CSC2547 3D & Geometric Deep Learning

Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, Yaron Lipman
Weizmann Institute of Science

Date: February 16, 2021

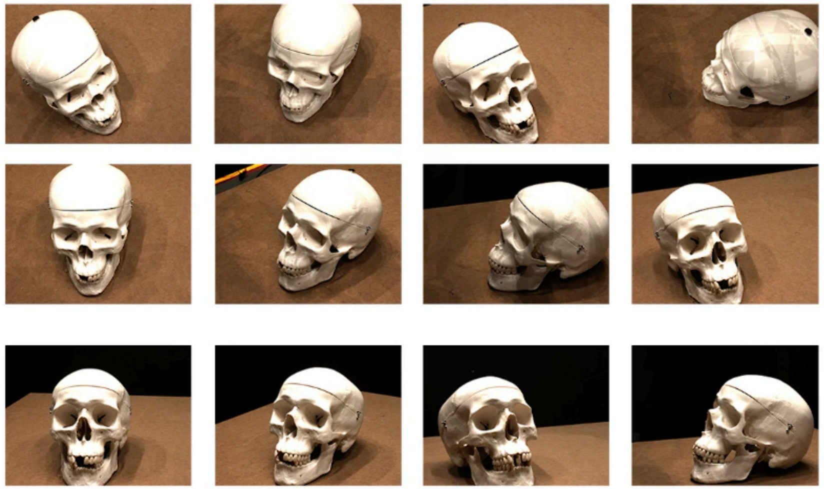
Presenter: Wenzhi Guo

Instructor: Animesh Garg

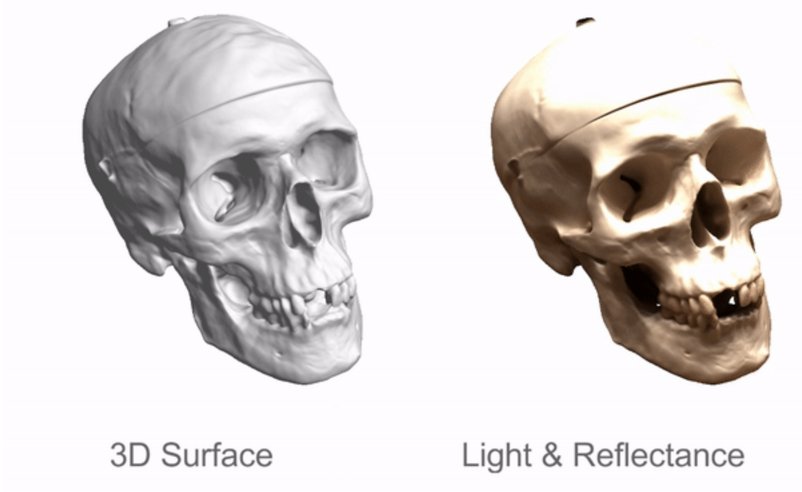
UNIVERSITY OF TORONTO

Main Problem

- Multiview 3D surface reconstruction from 2D images



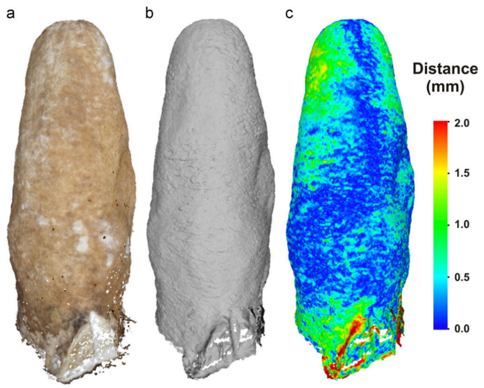
Input



Output

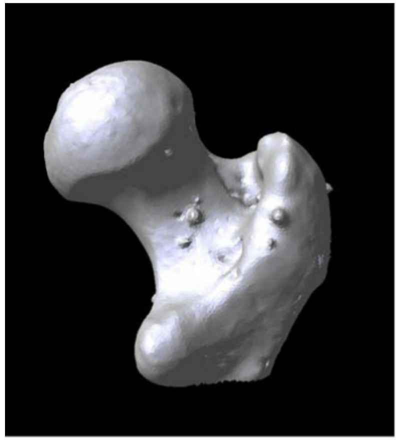
Motivation

Applications:



Geoscience[1]

Robotics/Navigation[2]



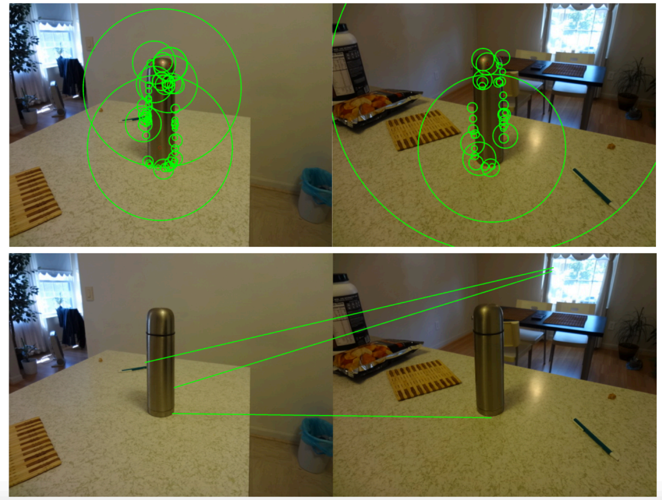
Medical Applications[3]

Entertainment[4]

1. https://doi.org/10.1016/j.cageo.2011.09.012
2. https://www.spotnik.net/blogs/4/Mobile%20Automata%20Robot%20With%20Agents%20Network
3. https://doi.org/10.1016/j.media.2008.12.003
4. https://www.wired.com/2015/02/3d-printed-selfies/

Challenges

- Ambiguities in feature matching -> hard to get an accurate and dense reconstruction
- Missing camera information
- Fine structure capture
- Occlusion handling for multiple objects
- Memory/computation limitations
- Post processing steps for surface reconstruction



http://www.cs.cmu.edu/~ehsiao/thesis/ehsiao_thesis.pdf

Prior Work (Known Camera)

- Recover depth information with Multi-View Stereo (MVS) via feature matching

  - often require post-processing steps for surface reconstruction

- Neural representation:

  - (Vincent S., et al., 2019): Encode scene geometry with LSTM to simulate ray marching

  - Nerf (Ben M. et al., 2020): NN to predict volume density and view dependent radiance

  - (Michael O. et al., 2020): use NN to learn the surface light fields

  - Can’t handle unknown cameras or 3D surface reconstruction

Prior Work (Unknown Camera)

- Structure From Motion (SfM) :

  - estimate camera and 3D representation jointly

  - (Chengzhou T. et al, 2019): Use a reference frame to help with depth estimation and features from nearby images to help with depth and camera parameters

  - only sparse representation (e.g. point cloud)

Contributions

- Introduces an end-to-end architecture that handles unknown geometry, appearance, and cameras (unknown camera + no post-processing)
- Produces SOTA watertight 3D surface reconstructions of different objects with a wide range of appearances (no feature matching + general appearance model)
- Demonstrates the disentangled geometry and appearance representation

General Background

- Implicit Differentiable Renderer (IDR):
- **Color of pixel**: differentiable function in the three unknowns of a scene: geometry, appearance, and the cameras.
- **Appearance**: all the factors that define the surface light field, *excluding* the geometry, i.e., the surface bidirectional reflectance distribution function (BRDF) and the scene’s lighting conditions.
- **Capability**: all surface light fields that can be represented as continuous functions of the point on the surface, its normal, and the viewing direction.

DeepSDF(Recap)

- Signed Distance Function (SDF):

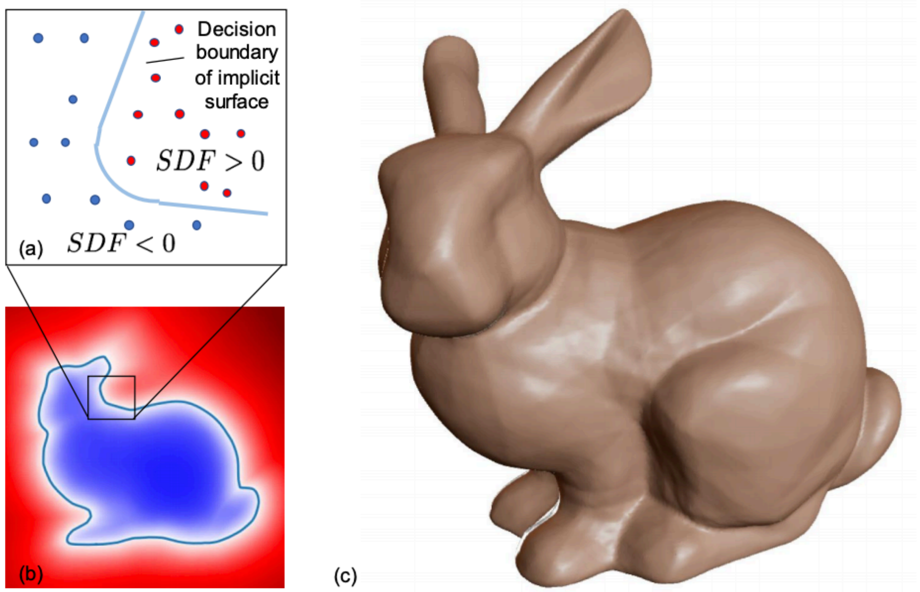
Implicit shape representation

Represents distance to surface (SDF = 0)

- DeepSDF:

MLP to approximate SDF in continuous space

Input: 3D coordinates + shape code; output: SDF value



Deep SDF: Learning Continuous Signed Distance Functions for Shape Representation

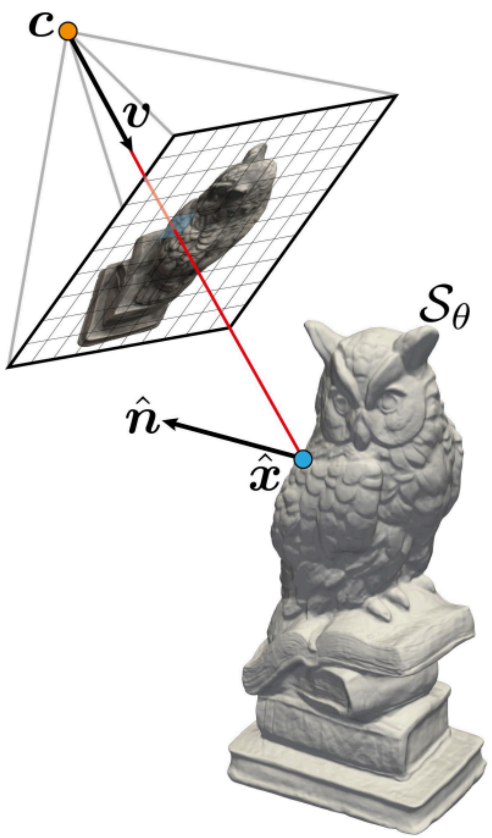
Ray Casting (Overview)

- An algorithm for realistic rendering
- For every pixel:

a) Construct a ray from the viewer/camera center

b) Find the intersection with the scene

c) Find the color*



* Color depends on many factors such as:

Light properties, material properties, surface properties

Problem Setting

- Unknowns: geometry (θ), appearance(γ), cameras(τ)
- Setup (fixed pixel p):

c: unknown center of the respective camera

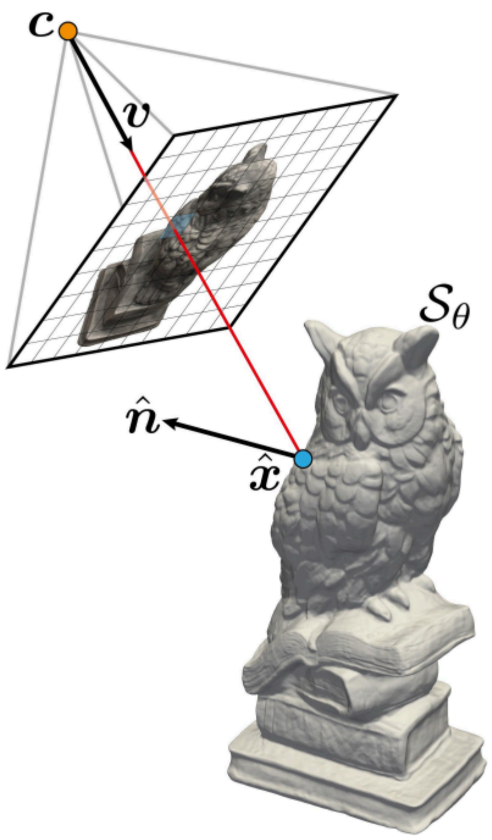
v: direction of ray

$\hat{x}$: first intersection of the ray and the surface S_θ

$\hat{n}$: surface normal at $\hat{x}$

$\hat{z}$: global geometry feature vector

- Rendered color: $L(\theta, \gamma, \tau) = M(\hat{x}, \hat{n}, \hat{z}, v, \gamma)$



Algorithm (Intersection + Surface Normal)

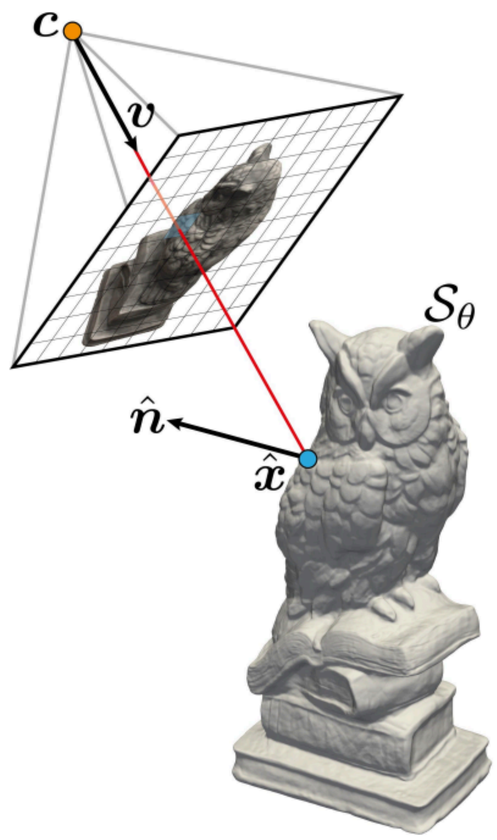
- Intersection point: $\hat{\boldsymbol{x}}(\theta, \tau) = \boldsymbol{c} + t(\theta, \boldsymbol{c}, \boldsymbol{v})\boldsymbol{v}$
- Find $\hat{\boldsymbol{x}}$ in a gradient descent-like algorithm

- by implicit Differentiation:

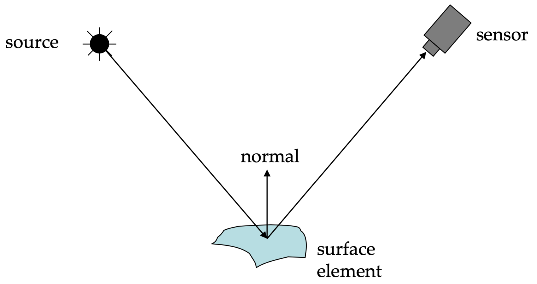
$$\hat{\boldsymbol{x}}(\theta, \tau) = \boldsymbol{c} + t_0\boldsymbol{v} - \frac{\boldsymbol{v}}{\nabla_x f(\boldsymbol{x_0}; \theta_0) \cdot \boldsymbol{v_0}} f(\boldsymbol{c} + t_0\boldsymbol{v}; \theta)$$

- Normal vector:

$$\hat{\boldsymbol{n}}(\theta, \tau) = \nabla_x f(\hat{\boldsymbol{x}}(\theta, \tau), \theta)$$



Algorithm (Surface Light Field)



BRDF:https://www.cs.cmu.edu/afs/cs/academic/class/15462-f09/www/lec/lec8.pdf

- Surface light field radiance:

$$L(\hat{\boldsymbol{x}}, \boldsymbol{w}^o) = L^e(\hat{\boldsymbol{x}}, \boldsymbol{w}^o) + \int_{\Omega} B(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{w}^i, \boldsymbol{w}^o) L^i(\hat{\boldsymbol{x}}, \boldsymbol{w}^i)(\hat{\boldsymbol{n}} \cdot \boldsymbol{w}^i) \mathrm{d}\boldsymbol{w}^i$$

- $L^e(\hat{\boldsymbol{x}}, \boldsymbol{w}^o)$: light sources (emitted radiance of light by the surface)
- $B(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{w}^i, \boldsymbol{w}^o)$: BRDF (reflectance and color properties of the surface)
- $L^i(\hat{\boldsymbol{x}}, \boldsymbol{w}^i)$: incoming radiance
- $\hat{\boldsymbol{n}} \cdot \boldsymbol{w}^i$: weakening factor (non-orthogonal incoming light)
- Ω : half sphere centered at $\hat{\boldsymbol{n}}$

Algorithm (Surface Light Field)

- Continuous Function: $L(\hat{\boldsymbol{x}}, \boldsymbol{w}^o) = M_0(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{v})$
- Using MLP (M) to approximate M_0 : $L(\theta, \gamma, \tau) = M(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{v}, \gamma)$
- ***v*** and ***n*** are necessary parameters to be able to learn appearance independent from geometry and work with general appearance model (e.g. Phong reflection model)
- Global feature vector $\hat{\boldsymbol{z}}$ **(input to the renderer):**

  - encode the geometry relative to the surface sample **x**

  - global lighting effects: secondary lighting + self shadows

Algorithm (Masked Rendering)

- 2D supervision of geometry: binary mask (foreground/background)

- Test for pixel occupancy (ray intersection):

$$S(\theta, \tau) = \begin{Bmatrix} 1 & R(\tau) \cap S_\theta \neq \emptyset \\ 0 & otherwise \end{Bmatrix}$$

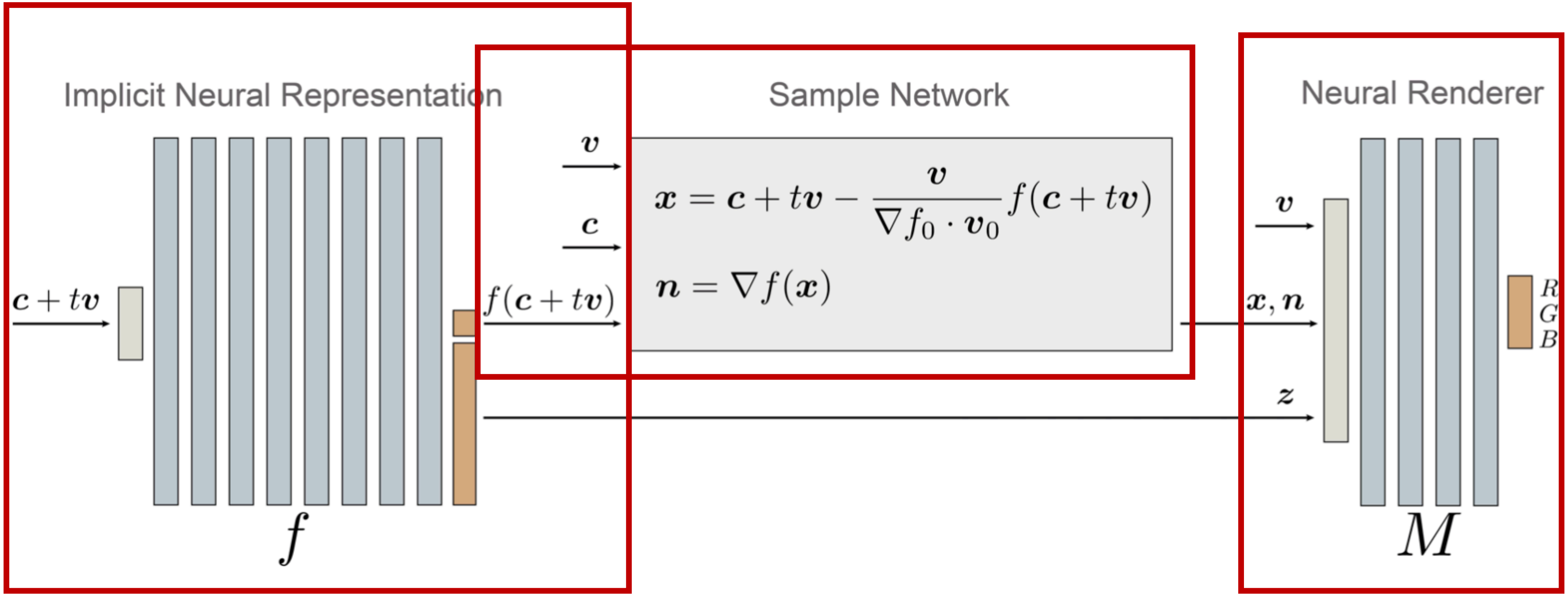
- Approximation (differentiable):

$$S_\alpha(\theta, \tau) = sigmoid(-\alpha \min_{t \geq 0} f(\boldsymbol{c} + t\boldsymbol{v}; \theta))$$

Loss Function

- $loss(\theta, \gamma, \tau) = loss_{RGB}(\theta, \gamma, \tau) + \rho loss_{MASK}(\theta, \tau) + loss_{E}\ (\theta)$
- Appearance: $loss_{RGB}(\theta, \gamma, \tau) = \frac{1}{|P|} \sum_{p \in P^{in}} \left| I_p\ -\ L_p(\theta, \gamma, \tau) \right|$
- Geometry: $loss_{MASK}(\theta, \tau) = \frac{1}{\alpha |P|} \sum_{p \in P^{out}} CE(O_p, S_{p,\alpha}\ (\theta, \tau))$
- Regularizer (Eikonal): $loss_{E}(\theta) = E_x(\ \|\nabla_x f(x; \theta)\| - 1)\ ^2$

End-to-end Network



Evaluation Dataset

- Dataset: DTU MVS dataset
- 15 scans (49 or 64 high resolution images)
- Manually annotation of binary masks
- Contains ground truth 3D geometries and camera poses

3D Reconstruction Results (Fixed Camera)

Baseline Methods

Proposed Method

Scan Index

Scan	Watertight Mesh							
	Colmap$_{trim=0}$		**Furu**$_{trim=0}$		**DVR** [40]		**IDR**	
	Chamfer	PSNR	Chamfer	PSNR	Chamfer	PSNR	Chamfer	PSNR
24	**0.81**	20.28	0.85	20.35	4.10(4.24)	16.23(15.66)	1.63	**23.29**
37	2.05	15.5	**1.87**	14.86	4.54(4.33)	13.93(14.47)	**1.87**	**21.36**
40	0.73	20.71	0.96	20.46	4.24(3.27)	18.15(19.45)	**0.63**	**24.39**
55	1.22	20.76	1.10	21.36	2.61(0.88)	17.14(18.47)	**0.48**	**22.96**
63	1.79	20.57	2.08	16.75	4.34(3.42)	17.84(18.42)	**1.04**	**23.22**
65	1.58	14.54	2.06	13.53	2.81(1.04)	17.23(20.42)	**0.79**	**23.94**
69	1.02	21.89	1.11	**21.62**	2.53(1.37)	16.33(16.78)	**0.77**	20.34
83	3.05	**23.2**	2.97	20.06	2.93(2.51)	18.1(19.01)	**1.33**	21.87
97	1.4	18.48	1.63	18.32	3.03(2.42)	16.61(16.66)	**1.16**	**22.95**
105	2.05	21.3	1.88	20.21	3.24(2.42)	18.39(19.19)	**0.76**	**22.71**
106	1.0	22.33	1.39	22.64	2.51(1.18)	17.39(18.1)	**0.67**	**22.81**
110	1.32	18.25	1.45	17.88	4.80(4.32)	14.43(15.4)	**0.9**	**21.26**
114	0.49	20.28	0.69	20.09	3.09(1.04)	17.08(20.86)	**0.42**	**25.35**
118	0.78	25.39	1.10	**26.02**	1.63(0.91)	19.08(19.5)	**0.51**	23.54
122	1.17	25.29	1.16	25.95	1.58(0.84)	21.03(22.51)	**0.53**	**27.98**
Mean	1.36	20.58	1.49	20.01	3.20(2.28)	17.26(18.33)	**0.9**	**23.20**

Chamfer Distance (Geometry)

Peak Signal to Noise Ratio (Appearance)

3D Reconstruction Results (Fixed Camera)

3D Reconstruction Results (Trained Camera)

Scan	Watertight Mesh			
	Colmap$_{trim=0}$		**IDR**	
	Chamfer	PSNR	Chamfer	PSNR
24	**0.73**	20.46	1.96	**23.16**
37	**1.96**	15.51	2.92	**20.39**
40	**0.67**	20.86	0.7	**24.45**
55	1.17	21.22	**0.4**	**23.57**
63	1.8	20.67	**1.19**	**24.97**
65	1.61	14.59	**0.77**	**22.6**
69	1.03	21.93	**0.75**	**22.91**
83	3.07	**23.43**	**1.42**	21.97
97	1.37	**18.67**	-	-
105	2.03	21.22	**0.96**	**22.98**
106	0.93	**22.23**	**0.65**	21.18
110	**1.53**	18.28	2.84	**18.65**
114	**0.46**	20.25	0.51	**25.19**
118	0.74	**25.42**	**0.50**	22.58
122	1.17	**25.44**	**0.62**	24.42
Mean	1.35	20.68	**1.16**	**22.79**

Disentangling Geometry and Appearance

Render geometry network (f) and renderer (M) trained on different scenes:

Ablation Study

a) Remove viewing direction v

b) Remove surface normal $\hat{n}$

c) Remove feature vector $\hat{z}$

d) full blown renderer M

e) No camera optimization

Discussion of results

- The proposed method can produce SOTA 3D reconstruction for both fixed camera and trained camera cases

- It also showcases a way to optimize camera parameters and 3D geometry jointly

- It demonstrates that it is possible to disentangle the representation for geometry and appearance

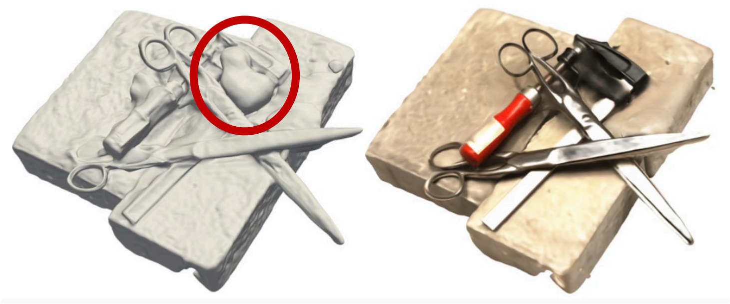
Limitations (Algorithm)

- Needs a reasonable camera initialization (camera optimization)

- Fails to capture fine structures sometimes:

Fixed cameras

Trained cameras



- Requires a binary mask for background/foreground:

$$loss(\theta,\gamma,\tau) = loss_{RGB}(\theta,\gamma,\tau) + \rho loss_{MASK}(\theta,\tau) + loss_E\ (\theta)$$

Missing Results + Critiques

- Does not include training/inference time comparison
- Does not include the effect of # of input images
  - More images -> better results?
  - Minimum number of input images required?
- [it] can only represent single scenes with the original lighting[1]
- It only works with "static scene without moving objects" [2]

1. Learning Implicit Surface Light Fields: 10.1109/3DV50981.2020.00055
2. D-NeRF: Neural Radiance Fields for Dynamic Scenes : arXiv:2011.13961

Contributions (Recap)

- Main Problem: Multiview 3D surface reconstruction from 2D images
- Contributions:

a) Introduces an end-to-end architecture that handles unknown geometry, appearance, and cameras

b) Produces SOTA watertight 3D surface reconstructions of different objects with a wide range of appearances

c) Demonstrates the disentangled geometry and appearance representation