence
- Former group website – Graphics, Vision and Video
  - gvv.mpi-inf.mpg.de



Sarkar et al., ECCV 2020

Yoon et al., CVPR 2021

max planck institut informatik

Christian Theobalt

References

- Patrick Esser, Johannes Haux, Timo Milbich, Björn Ommer (2018). Towards Learning a Realistic Rendering of Human Behavior. CVPR.
- Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, Francesc Moreno-Noguer (2018). Unsupervised Person Image Synthesis in Arbitrary Poses. CVPR.
- Chenyang Si, Wei Wang, Liang Wang, Tieniu Tan (2018). Multistage Adversarial Losses for Pose-Based Human Image Synthesis. CVPR.
- Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros (2019). Everybody Dance Now. ICCV.
- Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, Daniel Cohen-Or (2019). Deep Video-Based Performance Cloning. EG.
- A. Siarohin, S. Lathuiliere, S. Tulyakov, E. Ricci, N. Sebe (2019), First Order Motion Model for Image Animation, NeurIPS.
- Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, Mario Fritz (2018), Disentangled Person Image Generation, CVPR.
- Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, Sean Fanello (2018). LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering. ToG.
- Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, Bryan Catanzaro (2018). Video-to-Video Synthesis. NeurIPS.
- Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, Christian Theobalt (2019). Neural Rendering and Reenactment of Human Actor Videos. ToG.
- Yining Li, Chen Huang, Chen Change Loy (2019). Dense intrinsic appearance flow for human pose transfer. CVPR.
- Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, Christian Theobalt (2020). Neural Re-Rendering of Humans from a Single Image. ECCV.
- Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhoefer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, Christian Theobalt (2020). Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation. TVCG.

max planck institut informatik

Christian Theobalt

5/26/2021

References

- C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, Richard Szeliski (2004). High-quality video view interpolation using a layered representation. ToG.
- Carsten Stoll, Jürgen Gall, Edilson de Aguiar, Sebastian Thrun, Christian Theobalt (2010). Video-Based Reconstruction of Animatable Human Characters. ToG.
- Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, Christian Theobalt (2011). Video-based Characters: Creating New Human Performances from a Multi-view Video Database. SIGGRAPH.
- Guannan Li, Yebin Liu, Qionghai Dai (2014). Free-viewpoint Video Relighting from Multi-view Sequence Under General Illumination. MVA.
- Dan Casas, Marco Volino, John Collomosse, Adrian Hilton (2014). 4D Video Textures for Interactive Character Appearance. CGF.
- Marco Volino, Dan Casas, John Collomosse, Adrian Hilton (2014). Optimal Representation of Multiple View Video. BMVC.
- Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, Steve Sullivan (2015). High-quality streamable free-viewpoint video. ToG.

max planck institut informatik

Christian Theobalt

References

- Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, Victor Lempitsky (2019). Textured Neural Avatars. CVPR.
- Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, Yaser Sheikh (2019). Neural volumes: Learning dynamic renderable volumes from images. ToG.
- Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Niessner, Gordon Wetzstein, Michael Zollhöfer (2019). DeepVoxels: Learning Persistent 3D Feature Embeddings. CVPR.
- Abhimitra Meka, Rohit Pandey, Christian Haene, Sergio Orts-Escolano, Peter Barnum, Philip Davidson, Daniel Erickson, Yinda Zhang, Jonathan Taylor, Sofien Bouaziz, Chloe Legendre, Wan-Chun Ma, Ryan Overbeck, Thabo Beeler, Paul Debevec, Shahram Izadi, Christian Theobalt, Christoph Rhemann, Sean Fanello (2020). Deep Relightable Textures. SigAsia.
- Albert Pumarola, Enric Corona, Gerard Pons-Moll, Francesc Moreno-Noguer (2020). D-NeRF: Neural Radiance Fields for Dynamic Scenes. CVPR.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV.
- Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, Christian Theobalt (2021). Real-time Deep Dynamic Characters. SIGGRAPH.

max planck institut informatik

Christian Theobalt

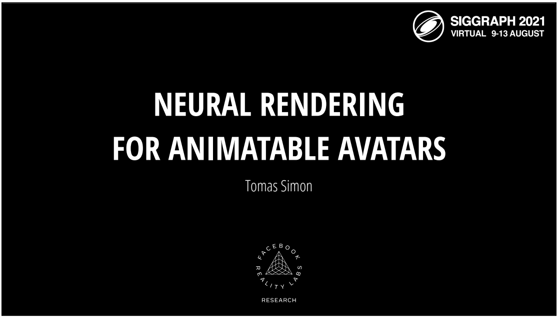
5/26/2021

References

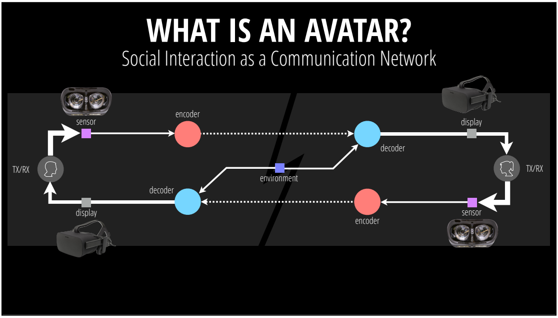
- J. S. Yoon, L. Liu, V. Golyanik, K. Sarkar, H. S. Park and C. Theobalt, Pose-Guided Human Animation from a Single Image in the Wild, Computer Vision and Pattern Recognition (CVPR), 2021
- K. Sarkar, V. Golyanik, L. Liu and C. Theobalt, Style and Pose Control for Image Synthesis of Humans from a Single Monocular View, arXiv.org, 2021.
- K. Sarkar, L. Liu, V. Golyanik, and C. Theobalt, HumanGAN: A Generative Model of Humans Images, arXiv.org, 2021

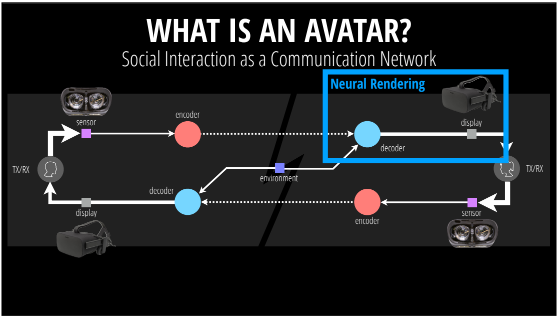
max planck institut informatik

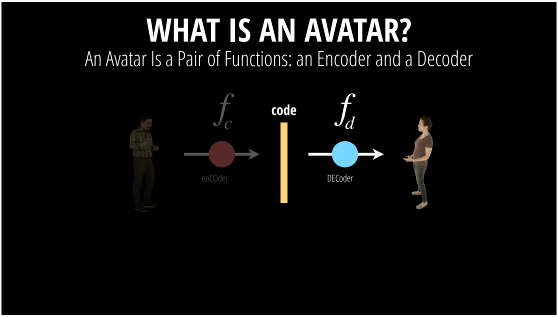
Christian Theobalt

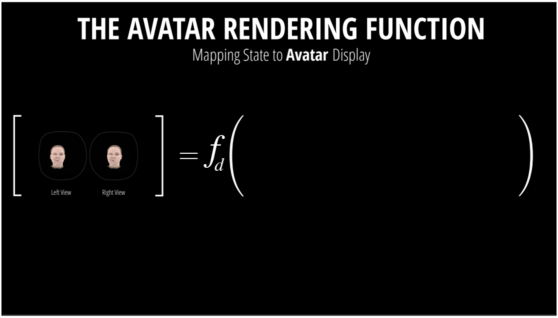

SIGGRAPH 2021
VIRTUAL 9-13 AUGUST
NEURAL RENDERING
FOR ANIMATABLE AVATARS
Tomas Simon
FACEBOOK REALITY LABS
RESEARCH

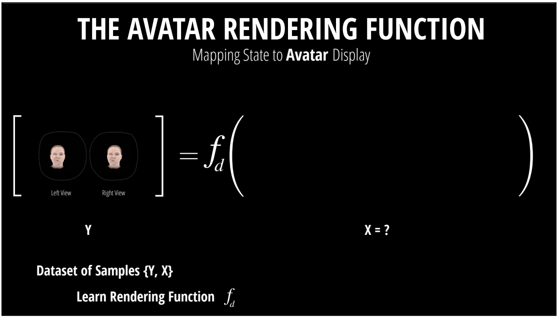

[Deep Appearance Models for Face Rendering, Lombardi et al., 2018]
[VR Facial Animation Via Multiview Image Translation, Wei et al., 2019]

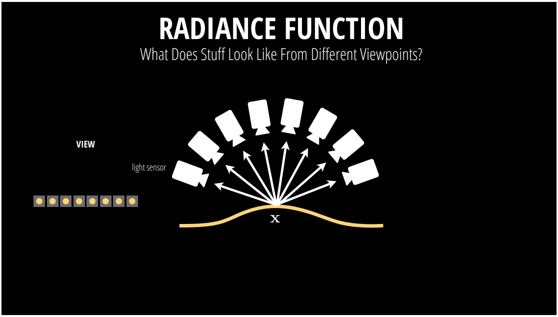

WHAT IS AN AVATAR?
Social Interaction as a Communication Network
sensor
encoder
display
decoder
TX/RX
TX/RX
environment
decoder
display
sensor
encoder

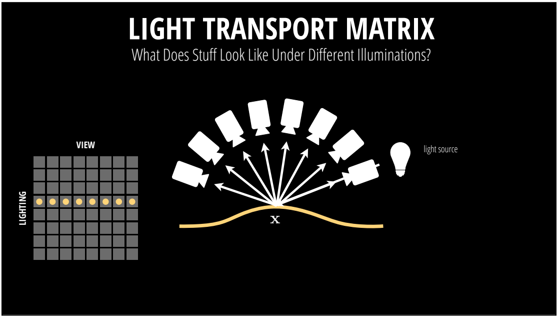

WHAT IS AN AVATAR?
Social Interaction as a Communication Network
Neural Rendering
sensor
encoder
decoder
display
TX/RX
environment
decoder
display
encoder
sensor
TX/RX

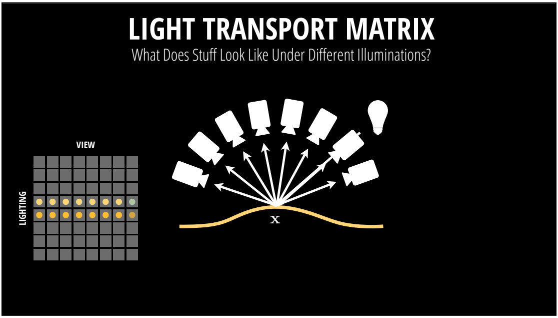

WHAT IS AN AVATAR?
An Avatar Is a Pair of Functions: an Encoder and a Decoder
f_c
code
f_d
enCOder
DECoder

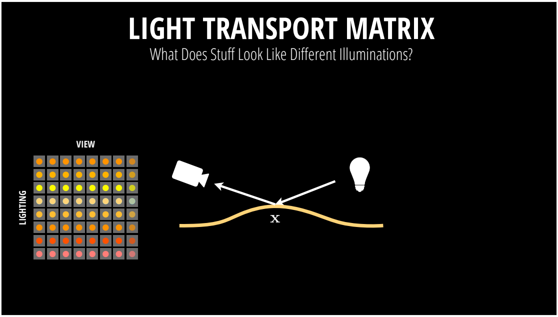

THE AVATAR RENDERING FUNCTION
Mapping State to **Avatar** Display
Left View
Right View
$= f_d($ $)$

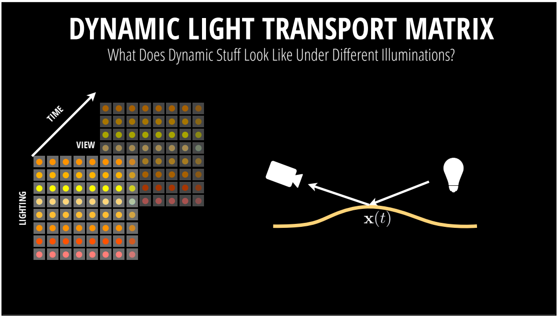

THE AVATAR RENDERING FUNCTION
Mapping State to **Avatar** Display
Left View
Right View
$= f_d$
Y
X = ?
Dataset of Samples {Y, X}
Learn Rendering Function f_d

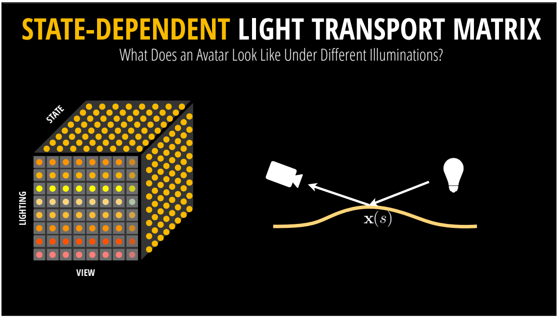

RADIANCE FUNCTION
What Does Stuff Look Like From Different Viewpoints?
VIEW
light sensor
x

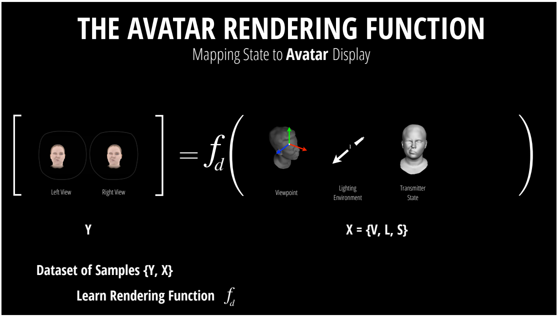

LIGHT TRANSPORT MATRIX
What Does Stuff Look Like Under Different Illuminations?
VIEW
LIGHTING
light source
x

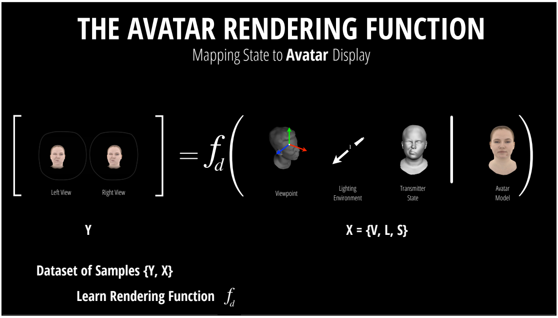

LIGHT TRANSPORT MATRIX
What Does Stuff Look Like Under Different Illuminations?
VIEW
LIGHTING
x

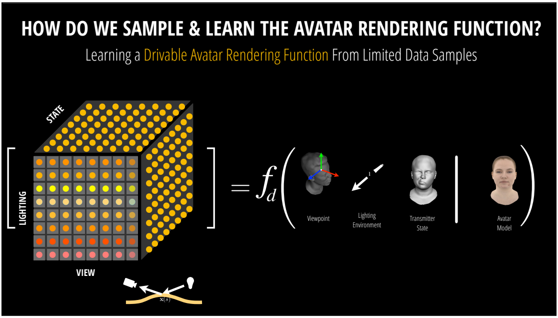

LIGHT TRANSPORT MATRIX
What Does Stuff Look Like Different Illuminations?
VIEW
LIGHTING
x

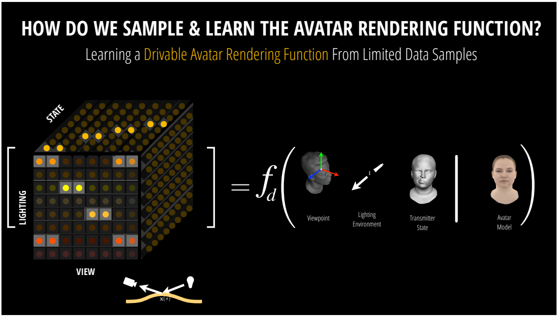

DYNAMIC LIGHT TRANSPORT MATRIX
What Does Dynamic Stuff Look Like Under Different Illuminations?
TIME
VIEW
LIGHTING
x(t)

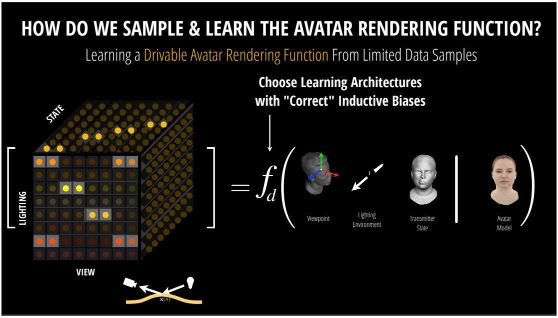

STATE-DEPENDENT LIGHT TRANSPORT MATRIX
What Does an Avatar Look Like Under Different Illuminations?
STATE
LIGHTING
VIEW
$\mathbf{x}(s)$

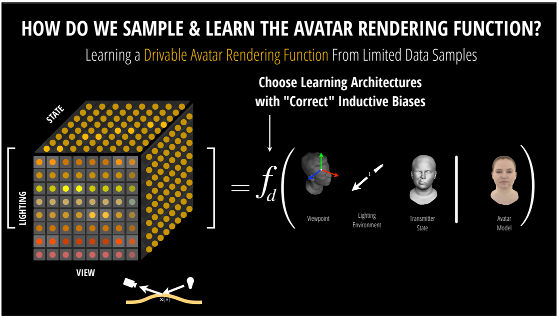

THE AVATAR RENDERING FUNCTION
Mapping State to **Avatar** Display
Left View
Right View
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Y
X = {V, L, S}
Dataset of Samples {Y, X}
Learn Rendering Function f_d

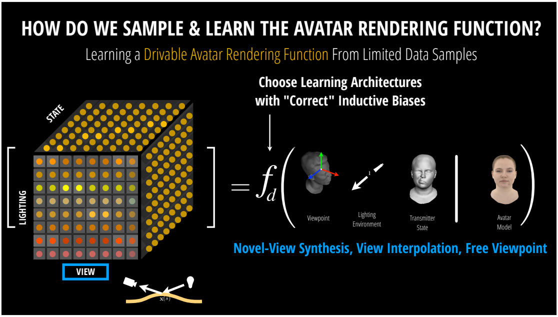

THE AVATAR RENDERING FUNCTION
Mapping State to **Avatar** Display
Left View
Right View
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Y
X = {V, L, S}
Dataset of Samples {Y, X}
Learn Rendering Function f_d

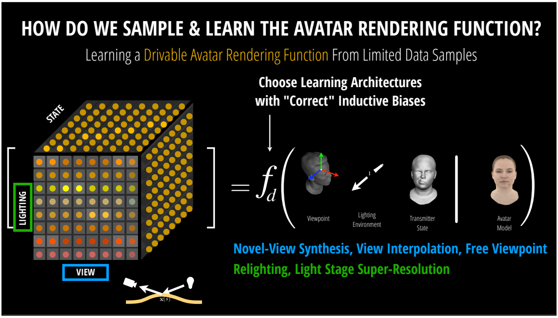

HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
x(s)

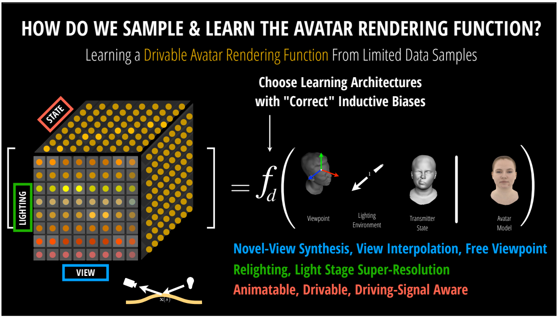

HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
x(s)


HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Learning Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
x(s)


HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Learning Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model

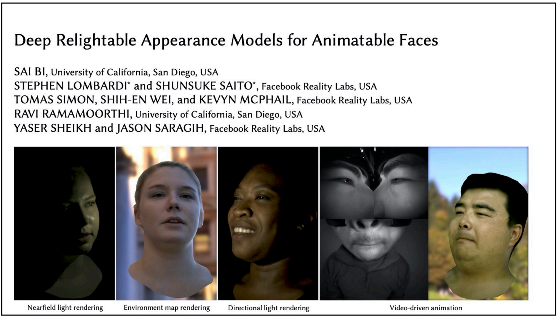

HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Learning Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Novel-View Synthesis, View Interpolation, Free Viewpoint


HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Learning Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Novel-View Synthesis, View Interpolation, Free Viewpoint
Relighting, Light Stage Super-Resolution


HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Learning Architectures with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Novel-View Synthesis, View Interpolation, Free Viewpoint
Relighting, Light Stage Super-Resolution
Animatable, Drivable, Driving-Signal Aware

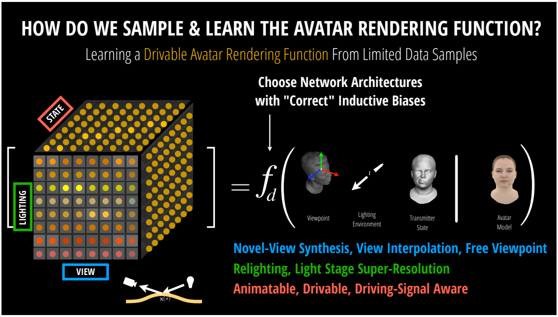

1) DEEP RELIGHTABLE APPEARANCE MODELS
RELIT ON
RELIT OFF
[Deep Relightable Appearance Models for Animatable Faces, Saito et al. SIGGRAPH 2021]

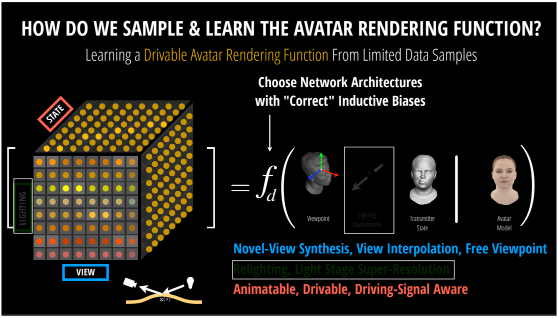

2) DRIVING-SIGNAL AWARE FULL BODY AVATARS
Driving-Aware ON
Driving-Aware OFF
[Driving-Signal Aware Full Body Avatars, Bagautdinov et al. SIGGRAPH 2021]


Deep Relightable Appearance Models for Animatable Faces

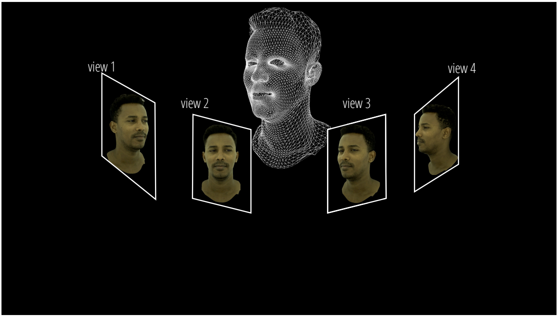
SAI BI, University of California, San Diego, USA
STEPHEN LOMBARDI* and SHUNSUKE SAITO*, Facebook Reality Labs, USA
TOMAS SIMON, SHIH-EN WEI, and KEVYN MCPHAIL, Facebook Reality Labs, USA
RAVI RAMAMOORTHI, University of California, San Diego, USA
YASER SHEIKH and JASON SARAGIH, Facebook Reality Labs, USA

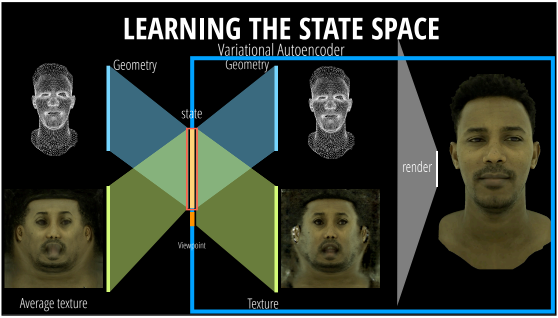
Nearfield light rendering | Environment map rendering | Directional light rendering | Video-driven animation

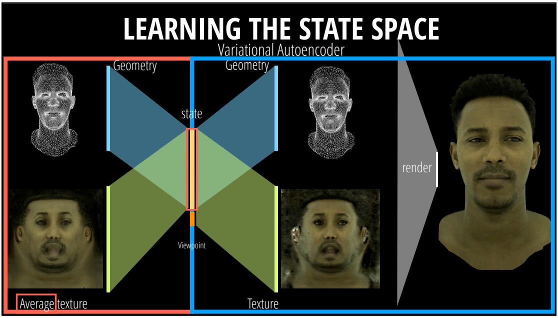


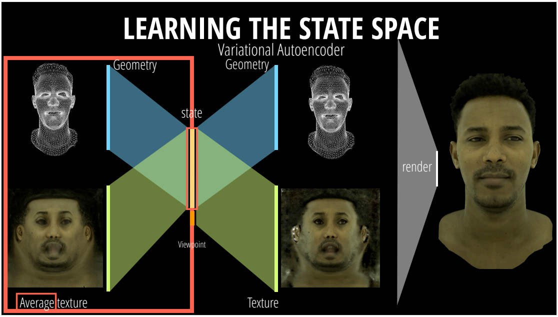


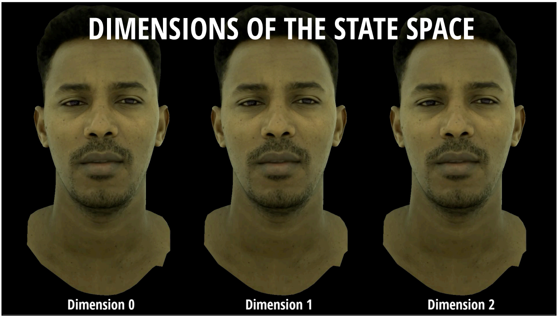

FACE CAPTURE SYSTEM
Multiview Capture System
~160 Cameras, ~450 Lights
[Deep Appearance Models for Face Rendering, Lombardi et al., 2018]

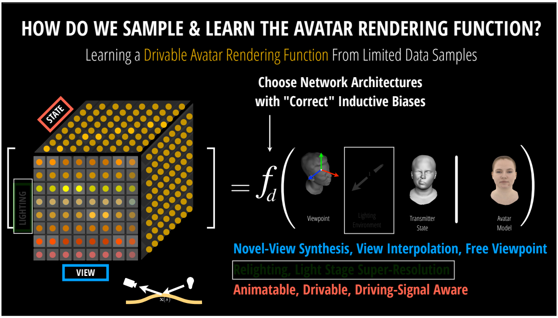

view 1
view 2
view 3
view 4

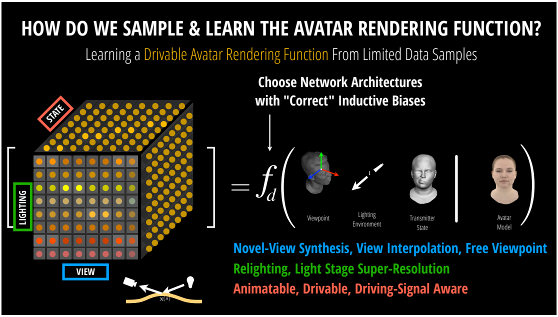

LEARNING THE STATE SPACE
Variational Autoencoder
Geometry
Geometry
state
Viewpoint
render
Average texture
Texture

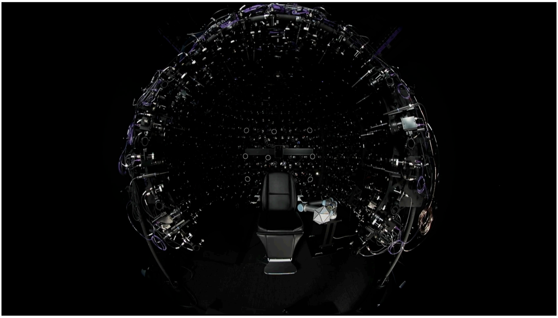

LEARNING THE STATE SPACE
Variational Autoencoder
Geometry
Geometry
state
Viewpoint
render
Average texture
Texture

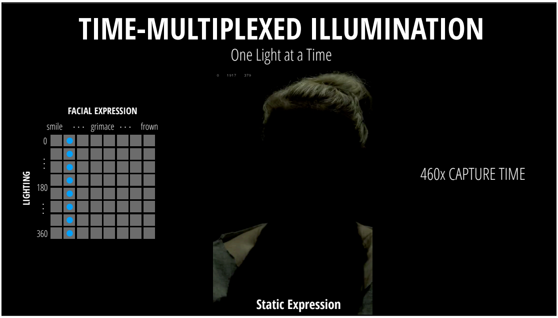

LEARNING THE STATE SPACE
Variational Autoencoder
Geometry
Geometry
state
Viewpoint
render
Average texture
Texture

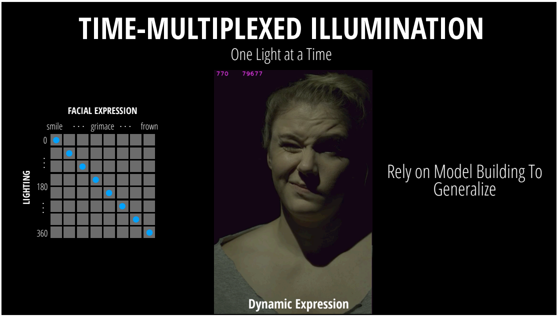

DIMENSIONS OF THE STATE SPACE
Dimension 0
Dimension 1
Dimension 2


HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Network Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Novel-View Synthesis, View Interpolation, Free Viewpoint
Relighting, Light Stage Super-Resolution
Animatable, Drivable, Driving-Signal Aware

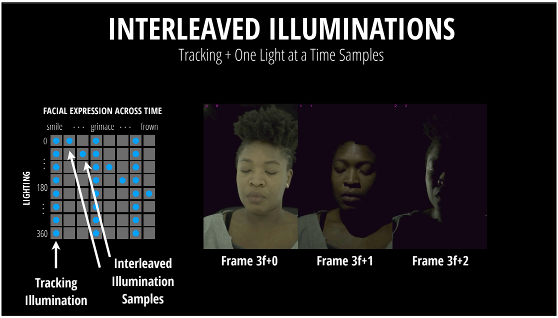

HOW DO WE SAMPLE & LEARN THE AVATAR RENDERING FUNCTION?
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Choose Network Architectures
with "Correct" Inductive Biases
STATE
LIGHTING
VIEW
$= f_d$
Viewpoint
Lighting Environment
Transmitter State
Avatar Model
Novel-View Synthesis, View Interpolation, Free Viewpoint
Relighting, Light Stage Super-Resolution
Animatable, Drivable, Driving-Signal Aware

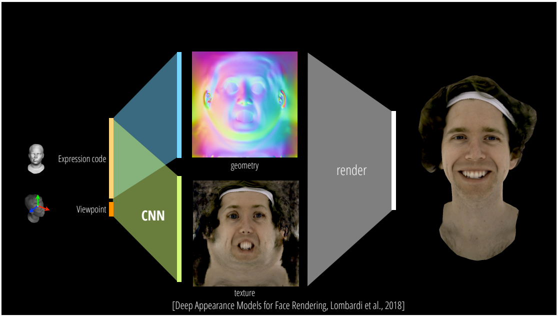


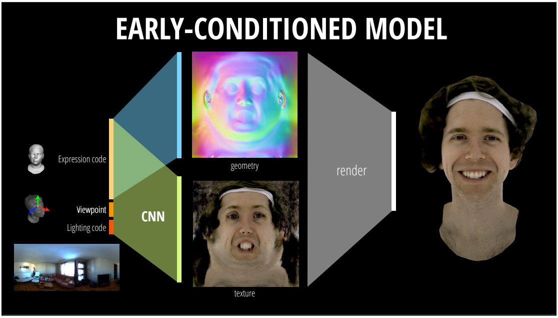

TIME-MULTIPLEXED ILLUMINATION
One Light at a Time
FACIAL EXPRESSION
smile · · · grimace · · · frown
LIGHTING
0
180
360
460x CAPTURE TIME
Static Expression

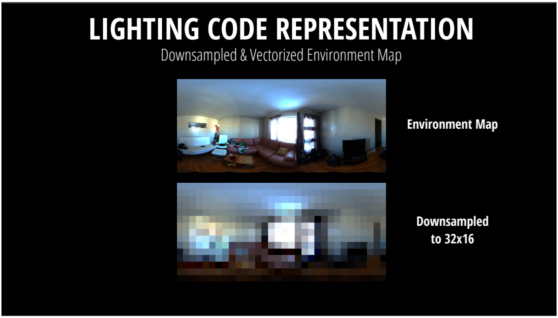

TIME-MULTIPLEXED ILLUMINATION
One Light at a Time
770 79677
FACIAL EXPRESSION
smile · · · grimace · · · frown
LIGHTING
0
180
360
Rely on Model Building To Generalize
Dynamic Expression

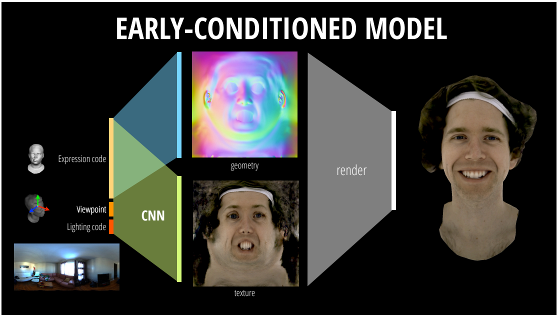

INTERLEAVED ILLUMINATIONS

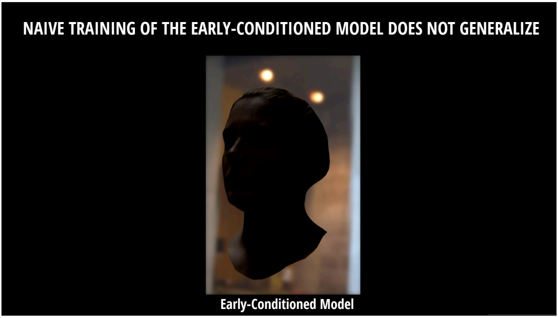

INTERLEAVED ILLUMINATIONS
Tracking + One Light at a Time Samples
FACIAL EXPRESSION ACROSS TIME
smile
grimace
frown
LIGHTING
0
180
360
Tracking Illumination
Interleaved Illumination Samples
Frame 3f+0
Frame 3f+1
Frame 3f+2

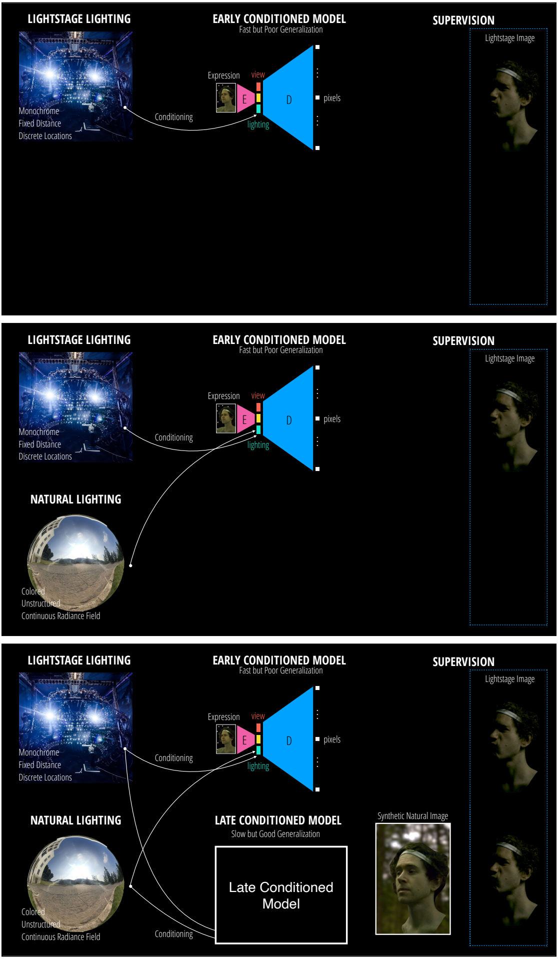

Expression code
Viewpoint
CNN
geometry
texture
render
[Deep Appearance Models for Face Rendering, Lombardi et al., 2018]

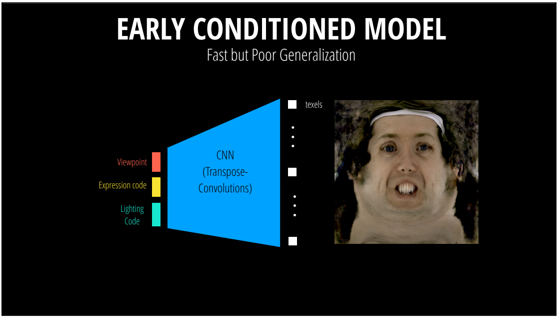

EARLY-CONDITIONED MODEL
Expression code
Viewpoint
Lighting code
CNN
geometry
texture
render

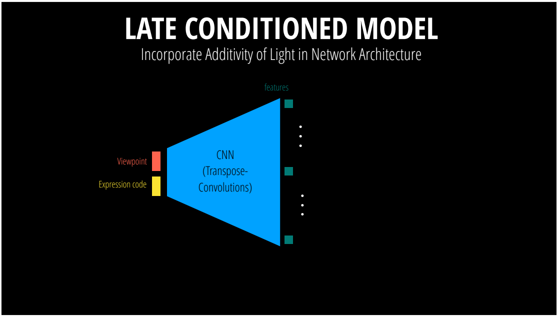

LIGHTING CODE REPRESENTATION
Downsampled & Vectorized Environment Map
Environment Map
Downsampled
to 32x16

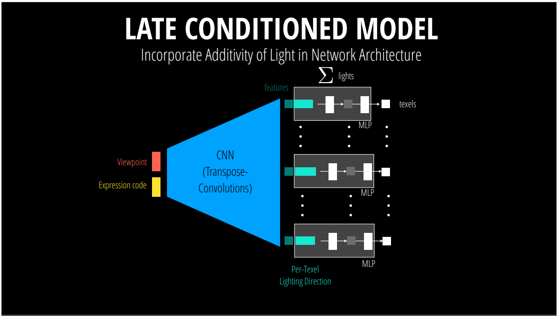

EARLY-CONDITIONED MODEL
Expression code
Viewpoint
Lighting code
CNN
geometry
texture
render

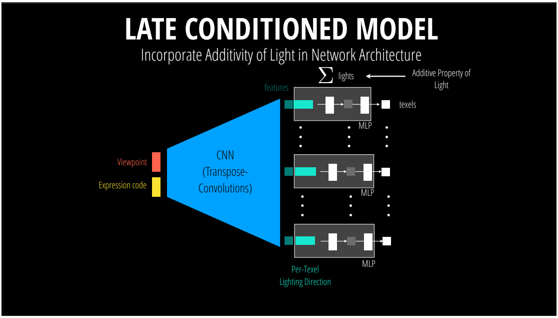

NAIVE TRAINING OF THE EARLY-CONDITIONED MODEL DOES NOT GENERALIZE
Early-Conditioned Model

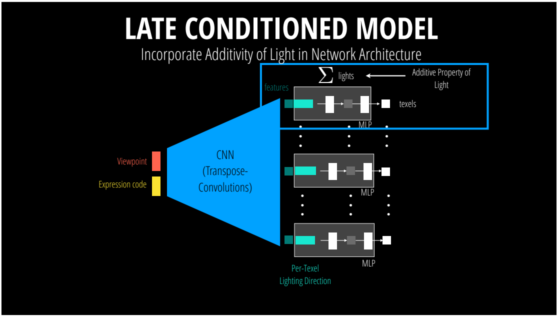

LIGHTSTAGE LIGHTING
Monochrome
Fixed Distance
Discrete Locations
EARLY CONDITIONED MODEL
Fast but Poor Generalization
Expression
view
E
D
pixels
lighting
Conditioning
SUPERVISION
Lightstage Image
NATURAL LIGHTING
Colored
Unstructured
Continuous Radiance Field
LATE CONDITIONED MODEL
Slow but Good Generalization
Late Conditioned Model
Synthetic Natural Image

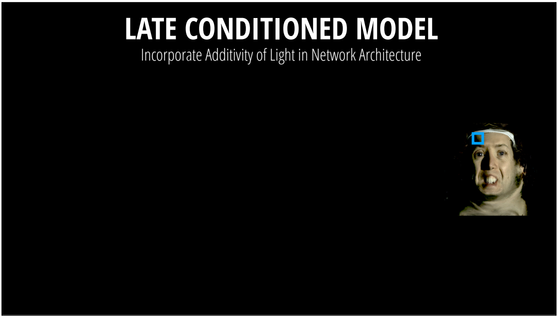

EARLY CONDITIONED MODEL
Fast but Poor Generalization
Viewpoint
Expression code
Lighting Code
CNN (Transpose-Convolutions)
texels

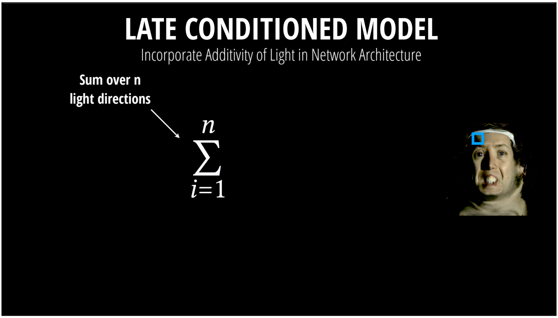

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
features
Viewpoint
Expression code
CNN (Transpose-Convolutions)

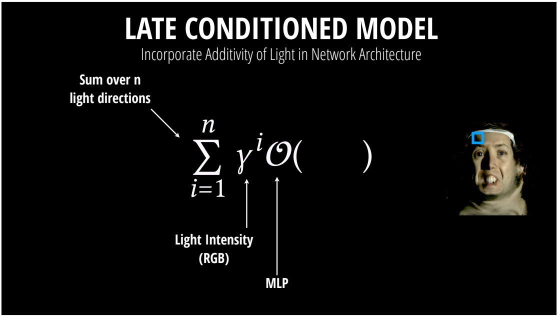

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
∑ lights
features
texels
MLP
Viewpoint
Expression code
CNN (Transpose-Convolutions)
MLP
MLP
Per-Texel Lighting Direction

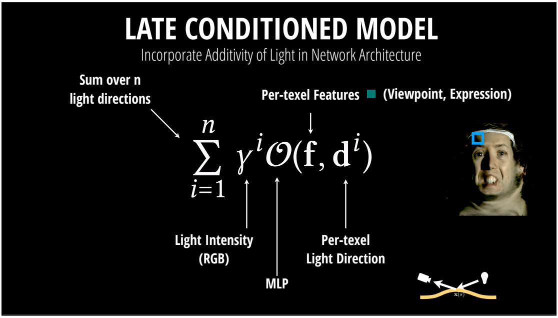

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
∑ lights
Additive Property of Light
features
texels
MLP
MLP
MLP
Viewpoint
Expression code
CNN (Transpose-Convolutions)
Per-Texel Lighting Direction

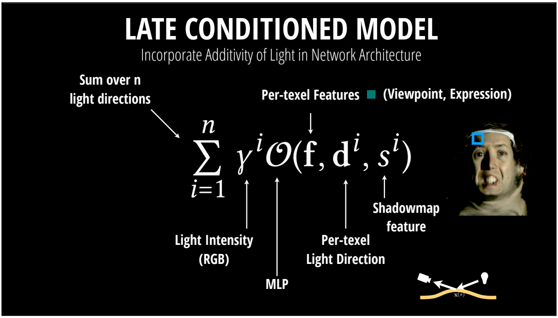

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
∑ lights
Additive Property of Light
features
texels
MLP
MLP
MLP
Viewpoint
Expression code
CNN (Transpose-Convolutions)
Per-Texel Lighting Direction

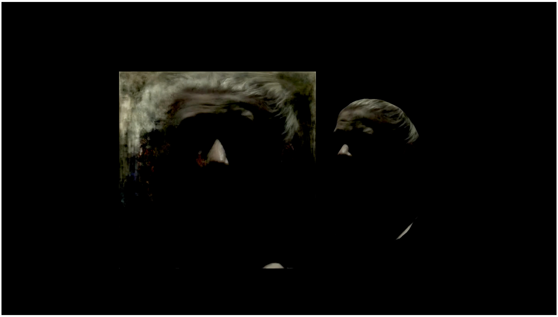

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture

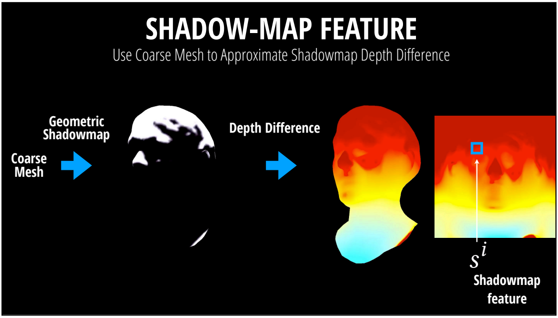

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
Sum over n
light directions
$\sum_{i=1}^{n}$

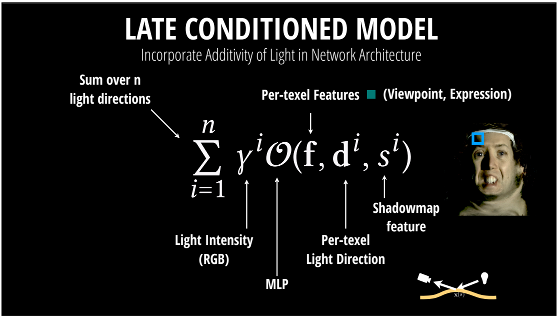

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
Sum over n
light directions
$\sum_{i=1}^{n} \gamma^{i} \mathcal{O}(\quad)$
Light Intensity
(RGB)
MLP

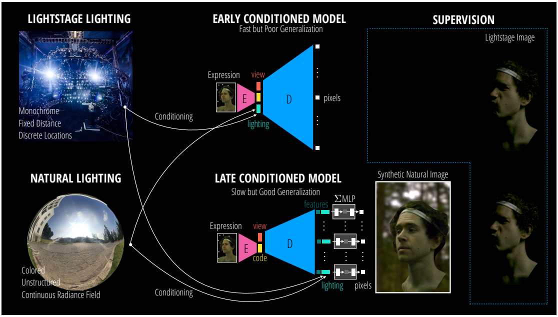

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
Sum over n
light directions
Per-texel Features
(Viewpoint, Expression)
$\sum_{i=1}^{n} \gamma^{i} \mathcal{O}(\mathrm{f}, \mathrm{d}^{i})$
Light Intensity
(RGB)
MLP
Per-texel
Light Direction

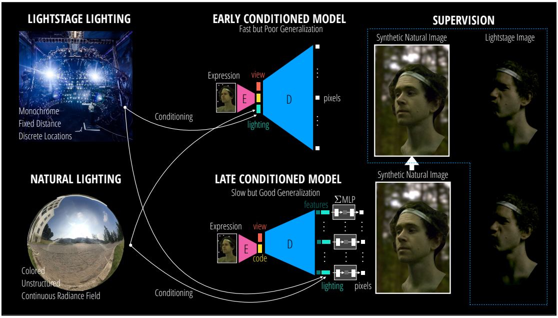

LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
Sum over n light directions
Per-texel Features (Viewpoint, Expression)
$\sum_{i=1}^{n} \gamma^i \mathcal{O}(\mathbf{f}, \mathbf{d}^i, s^i)$
Light Intensity (RGB)
MLP
Per-texel Light Direction
Shadowmap feature

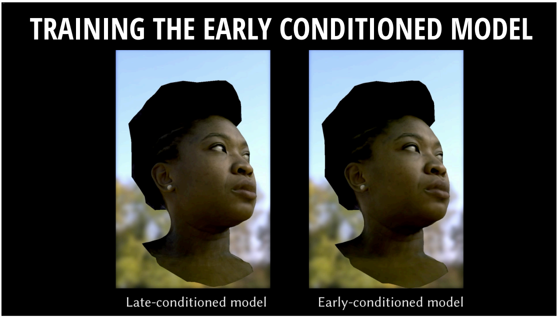


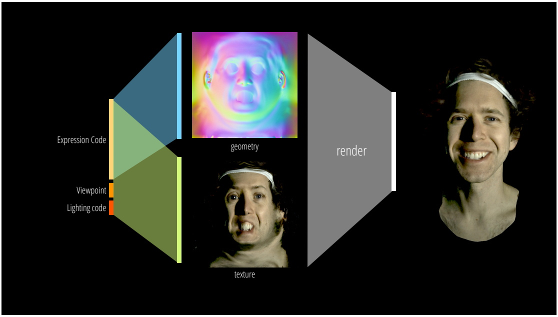

SHADOW-MAP FEATURE
Use Coarse Mesh to Approximate Shadowmap Depth Difference
Geometric Shadowmap
Coarse Mesh
Depth Difference
s^i
Shadowmap feature


LATE CONDITIONED MODEL
Incorporate Additivity of Light in Network Architecture
Sum over n light directions
Per-texel Features
(Viewpoint, Expression)
$\sum_{i=1}^{n} \gamma^i \mathcal{O}(\mathrm{f}, \mathrm{d}^i, s^i)$
Light Intensity (RGB)
MLP
Per-texel Light Direction
Shadowmap feature

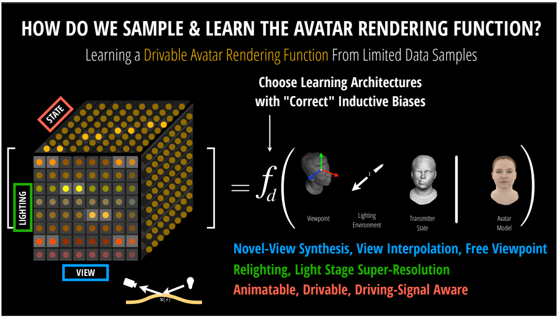

LIGHTSTAGE LIGHTING
Monochrome
Fixed Distance
Discrete Locations
NATURAL LIGHTING
Colored
Unstructured
Continuous Radiance Field
EARLY CONDITIONED MODEL
Fast but Poor Generalization
Expression
view
E
D
pixels
lighting
Conditioning
LATE CONDITIONED MODEL
Slow but Good Generalization
ΣMLP
features
code
lighting
pixels
Conditioning
SUPERVISION
Lightstage Image
Synthetic Natural Image

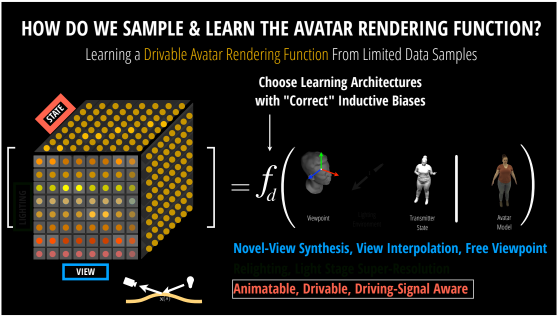

LIGHTSTAGE LIGHTING
Monochrome
Fixed Distance
Discrete Locations
NATURAL LIGHTING
Colored
Unstructured
Continuous Radiance Field
EARLY CONDITIONED MODEL
Fast but Poor Generalization
Expression
view
E
D
pixels
lighting
Conditioning
LATE CONDITIONED MODEL
Slow but Good Generalization
ΣMLP
features
code
lighting
pixels
Conditioning
SUPERVISION
Synthetic Natural Image
Lightstage Image
Synthetic Natural Image

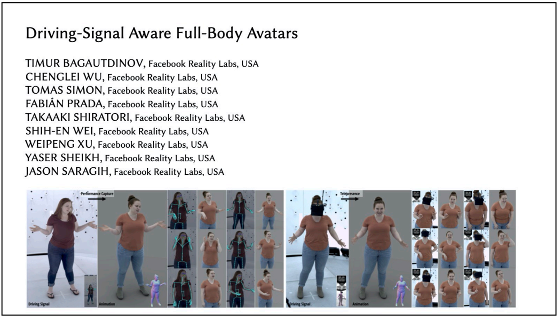

TRAINING THE EARLY CONDITIONED MODEL
Late-conditioned model
Early-conditioned model


Expression Code
Viewpoint
Lighting code
geometry
texture
render

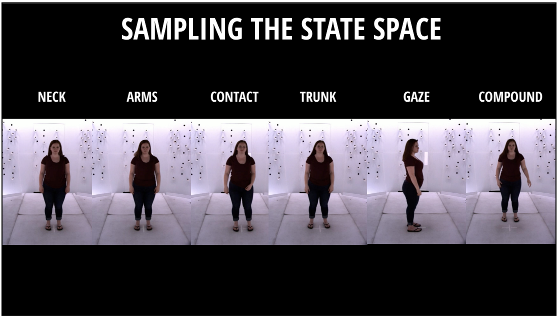

DRIVING THE RELIGHTABLE MODEL



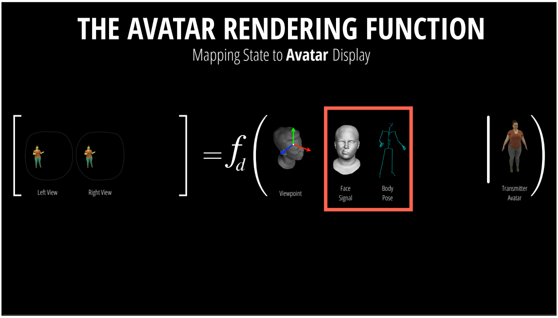


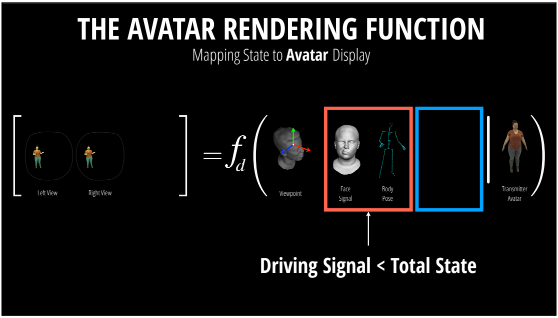

Driving-Signal Aware Full-Body Avatars

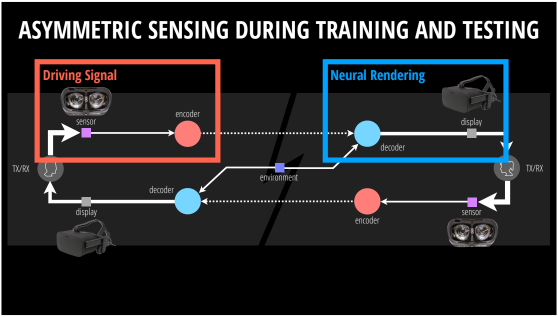
TIMUR BAGAUTDINOV, Facebook Reality Labs, USA
CHENGLEI WU, Facebook Reality Labs, USA
TOMAS SIMON, Facebook Reality Labs, USA
FABIÁN PRADA, Facebook Reality Labs, USA
TAKAAKI SHIRATORI, Facebook Reality Labs, USA
SHIH-EN WEI, Facebook Reality Labs, USA
WEIPENG XU, Facebook Reality Labs, USA
YASER SHEIKH, Facebook Reality Labs, USA
JASON SARAGIH, Facebook Reality Labs, USA

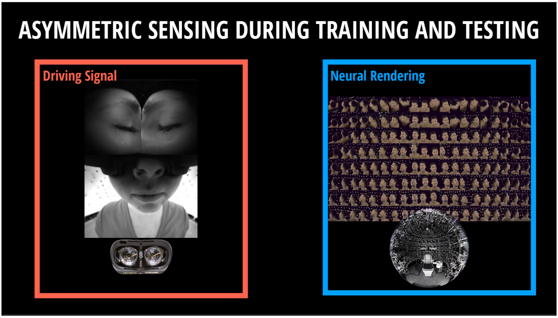

SAMPLING THE STATE SPACE
NECK
ARMS
CONTACT
TRUNK
GAZE
COMPOUND

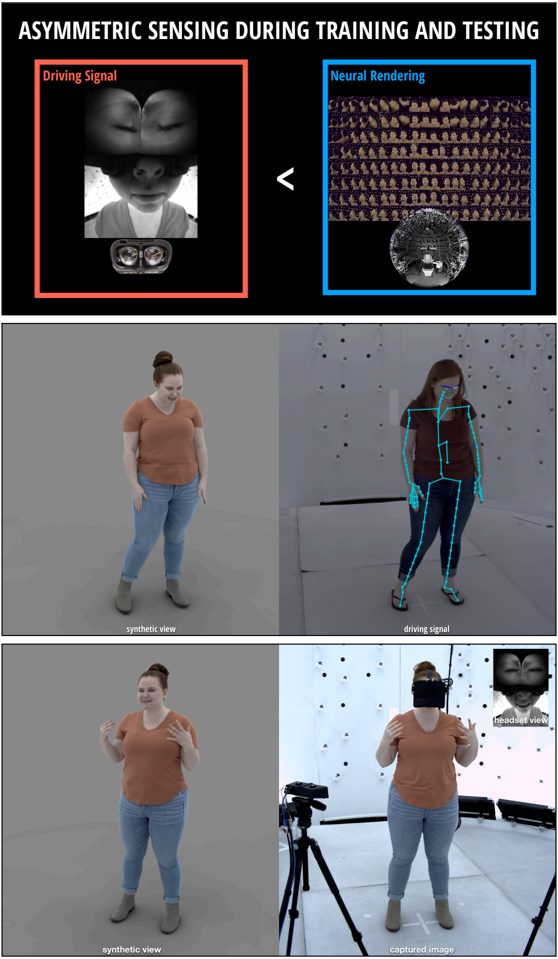

synthetic view
driving signal

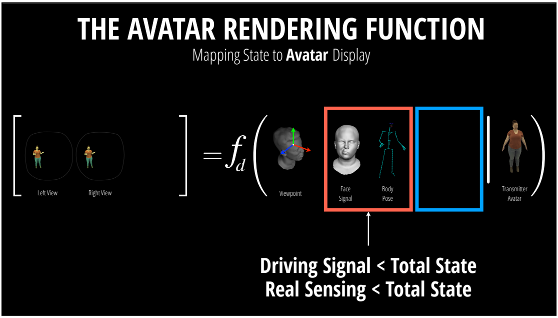

THE AVATAR RENDERING FUNCTION
Mapping State to **Avatar** Display
$= f_d$
Left View
Right View
Viewpoint
Face Signal
Body Pose
Transmitter Avatar

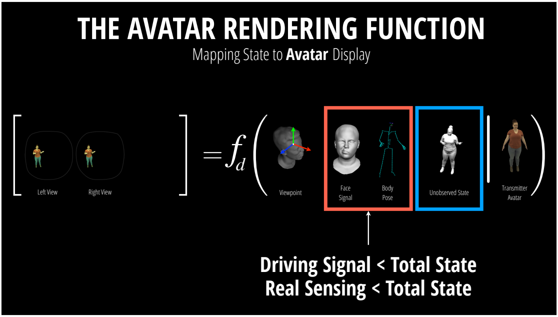

THE AVATAR RENDERING FUNCTION
Mapping State to Avatar Display
Left View
Right View
$= f_d$
Viewpoint
Face Signal
Body Pose
Transmitter Avatar
Driving Signal < Total State

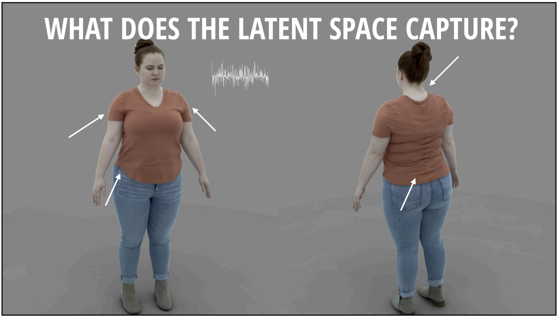

ASYMMETRIC SENSING DURING TRAINING AND TESTING
Driving Signal
Neural Rendering
sensor
encoder
display
decoder
TX/RX
environment
decoder
display
encoder
sensor
TX/RX


ASYMMETRIC SENSING DURING TRAINING AND TESTING
Driving Signal
Neural Rendering

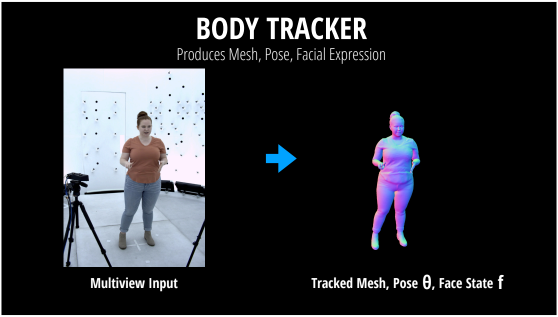

ASYMMETRIC SENSING DURING TRAINING AND TESTING
Driving Signal
<
Neural Rendering
synthetic view
driving signal
headset view
synthetic view
captured image

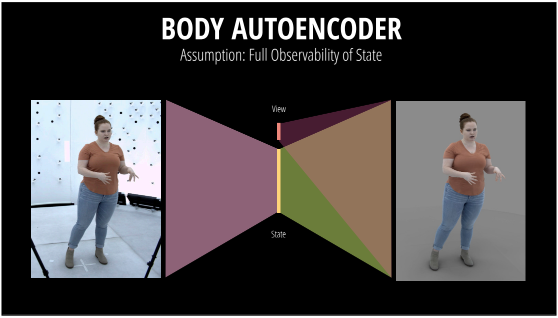

THE AVATAR RENDERING FUNCTION
Mapping State to Avatar Display
Left View
Right View
Viewpoint
Face Signal
Body Pose
Transmitter Avatar
Driving Signal < Total State
Real Sensing < Total State

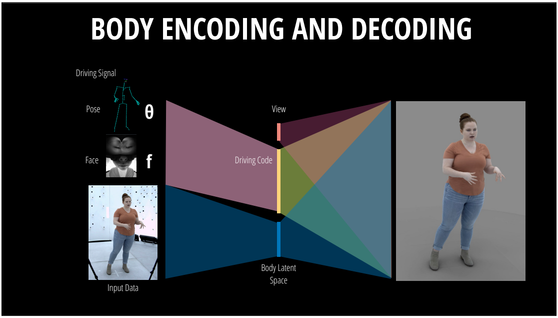

THE AVATAR RENDERING FUNCTION
Mapping State to Avatar Display
Left View
Right View
Viewpoint
Face Signal
Body Pose
Unobserved State
Transmitter Avatar
Driving Signal < Total State
Real Sensing < Total State

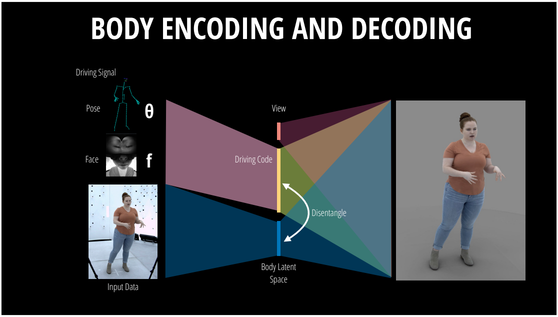

WHAT DOES THE LATENT SPACE CAPTURE?

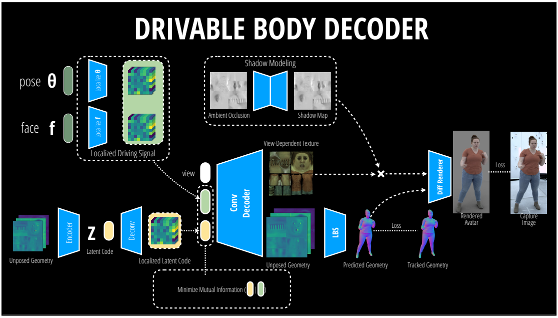

BODY CAPTURE SYSTEM
Multiview Capture System
Body Capture
~180 Cams

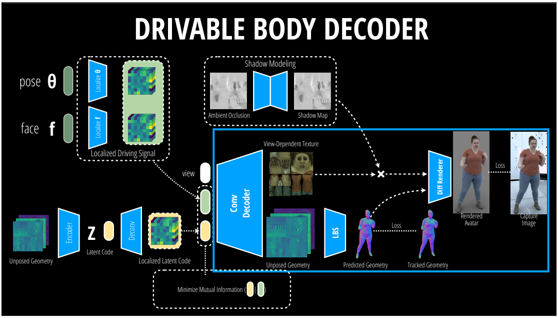

BODY TRACKER
Produces Mesh, Pose, Facial Expression
Multiview Input
Tracked Mesh, Pose θ, Face State f

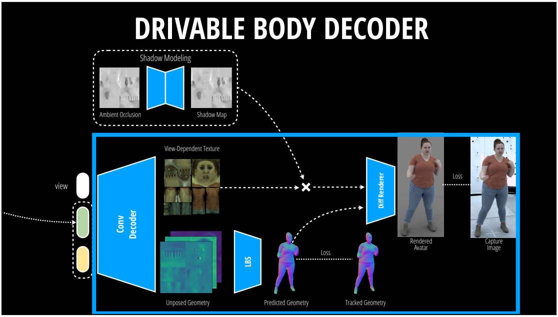

BODY AUTOENCODER
Assumption: Full Observability of State
View
State

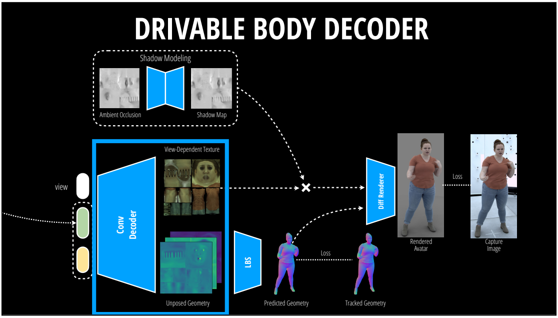

BODY ENCODING AND DECODING
Driving Signal
Pose
θ
Face
f
Input Data
View
Driving Code
Body Latent
Space

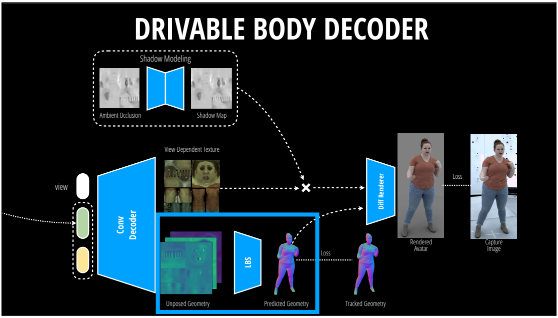

BODY ENCODING AND DECODING
Driving Signal
Pose
θ
Face
f
Input Data
View
Driving Code
Disentangle
Body Latent
Space

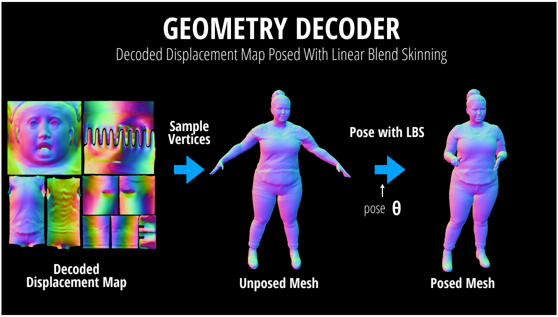

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv
Decoder
Diff Renderer
Rendered
Avatar
Loss
Capture
Image
Unposed Geometry
Encoder
z
Latent Code
Deconv
Localized Latent Code
Unposed Geometry
LBS
Predicted Geometry
Loss
Tracked Geometry
Minimize Mutual Information

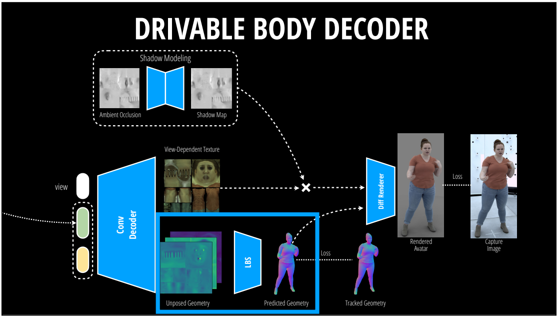

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
Unposed Geometry
Encoder
z
Latent Code
Deconv
Localized Latent Code
LBS
Predicted Geometry
Tracked Geometry
Minimize Mutual Information

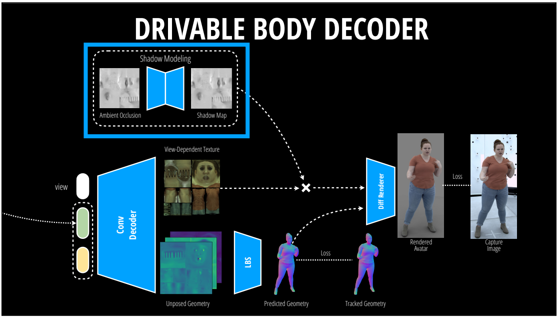

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Loss
Unposed Geometry
Predicted Geometry
Tracked Geometry

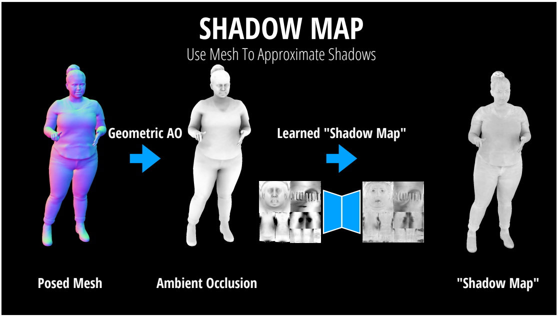

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Loss
Unposed Geometry
Predicted Geometry
Tracked Geometry

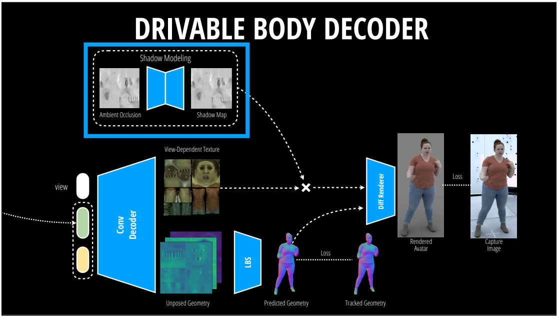

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Loss
Unposed Geometry
Predicted Geometry
Tracked Geometry

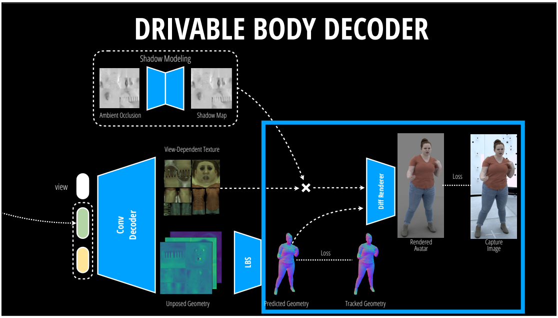

GEOMETRY DECODER
Decoded Displacement Map Posed With Linear Blend Skinning
Sample Vertices
Pose with LBS
pose θ
Decoded Displacement Map
Unposed Mesh
Posed Mesh


DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Loss
Unposed Geometry
Predicted Geometry
Tracked Geometry

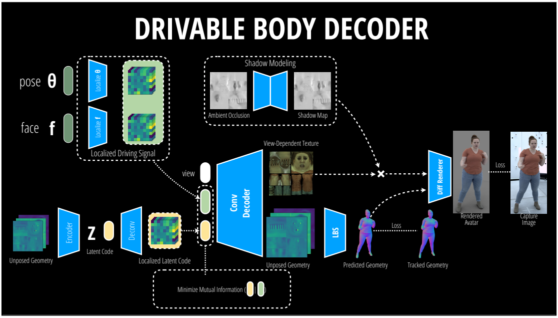

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Unposed Geometry
Predicted Geometry
Loss
Tracked Geometry

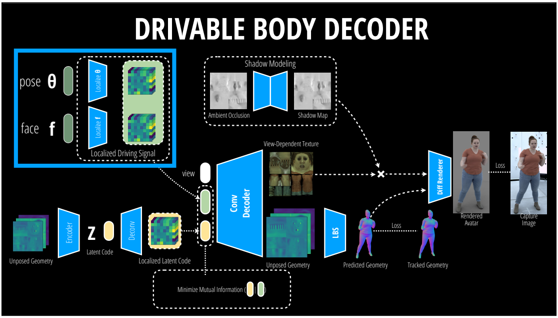

SHADOW MAP
Use Mesh To Approximate Shadows
Geometric AO
Learned "Shadow Map"
Posed Mesh
Ambient Occlusion
"Shadow Map"

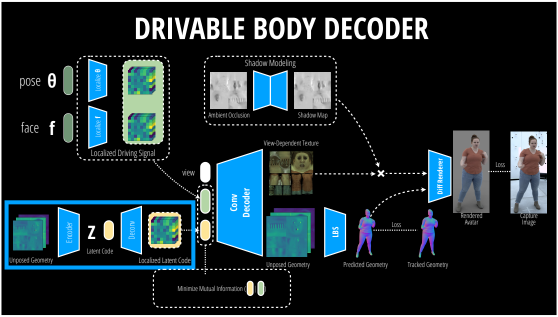

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Unposed Geometry
Predicted Geometry
Loss
Tracked Geometry

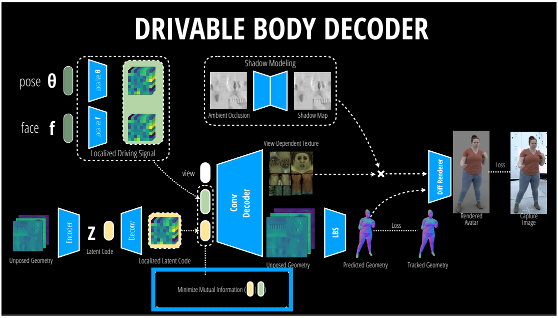

DRIVABLE BODY DECODER
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
LBS
Unposed Geometry
Loss
Predicted Geometry
Tracked Geometry


OURS
No Shadow Branch

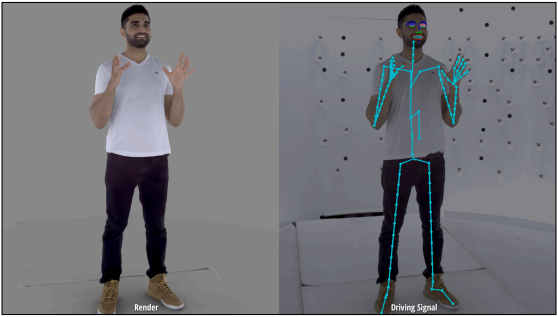

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
Unposed Geometry
Encoder
Z
Latent Code
Deconv
Localized Latent Code
Unposed Geometry
LBS
Loss
Predicted Geometry
Tracked Geometry
Minimize Mutual Information

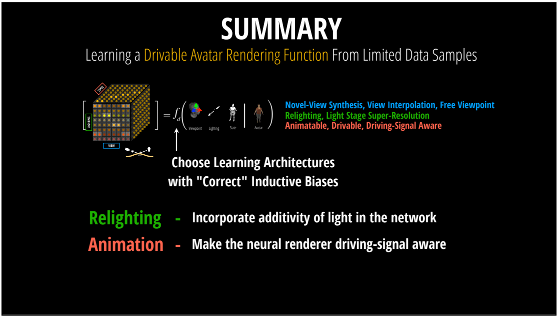

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
Unposed Geometry
Encoder
Z
Latent Code
Deconv
Localized Latent Code
Unposed Geometry
LBS
Predicted Geometry
Loss
Tracked Geometry
Minimize Mutual Information

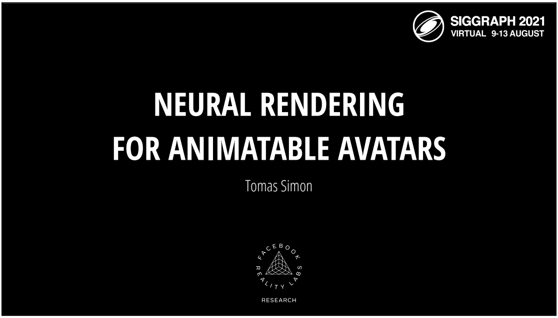

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
Unposed Geometry
Encoder
Z
Latent Code
Deconv
Localized Latent Code
Unposed Geometry
LBS
Predicted Geometry
Loss
Tracked Geometry
Minimize Mutual Information

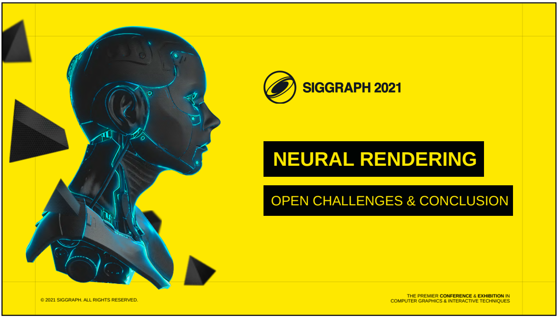

DRIVABLE BODY DECODER
pose θ
face f
Localize θ
Localize f
Localized Driving Signal
Shadow Modeling
Ambient Occlusion
Shadow Map
View-Dependent Texture
view
Conv Decoder
Diff Renderer
Rendered Avatar
Loss
Capture Image
Unposed Geometry
Encoder
Z
Latent Code
Deconv
Localized Latent Code
Unposed Geometry
LBS
Predicted Geometry
Loss
Tracked Geometry
Minimize Mutual Information

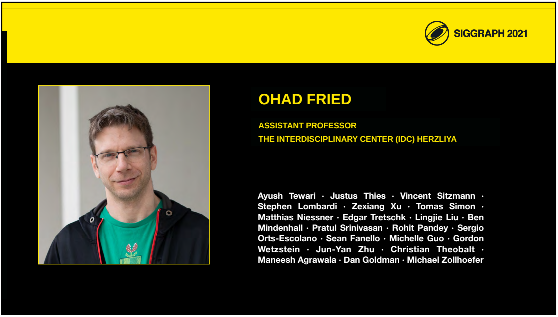

OURS
no latent space
latent space, no disentangling

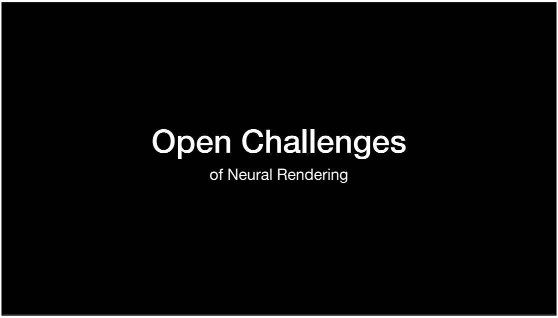

Render
Driving Signal

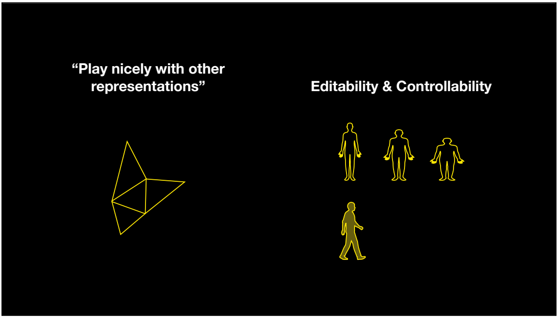

SUMMARY
Learning a Drivable Avatar Rendering Function From Limited Data Samples
Novel-View Synthesis, View Interpolation, Free Viewpoint
Relighting, Light Stage Super-Resolution
Animatable, Drivable, Driving-Signal Aware
Choose Learning Architectures
with "Correct" Inductive Biases
Relighting - Incorporate additivity of light in the network
Animation - Make the neural renderer driving-signal aware

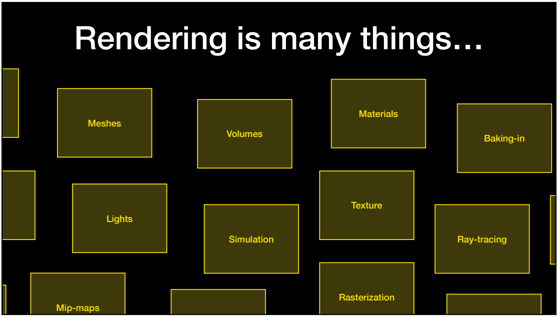

SIGGRAPH 2021
VIRTUAL 9-13 AUGUST
NEURAL RENDERING
FOR ANIMATABLE AVATARS
Tomas Simon
FACEBOOK REALITY LABS
RESEARCH

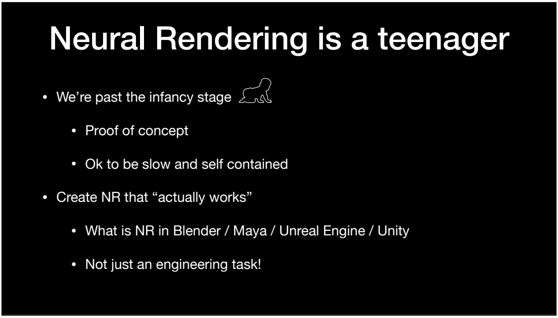

SIGGRAPH 2021
NEURAL RENDERING
OPEN CHALLENGES & CONCLUSION
© 2021 SIGGRAPH. ALL RIGHTS RESERVED.
THE PREMIER CONFERENCE & EXHIBITION IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES


SIGGRAPH 2021
OHAD FRIED
ASSISTANT PROFESSOR
THE INTERDISCIPLINARY CENTER (IDC) HERZLIYA
Ayush Tewari · Justus Thies · Vincent Sitzmann · Stephen Lombardi · Zexiang Xu · Tomas Simon · Matthias Niessner · Edgar Tretschk · Lingjie Liu · Ben Mindenhall · Pratul Srinivasan · Rohit Pandey · Sergio Orts-Escolano · Sean Fanello · Michelle Guo · Gordon Wetzstein · Jun-Yan Zhu · Christian Theobalt · Maneesh Agrawala · Dan Goldman · Michael Zollhoefer


Open Challenges
of Neural Rendering


"Play nicely with other representations"
Editability & Controllability


Rendering is many things…
Meshes
Volumes
Materials
Baking-in
Lights
Simulation
Texture
Ray-tracing
Mip-maps
Rasterization


Neural Rendering is a teenager
We're past the infancy stage
Proof of concept
Ok to be slow and self contained
Create NR that "actually works"
What is NR in Blender / Maya / Unreal Engine / Unity
Not just an engineering task!

Mixed Representations

- Meshes + NR primitives
- What is a NR primitive?
- How to package and share?
- Which representation when?
- How to render together?
- Edit appearance & animate

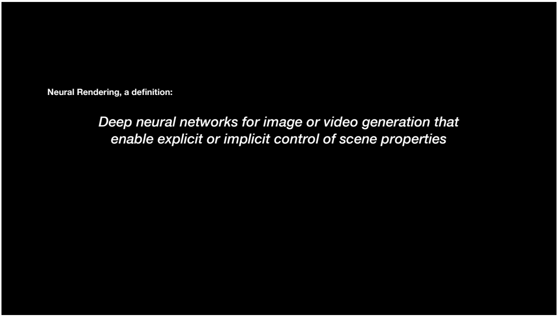
Edit appearance & animate

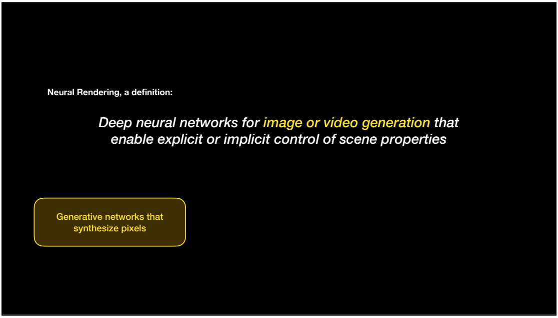
- Exciting progress
  - Edit lighting, camera, material, …
- Still, far from traditional rendering
  - Gain inspiration from traditional workflows and apply to NR

Many other open challenges
mentioned throughout the day…

Alas, the NR course is almost over


Recap


Neural Rendering, a definition:
Deep neural networks for image or video generation that enable explicit or implicit control of scene properties


Neural Rendering, a definition:
Deep neural networks for image or video generation that enable explicit or implicit control of scene properties
Generative networks that synthesize pixels

Neural Rendering, a definition:

Deep neural networks for image or video generation that enable explicit or implicit control of scene properties

Generative networks that synthesize pixels

Controllable by interpretable parameters or by conditioning input

Neural Rendering, a definition:

Deep neural networks for image or video generation that enable explicit or implicit control of scene properties

Generative networks that synthesize pixels

Controllable by interpretable parameters or by conditioning input

Illumination, camera, pose, geometry, appearance, semantic structure, animation, …

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Topic	Speakers
Introduction and Fundamentals	M. Zollhoefer
Generative Adversarial Networks	J-Y. Zhu, A. Tewari, G. Wetzstein
Novel View Synthesis for Objects of Scenes	V. Sitzmann, B. Mildenhall, L. Liu, D. B Goldman
Learning to Relight	Z. Xu, S. Orts-Escolano, P. Srinivasan
Compositional Scene Representations	M. Guo
Free Viewpoint Video	E. Tretschk, S. Lombardi, R. Pandey
Facial and Body Rendering	J. Thies, C. Theobalt, T. Simon
Open Challenges and Conclusion	O. Fried
Discussion on Ethical Implications	M. Agrawala
Panel Discussion & Questions	Join us!

Neural Rendering is expanding rapidly.

Can’t wait to see what you do next!

www.neuralrender.com