3D Reconstruction with Fast Dipole Sums

MS Thesis Presentation

Hanyu Chen

April 29, 2024

Carnegie Mellon University

Table of contents

1. Introduction
2. Volumetric Neural Rendering
3. Winding Number & Dipole Sums
4. Rendering
5. Results
6. Conclusion

Introduction

3D Reconstruction

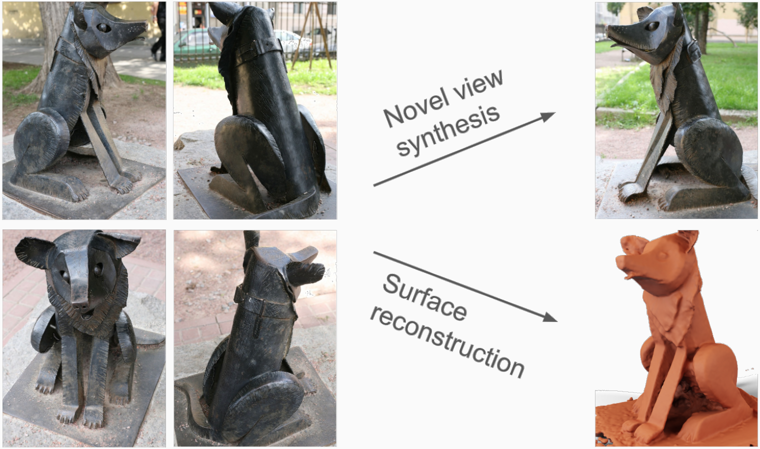
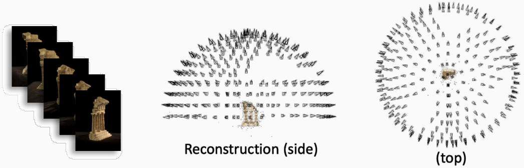


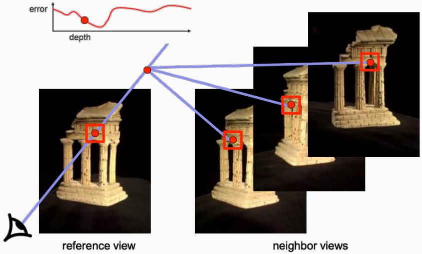
Figure 1: Dog scene from the Blended MVS dataset

Traditional Methods

- Structure from motion



- Multi-view stereo



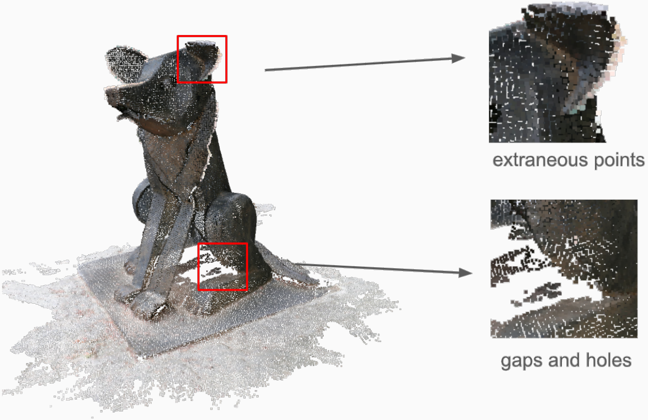
Kanazawa and Efros [2021]

Traditional Methods

What do the outputs look like?

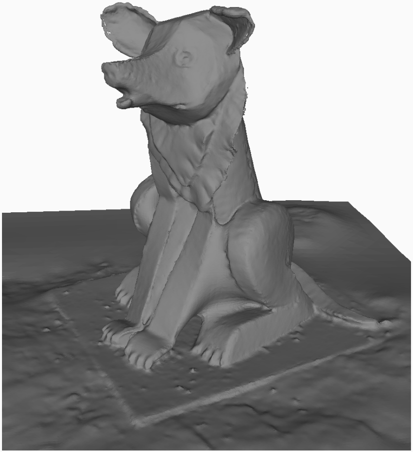
Traditional Methods

Looks pretty good, except...



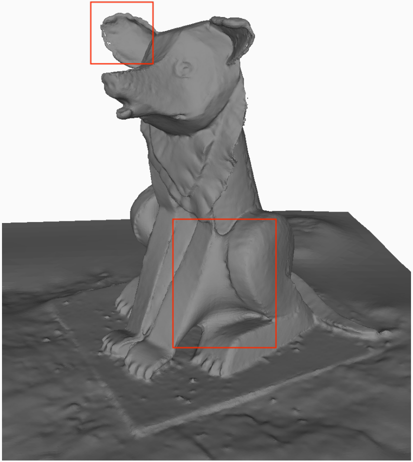
Traditional Methods

Typically, Poisson surface reconstruction [Kazhdan et al., 2006] is used to extract surfaces.



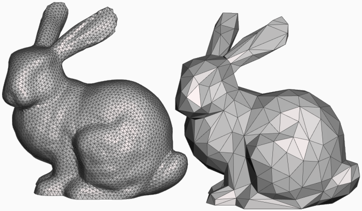
Traditional Methods

… and we see the same issues

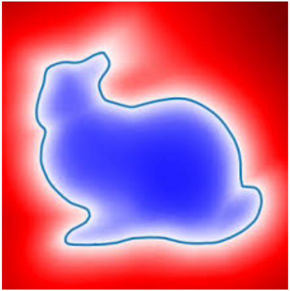


Volumetric Neural Rendering

Surface Rendering



(a) Triangle mesh



(b) Signed-distance function

Schmidt [2017], Park et al. [2019]

Volume Rendering

$$
\begin{aligned}
dL(\mathbf{x},\vec{\omega}) = &-\sigma_a(\mathbf{x})L(\mathbf{x},\vec{\omega})dz \quad -\sigma_s(\mathbf{x})L(\mathbf{x},\vec{\omega})dz \\
&+\sigma_a(\mathbf{x})L_e(\mathbf{x},\vec{\omega})dz + \sigma_s(\mathbf{x})L_s(\mathbf{x},\vec{\omega})dz
\end{aligned}
$$

Bitterli et al. [2018], Gkioulekas [2024]

Rendering Equation

Assuming an *emissive* volume with no scattering, we can derive the rendering equation as

$$L(\mathbf{x}, \vec{\omega}) = \int_0^z \mathrm{Tr}(\mathbf{x}, \mathbf{x}_t) \sigma(\mathbf{x}_t, \vec{\omega}) L_e(\mathbf{x}_t, \vec{\omega}) \, \mathrm{d}t,$$

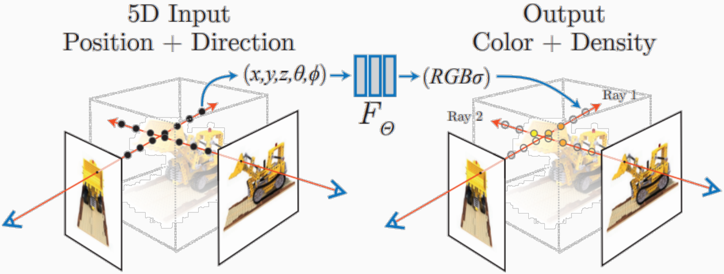
where $\mathrm{Tr}(\mathbf{x}, \mathbf{x}_t)$ is the transmittance

$$\mathrm{Tr}(\mathbf{x}, \mathbf{x}_t) = \exp\left(-\int_0^t \sigma(\mathbf{x}_s, \vec{\omega}) \, \mathrm{d}s\right),$$

and $\sigma(\mathbf{x}_s, \vec{\omega})$ is the spatially varying attenuation coefficient.

Neural Radiance Fields (NeRF)

Representing objects as volumes (instead of surfaces).



Color $L_e(\mathbf{x}, \vec{\omega})$ and density $\sigma(\mathbf{x}, \vec{\omega})$ modeled by a neural network.

[Mildenhall et al., 2020]

Neural Radiance Fields (NeRF)

NeRFs work particularly well for novel-view synthesis tasks
… but not so much for surface extraction.

[Mildenhall et al., 2020]

Neural Surface Representations

Can we explicitly model the geometry of the scene?

Yes, there are many works that already do this:

- Occupancy $\leftrightarrow$ density (NeuS [Wang et al., 2021])
- Signed-distance function $\leftrightarrow$ density (VolSDF [Yariv et al., 2021])

Follow-up works improve reconstruction quality and efficiency:

- Multi-resolution hash grids (Neuralangelo [Li et al., 2023], NeuS2 [Wang et al., 2023])
- Photo-consistency constraints (Neuralwarp [Darmon et al., 2022], Geo-NeuS [Fu et al., 2022])
- Sparse voxel representation (Voxurf [Wu et al., 2023])

Neural Surface Representations

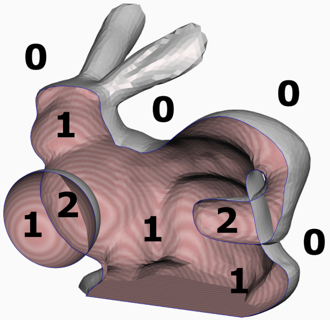
Limitations of existing methods:

- Trade-off between speed and reconstruction quality
- Cannot effectively leverage known information

Our solution: Directly build off of the output of traditional methods (i.e. dense point clouds)

Winding Number & Dipole Sums

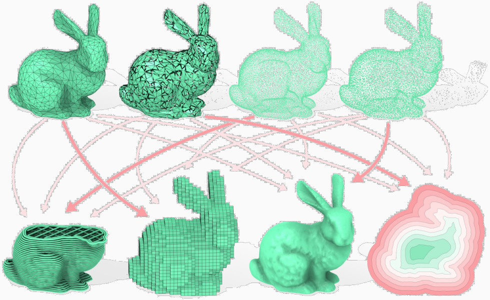
The Winding Number



$$w_S(\mathbf{q}) = \frac{1}{4\pi}\int_S \mathrm{d}\Omega(\mathbf{q}) = \int_S \frac{(\mathbf{p}-\mathbf{q})\cdot\hat{\mathbf{n}}}{4\pi\|\mathbf{p}-\mathbf{q}\|^3}\,\mathrm{d}\mathbf{p}$$

Schmidt [2018], Barill et al. [2018]

The Generalized Winding Number



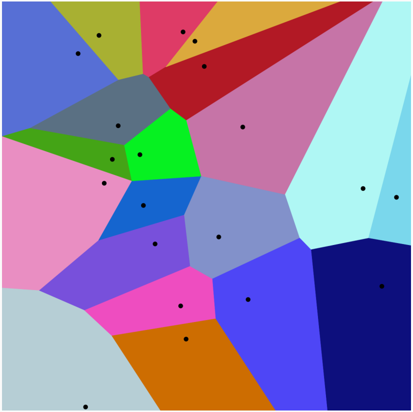
For point clouds,

$$w_S(\mathbf{q}) \approx \sum_{m=1}^{N} a_m \frac{(\mathbf{p}_m - \mathbf{q}) \cdot \hat{\mathbf{n}}_m}{4\pi \|\mathbf{p}_m - \mathbf{q}\|^3}$$

where a_m are “area-weights” (e.g. geodesic Voronoi area).

Barill et al. [2018]

The Generalized Winding Number



Aside: Laplace BVP

The winding numbers are also the solutions to the Laplace boundary value problem (BVP)

$$\begin{cases} \Delta w(\mathbf{p}) = 0, & \mathbf{p} \in \mathbb{R}^3 \backslash \partial\Omega \\ w^+(\mathbf{p}) - w^-(\mathbf{p}) = f(\mathbf{p}), & \mathbf{p} \in \partial\Omega \\ \frac{\partial w^+}{\partial \widehat{\mathbf{n}}}(\mathbf{p}) - \frac{\partial w^-}{\partial \widehat{\mathbf{n}}}(\mathbf{p}) = 0, & \mathbf{p} \in \partial\Omega, \end{cases}$$

for the case where $f(\mathbf{p}) \equiv 1$ on the boundary.

The Generalized Winding Number

Naively, we can directly extract the 0.5-isosurface of the winding number field. However, this often fails in practice...

What if we further generalize the winding number

… and allow $f(\mathbf{p}) \not\equiv 1$?

$$w_S(\mathbf{q}) = \int_S \frac{(\mathbf{p}-\mathbf{q}) \cdot \hat{\mathbf{n}}}{4\pi\|\mathbf{p}-\mathbf{q}\|^3} \cdot 1 \,\mathrm{d}\mathbf{p}$$

$$\Downarrow$$

$$u_S^f(\mathbf{q}) = \int_S \frac{(\mathbf{p}-\mathbf{q}) \cdot \hat{\mathbf{n}}}{4\pi\|\mathbf{p}-\mathbf{q}\|^3} \cdot f(\mathbf{p}) \,\mathrm{d}\mathbf{p}$$

The Dipole Sum

This is equivalent to having non-unit per-point attributes f_m:

$$w_{\mathrm{pc}}(\mathbf{q}) = \sum_{m=1}^{N} a_m \frac{(\mathbf{p}_m - \mathbf{q}) \cdot \hat{\mathbf{n}}_m}{4\pi \|\mathbf{p}_m - \mathbf{q}\|^3} \cdot 1$$

$$\Downarrow$$

$$u^f_{\mathrm{pc}}(\mathbf{q}) = \sum_{m=1}^{N} a_m \frac{(\mathbf{p}_m - \mathbf{q}) \cdot \hat{\mathbf{n}}_m}{4\pi \|\mathbf{p}_m - \mathbf{q}\|^3} \cdot f_m$$

In a differentiable rendering pipeline, we can make them learnable!

The Fast Dipole Sum

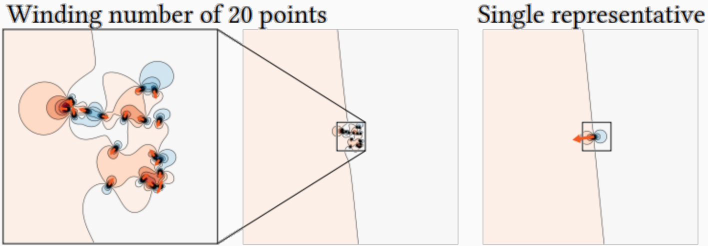
A dense point cloud can have tens of thousands of points

… and to render a single image, we need to potentially query the dipole sum at billions of distinct locations in the scene.

Obviously, we cannot compute the entire sum for every query…

The Barne-Hut Approximation

As proposed by Barill et al. [2018], we can consider a cluster of points far away from the query point as a single “dipole” with attributes computed using the Barnes-Hut approximation.



The Barne-Hut Approximation

We construct an octree from the point cloud, and assign each tree node t a centroid, a weighted normal, and a radius, computed from its leaves $\mathcal{L}(t)$:

$$\tilde{\mathbf{p}}_t \equiv \frac{\sum_{m \in \mathcal{L}(t)} a_m \mathbf{p}_m}{\sum_{m \in \mathcal{L}(t)} a_m}$$

$$\tilde{\mathbf{n}}_t^f \equiv \sum_{m \in \mathcal{L}(t)} a_m \hat{\mathbf{n}}_m \cdot f_m$$

$$\tilde{r}_t \equiv \max_{m \in \mathcal{L}(t)} \|\tilde{\mathbf{p}}_t - \mathbf{p}_m\|$$

To query the fast dipole sum, we recurse the octree with aggressive pruning in $O(\log N)$ time.

Fast Backpropogation

Naively backpropagating through the dipole sum takes $O(N \cdot M)$ time for M queries.

We propose a two-stage backprop scheme:

1. We detach the per-node parameters from the per-point parameters and cache gradients at nodes.
2. We use the chain rule to propogate gradients from the nodes to all leaves (points).

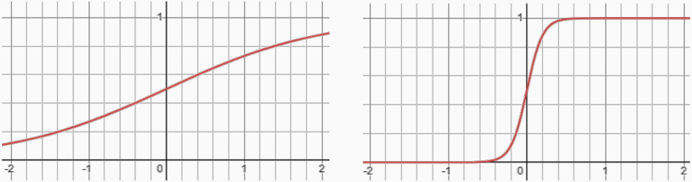
Each iteration of backprop only takes $O((M + N) \log N)$ time.

Rendering

To convert the dipole sum[†] into occupancy values between 0 and 1, we apply a logistic sigmoid

$$O(\mathbf{q}) = \frac{1}{1 + \exp\left(-s \cdot (u^f_{\mathrm{pc}}(\mathbf{q}) - 0.5)\right)}$$

with a learnable scale parameter s.



† We actually use a regularized version of the dipole sum to avoid singularities

Occupancy $\rightarrow$ Density

To convert the occupancy into density (attenuation coefficient) values, we follow Miller et al. [2023]:

$$\sigma(\mathbf{q}, \vec{\omega}) = \frac{|\nabla O(\mathbf{q}) \cdot \vec{\omega}|}{1 - O(\mathbf{q})}$$

Intuitively, we have

$\sigma(\mathbf{q}, \vec{\omega})$	High	Low
$O(\mathbf{q})$	High	Low
$\nabla O(\mathbf{q})$	Large magnitude	Small magnitude
$\vec{\omega}$	Normal incidence	Grazing angle

Neural Features

We can also use the dipole sum as an interpolation scheme to interpolate per-point neural features $\mathbf{l}_m \in \mathbb{R}^d$:

$$u_{\mathrm{pc}}^{\mathbf{l}}(\mathbf{q}) = \sum_{m=1}^{N} a_m \frac{(\mathbf{p}_m - \mathbf{q}) \cdot \hat{\mathbf{n}}_m}{4\pi \|\mathbf{p}_m - \mathbf{q}\|^3} \cdot \mathbf{l}_m$$

Unlike for geometry, we do not want sharp discontinuities in the interpolated features, so in practice, we instead use

$$u_{\mathrm{pc}}^{\mathbf{l}}(\mathbf{q}) = \sum_{m=1}^{N} a_m \frac{1}{4\pi \|\mathbf{p}_m - \mathbf{q}\|^2} \cdot \mathbf{l}_m$$

Neural Features

We predict colors using a simple neural network

$$L_e(\mathbf{q}, \vec{\omega}) \approx \mathrm{NN}(u^{\mathbf{l}}_{\mathrm{pc}}(\mathbf{q}), \mathrm{Enc}(\vec{\omega}))$$

with a directional encoding function that encodes the viewing direction using spherical harmonics, following Verbin et al. [2022].

Results

Setup

We implement our method in PyTorch based on the NeuS codebase [Wang et al., 2021]. Dipole sum and octree operations are implemented in C++ and CUDA with PyTorch bindings.

We evaluate our method against Neuralangelo [Li et al., 2023] on the DTU dataset [Jensen et al., 2014] and BlendedMVS dataset [Yao et al., 2020] on a single NVIDIA RTX4090 GPU. Each scene takes around 3-4 hours to fully train.

Results: DTU

We quantitatively evaluate our method on the DTU dataset using the Chamfer distance metric (↓).

Scan	Ours	Neuralangelo
24	0.46	**0.37**
37	**0.65**	0.72
40	**0.33**	0.35
55	**0.33**	0.35
63	0.95	**0.87**
65	0.78	**0.54**
69	**0.53**	**0.53**
83	**1.23**	1.29
97	**0.84**	0.97
105	**0.70**	0.73
106	**0.46**	0.47
110	**0.55**	0.74
114	0.33	**0.32**
118	**0.37**	0.41
122	**0.36**	0.43
Mean†	**0.50**	0.52

† We exclude scans 63, 83, 105 from the mean calculation due to inaccurate ground truths.

Results: BlendedMVS

Qualitative equal-time comparisons on the BlendedMVS dataset:

Results: BlendedMVS

Neuralangelo completely fails when there are few views available:

Visualizations: Occupancy & Color

We visualize the occupancy and color on slices of the scene:

Conclusion

Summary

We propose a novel method for 3D reconstruction that combines the efficiency of point clouds (via the Barnes-Hut approximation) with the expressiveness of neural rendering (via learnable boundary conditions and neural features).

Our method achieves results comparable to the state-of-the-art in less than a quarter of the training time.

Our representation is extremely compact: the entire model consists of an oriented point cloud with a single additional scalar attribute at each point (< 50 MB).

Limitations

Our reconstruction quality is highly dependent on the quality of the input point cloud, as we do not optimize point locations.

For larger scale scenes, estimating a dense point cloud with Colmap can be a bottleneck for performance.

Our method inherits certain limitations from Colmap, such as the difficulty in handling textureless surfaces and highly specular surfaces, where points are noisy or completely missing.

Questions?

References i

Gavin Barill, Nia Dickson, Ryan Schmidt, David I.W. Levin, and Alec Jacobson. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics*, 2018.

Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. A radiative transfer framework for non-exponential media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 37(6): 225:1–225:17, November 2018. doi: 10/gfz2cm.

François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *CVPR*, 2022.

Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ioannis Gkioulekas. Participating media. `http://graphics.cs.cmu.edu/courses/15-468/lectures/lecture_12.pdf`, 2024. [Online; accessed 27-April-2024].

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

Angjoo Kanazawa and Alexei Efros. Structure-from-Motion (SfM) and Multi-View Stereo (MVS). https://inst.eecs.berkeley.edu/~cs194-26/fa21/Lectures/mvssfm.pdf, 2021. [Online; accessed 27-April-2024].

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Bailey Miller, Hanyu Chen, Alice Lai, and Ioannis Gkioulekas. A theory of volumetric representations for opaque solids. *arXiv*, 2023.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ryan Schmidt. Mesh Simplification with g3Sharp. `https://www.gradientspace.com/tutorials/2017/8/30/mesh-simplification`, 2017. [Online; accessed 27-April-2024].

Ryan Schmidt. Surfacing Point Sets with Fast Winding Numbers. `https://www.gradientspace.com/tutorials/2018/9/14/point-set-fast-winding`, 2018. [Online; accessed 27-April-2024].

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021.

Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *International Conference on Learning Representations (ICLR)*, 2023.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Thank you!