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
July 5, 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 Background

1.1 What is 3D Reconstruction from Images?

The goal of 3D reconstruction is to recover the structure of a 3D scene using only 2D images.

Definition 1.1 (3D Scene Representation)**.**
A 3D scene is represented by a set of points $\boldsymbol{P} = [P_x, P_y, P_z]^\top \in \mathbb{R}^3$ in Euclidean space.

Definition 1.2 (Image Projection)**.**
Each image I_n of the 3D scene records a set of pixel coordinates $\boldsymbol{p} = [p_x, p_y]^\top \in \mathbb{R}^2$.

The process of capturing a 3D point in a 2D image I_n can be modeled as a projection function π_n:

$$\boxed{\pi_n : \mathbb{R}^3 \to \mathbb{R}^2, \quad [P_x, P_y, P_z]^\top \mapsto [p_x, p_y]^\top} \tag{1}$$

This projection function represents how a camera maps a 3D point to a 2D pixel in the n-th image. To reconstruct the 3D scene, we need to solve the **inverse problem** π_n^{-1}:

$$\boxed{\pi_n^{-1}(\boldsymbol{p}) = \left\{\boldsymbol{P} \in \mathbb{R}^3 \mid \pi_n(\boldsymbol{P}) = \boldsymbol{p}\right\}} \tag{2}$$

However, this inverse problem is typically **ill-posed**, as multiple 3D points may project to the same 2D pixel, leading to ambiguity. We will detail this in Section 1.4.

1.2 Information Encoded in 2D Images

A 2D image I_n can provide multiple types of information encoded as mathematical structures:

Information Available from an Image:

(i) **Pixel coordinates**: $\boldsymbol{p} = [p_x, p_y]^\top \in \mathbb{R}^2$, represents the spatial location of each pixel

(ii) **Color values**: $C_n(\boldsymbol{p}) = [r, g, b]^\top \in [0, 1]^3$, represents the RGB tristimulus values

(iii) **RGB gradient matrix**:

$$\nabla I_n(\boldsymbol{p}) = \begin{bmatrix} \frac{\partial r}{\partial p_x} & \frac{\partial r}{\partial p_y} \\ \frac{\partial g}{\partial p_x} & \frac{\partial g}{\partial p_y} \\ \frac{\partial b}{\partial p_x} & \frac{\partial b}{\partial p_y} \end{bmatrix} \in \mathbb{R}^{3\times 2} \tag{3}$$

This Jacobian matrix captures local intensity variations, indicating edges or texture information.

(iv) **Learned feature embedding**: $\phi(I_n)(\boldsymbol{p}) \in \mathbb{R}^d$, represents high-dimensional features extracted via neural networks

These data structures result from projecting 3D geometry through camera optics, where $C_n(\boldsymbol{p})$ corresponds to visible surface reflectance and $\nabla I_n(\boldsymbol{p})$ encodes geometric boundaries.

1.3 The Forward Projection: From 3D World to 2D Image

We formalize the perspective projection process using homogeneous coordinates and transformation matrices.

Camera Parameter Matrices:

Definition 1.3 (Extrinsic Parameters)**.**
The world-to-camera transformation is characterized by:

$$\text{Camera center:} \quad \boldsymbol{C} = [C_x, C_y, C_z]^\top \in \mathbb{R}^3 \tag{4}$$

$$\text{Rotation matrix:} \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathrm{SO}(3) \tag{5}$$

$$\text{Translation vector:} \quad \boldsymbol{t} = -R\boldsymbol{C} \in \mathbb{R}^3 \tag{6}$$

Definition 1.4 (Intrinsic Parameters)**.**
The camera's internal geometry is encoded by:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{7}$$

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

Forward Projection Pipeline:

Proposition 1.1 (Perspective Projection Transform)**.**
The complete forward projection involves three sequential transformations:

Step 1: World to camera coordinates

$$\boldsymbol{P}_{\text{cam}} = R\boldsymbol{P} + \boldsymbol{t} \tag{8}$$

Step 2: Camera to image coordinates

$$\boldsymbol{P}_{\text{hom}} = K\boldsymbol{P}_{\text{cam}} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix} \tag{9}$$

Step 3: Perspective division

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix}, \quad z' \neq 0 \tag{10}$$

The complete transformation matrix can be expressed as:

$$\boxed{\boldsymbol{P}_{\text{hom}} = K[R \mid \boldsymbol{t}] \begin{bmatrix} \boldsymbol{P} \\ 1 \end{bmatrix}, \quad \boldsymbol{p} = \frac{1}{z'} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix}} \tag{11}$$

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

We now tackle the fundamental challenge of inverting the projection function.

Lemma 1.1 (Camera Ray Parametrization)**.**
Given a pixel $\boldsymbol{p} = [p_x, p_y]^\top$ and camera parameters $(K, R, \boldsymbol{C})$, the corresponding 3D points form a ray:

$$\boxed{\boldsymbol{P}(\lambda) = \boldsymbol{C} + \lambda \cdot \boldsymbol{d}, \quad \lambda > 0} \tag{12}$$

where the ray direction is:

$$\boxed{\boldsymbol{d} = R^{-1}K^{-1}\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}} \tag{13}$$

Remark 1.1 (Normalization)**.**
The direction vector $\boldsymbol{d}$ can optionally be normalized to unit length for physical ray tracing but not strictly necessary for the ray parametrization.

Proof. Starting from the forward projection equation (11):

$$K(R\boldsymbol{P} + \boldsymbol{t}) = z' \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \tag{14}$$

$$R\boldsymbol{P} + \boldsymbol{t} = z'K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \tag{15}$$

$$\boldsymbol{P} = R^{-1}\left(z'K^{-1}\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} - \boldsymbol{t}\right) \tag{16}$$

Since $\boldsymbol{t} = -R\boldsymbol{C}$, we have $-R^{-1}\boldsymbol{t} = \boldsymbol{C}$. Setting $\lambda = z'$:

$$\boldsymbol{P}(\lambda) = \boldsymbol{C} + \lambda \cdot R^{-1}K^{-1}\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \tag{17}$$

□

Proposition 1.2 (Ill-posed Nature of Single-View Reconstruction)**.**
The inverse projection problem is fundamentally **ill-posed** because:

(a) The depth parameter λ is undetermined

(b) Each pixel $\boldsymbol{p}$ defines a ray of infinitely many possible 3D points

(c) Additional constraints are required for unique reconstruction

Remark 1.2 (Toward Well-posed Reconstruction)**.**
To achieve unique reconstruction, we require additional information such as:

- **Stereo correspondence**: Multiple viewpoints providing triangulation
- **Depth sensors**: Direct measurement of λ
- **Shadow constraints**: Geometric relationships via light ray intersections

2 The ShadowNeuS Framework

2.1 Motivation for Shadow-based Learning

2.2 Signed Distance Fields (SDFs)

2.3 Shadow Rays and Surface Visibility

2.4 Learning via Shadow Supervision

2.5 Optimization Process

2.6 Geometric Interpretation and Classical Analogs

References

[LWX23] Jingwang Ling, Zhibo Wang, Feng Xu. *ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision*. arXiv: 2211.14086, 2023.