

From Projection to Perception: A Mathematical Exploration of Shadow-based Neural Reconstruction

A research report submitted to the Scientific Committee of the Hang Lung Mathematics Award

Team Number

2596873

Team Members

Wong Yuk To, Hung Kwong Lam

Cheung Tsz Lung, Chan Ngo Tin, Zhou Lam Ho

Teacher

Mr. Chan Ping Ho

School

Po Leung Kuk Celine Ho Yam Tong College

Date

August 13, 2025

Abstract

This paper explores SHADOWNEUS [\[LWX23\]](#), a neural network that reconstructs 3D geometry from single-view camera images using shadow and light cues. Unlike traditional 3D reconstruction methods relying on multi-view cameras or sensors, SHADOWNEUS leverages a neural signed distance field (SDF) for accurate 3D geometry reconstruction. We analyze the training process and uncover its connections to projective geometry, spatial reasoning in \mathbb{R}^3 , and the neural network's learned geometric representation of space.

Contents

1	Introduction	2
1.1	Background	2
1.2	Motivation	2
2	Fundamentals of 3D Reconstruction from 2D Images	2
2.1	3D Reconstruction in Computer Vision	2
2.2	Information Encoded in a Single-View Image	3
2.3	Forward Projection: Mapping from 3D to 2D	3
2.4	The Inverse Problem: From 2D Image to 3D World	4
2.5	Ill-posedness and Ambiguity in Single-View Reconstruction	5
3	Shadows as Geometric Constraints	5
3.1	Light Rays and Shadow Formation due to Blockage	5
3.2	Surface Normals and Shadow Formation due to Orientation	6
4	ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision	7
5	Experimental Demonstration	7
6	Discussion and Analysis	7
7	Conclusion	7

1 Introduction

1.1 Background

3D reconstruction is the process of recovering the shape and structure of an object in \mathbb{R}^3 from measured data. It has broad applications in medical imaging (MRI, CT), robotics, augmented/virtual reality (AR/VR), and cultural heritage preservation. Most conventional methods rely heavily on geometric and physical modeling techniques and require rich spatial data obtained through multi-view camera setups, LiDAR sensors, or photogrammetry.

1.2 Motivation

Traditional approaches typically depend on **multiple viewpoints or depth sensors**, which is costly and complex. This leads to the question: **Is it possible to reconstruct 3D geometry from a single fixed camera?** A single image inherently lacks depth information, and multiple 3D points can project to the same 2D pixel location (as discussed in Section 2.5). Therefore, additional cues are essential to resolve this ambiguity.

In exploring this problem, we discovered the paper *ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision* [LWX23] by Jingwang Ling, Zhibo Wang, and Feng Xu. Their approach demonstrates that leveraging neural signed distance fields and supervising the network with shadow ray information under varying lighting enables accurate 3D reconstruction from single-view images. Motivated by their work, we present a study analyzing their method and propose a 1D-to-2D experimental validation to examine our understanding.

2 Fundamentals of 3D Reconstruction from 2D Images

2.1 3D Reconstruction in Computer Vision

According to an article on Medium [VK23], 3D computer vision is a field of computer science focusing on the analysis, interpretation, and understanding of three-dimensional visual data.

The article highlights a traditional approach for 3D reconstruction:

Structure from Motion (SfM)

Recover the 3D structure by estimating camera positions from multiple images.

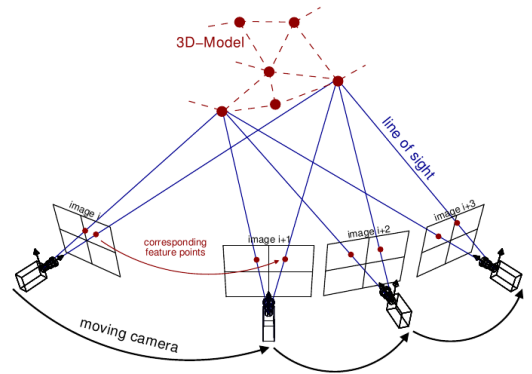


Figure 1. Structure from Motion

2.2 Information Encoded in a Single-View Image

When only single-view is available, the following information remains exploitable:

- **Pixel position** (u, v) — relates to a 3D ray from camera to that pixel
- **Color / intensity** — encodes surface information (material, orientation, etc.)
- **Embedded features** (edges, recognized objects) — encode geometry cues

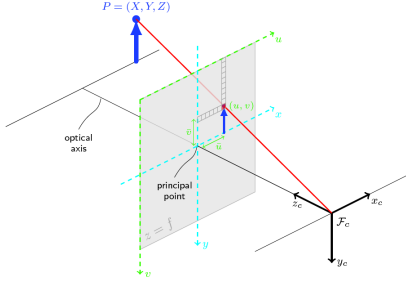


Figure 2. Ray from camera to pixel



Figure 3. Texture

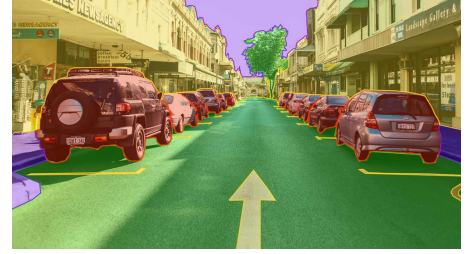


Figure 4. Object segmentation (YOLO)

2.3 Forward Projection: Mapping from 3D to 2D

In this section, we want to show that mapping from a 3D point $\mathbf{P} = (x, y, z)^T \in \mathbb{R}^3$ in the world coordinate system onto a 2D image pixel coordinate $\mathbf{p} = (u, v)^T \in \mathbb{R}^2$ is a projection-like process.

Extrinsic parameters: define the camera's position and orientation relative to the world:

$$\underbrace{C = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}}_{\text{Camera center}} \in \mathbb{R}^3, \quad \underbrace{R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{\text{Rotation matrix}} \in \text{SO}(3), \quad t = \underbrace{-RC}_{\text{Translation vector}} \in \mathbb{R}^3 \quad (1)$$

Intrinsic parameters: encode the internal camera geometry:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (2)$$

where (f_x, f_y) are the focal lengths in pixels, and (c_x, c_y) is the principal point (image center).

The forward projection consists of three steps:

1. Transform \mathbf{P} from world coordinates to camera coordinates: $\mathbf{P}_{\text{cam}} = R(\mathbf{P} - C) = R\mathbf{P} + t$.
2. Project camera coordinates to homogeneous image coordinates: $\mathbf{P}_{\text{hom}} = K\mathbf{P}_{\text{cam}} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix}$.
3. Normalize by depth z' to get pixel coordinates: $\mathbf{p} = \frac{1}{z'} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix}, \quad z' \neq 0$.

Combined expressions:

$$\boxed{\mathbf{P}_{\text{hom}} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix} = K[R \mid t] \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}, \quad \mathbf{p} = \frac{1}{z'} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix}} \quad (3)$$

2.4 The Inverse Problem: From 2D Image to 3D World

After understanding the forward projection process, 3D reconstruction can be viewed as the inverse calculation: recovering 3D points from their 2D image projections.

We now tackle the challenge of inverting the projection function introduced in Section 2.3.

Lemma 2.4.1 (Camera Ray Parametrization).

Given a pixel $p = [p_x, p_y]^\top$ and camera parameters (K, R, C) , the corresponding 3D points lie on a ray:

$$P(\lambda) = C + \lambda d, \quad \lambda > 0 \quad (4)$$

where the ray direction is

$$d = R^{-1} K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad (5)$$

Remark 2.4.1 (Normalization).

The vector d can be normalized to unit length for physical interpretation, but this is not essential for the ray parametrization.

Proof. Starting from the forward projection (3):

$$\begin{aligned} K(RP + t) &= z' \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \\ RP + t &= z' K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \\ P &= R^{-1} (z' K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} - t). \end{aligned}$$

Since $t = -RC$, we have $-R^{-1}t = C$. Setting $\lambda = z'$, the parametric form of the ray follows:

$$P(\lambda) = C + \lambda R^{-1} K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}.$$

□

By parametrizing the ray originating at C in the direction d , we capture the geometric meaning that a single pixel in an image does not correspond to a unique 3D point but rather to an infinite set of points lying along this ray.

2.5 Ill-posedness and Ambiguity in Single-View Reconstruction

Recovering 3D points from a single 2D image is **ill-posed**, as defined by Hadamard: the solution may be non-unique, unstable, or nonexistent.

Proposition 2.5.1 (Ill-posed Nature of Single-View Reconstruction).

- (a) The depth λ is unknown.
- (b) Each pixel \mathbf{p} corresponds to infinitely many 3D points along a ray.
- (c) Additional constraints are needed for unique reconstruction.

This ill-posedness motivates using extra cues like multiple views (SfM), depth sensors, or shadow information.

3 Shadows as Geometric Constraints

To what extent can shadows constrain 3D geometry from a single image? In this section, we will explore how shadows form from interaction of light and geometry, how these shadows partition the 3D surface and the 2D image, and how this partitioning can help recover depth information.

3.1 Light Rays and Shadow Formation due to Blockage

Definition 3.1.1 (Light Ray).

Given a point light source $\mathbf{L} \in \mathbb{R}^3$ and a surface point $\mathbf{P} \in \mathbb{R}^3$, the light ray is:

$$\boxed{r(t) = \mathbf{L} + t(\mathbf{P} - \mathbf{L}), \quad t \in [0, 1]} \quad (6)$$

Definition 3.1.2 (Shadow Occlusion Test).

A point \mathbf{P} is in shadow if there exists $t \in (0, 1)$ such that the light ray intersects a surface \mathcal{S} :

$$\boxed{r(t) \cap \mathcal{S} \neq \emptyset, \quad t \in (0, 1)} \quad (7)$$

Remark 3.1.1 (Physical Interpretation).

The interval $(0, 1)$ excludes the light source ($t = 0$) and the target point ($t = 1$), ensuring the test checks for obstructions between \mathbf{L} and \mathbf{P} . This models physical light transport where occlusion by another surface causes a shadow.

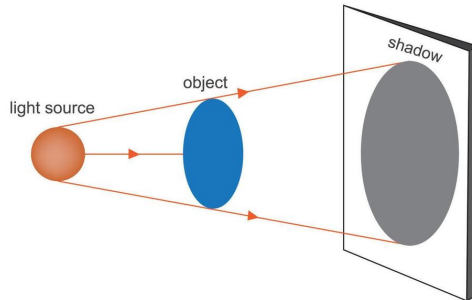


Figure 5. Shadow Formation due to Blockage

3.2 Surface Normals and Shadow Formation due to Orientation

Theorem 3.2.1 (Tangency Condition).

A point $\mathbf{Q} \in \mathcal{S}$ lies on the shadow boundary if and only if the light direction at \mathbf{Q} is tangent to the surface:

$$\boxed{(\mathbf{Q} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{Q}) = 0} \quad (8)$$

where $\mathbf{n}(\mathbf{Q})$ is the outward unit normal vector at \mathbf{Q} .

Proof. At the shadow boundary, the light ray $r(t)$ just grazes the surface \mathcal{S} at \mathbf{Q} . This means the vector from the light source to \mathbf{Q} , $\mathbf{Q} - \mathbf{L}$, lies in the tangent plane of \mathcal{S} at \mathbf{Q} . Since the tangent plane is orthogonal to the surface normal $\mathbf{n}(\mathbf{Q})$, their dot product is zero: $(\mathbf{Q} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{Q}) = 0$ \square

Definition 3.2.1 (Shadow Boundary Set).

The 3D shadow boundary \mathcal{B} consists of all points on \mathcal{S} where the light direction is tangent:

$$\boxed{\mathcal{B} = \{\mathbf{Q} \in \mathcal{S} \mid (\mathbf{Q} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{Q}) = 0\}} \quad (9)$$

Proposition 3.2.1 (Surface Illumination Partition).

The surface \mathcal{S} is divided into three disjoint regions based on the sign of the dot product between the light direction and surface normal:

$$\mathcal{S}_{\text{lit}} = \{\mathbf{P} \in \mathcal{S} \mid (\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P}) > 0\} \quad (\text{illuminated region}) \quad (10)$$

$$\mathcal{S}_{\text{shadow}} = \{\mathbf{P} \in \mathcal{S} \mid (\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P}) < 0\} \quad (\text{attached shadow region}) \quad (11)$$

$$\mathcal{B} = \{\mathbf{P} \in \mathcal{S} \mid (\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P}) = 0\} \quad (\text{shadow boundary}) \quad (12)$$

Remark 3.2.1 (Geometric Interpretation).

The sign of the dot product $(\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P})$ encodes the relative angle between the surface normal and the direction to the light source. Positive values indicate the surface faces toward the light (lit), negative values indicate it faces away (shadowed), and zero indicates a tangent point forming the shadow boundary.

4 ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision

5 Experimental Demonstration

6 Discussion and Analysis

7 Conclusion

References

- [LWX23] Jingwang Ling, Zhibo Wang, Feng Xu. *ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision*. arXiv: [2211.14086](#), 2023.
- [VK23] Venkatkumar. *3D Reconstruction Basic Terminology (Traditional Computer Vision Approach)*. Medium: [URL](#), 2023.
- [Figure 1.] Sjoerd van Riel. *Exploring the use of 3D GIS as an analytical tool in archaeological excavation practice*. ResearchGate: [URL](#), 2016.
- [Figure 2.] Catree. [Camera coordinate to pixel coordinate - OpenCV](#), 2016.
- [Figure 3.] Lemanoosh. [Blender material rendering](#), n.d.
- [Figure 4.] Michael Abramov. [Semantic Segmentation vs Object Detection: Understanding the Differences](#), 2024.
- [Figure 5.] Sandip Neogi. [shadow of object Pro Vector](#), n.d.