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 4, 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. Analysis of the training process reveals deep connections to projective geometry, spatial reasoning in $\mathbb{R}^3$, and the network's perception of three-dimensional 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. Consider a 3D scene represented by a set of points $P \in \mathbb{R}^3$, having coordinates $P(P_x, P_y, P_z)$. For each image taken for the 3D scene contains a set of pixel points $p \in \mathbb{R}^2$, having coordinates $p(p_x, p_y)$. The process of capturing a 3D point in a 2D image I_n can be modeled as a projection function π_n

$$\pi_n : \mathbb{R}^3 \to \mathbb{R}^2, \quad (P_x, P_y, P_z) \mapsto (p_x, p_y)$$

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

$$\pi_n^{-1}(p) \to P$$

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 in Section 1.4.

1.2 Information Encoded in 2D Images

A 2D image $I_n : \mathbb{R}^2 \to [0,1]^3$ can provide multiple information, such as color and texture. The information available from an image includes:

- **Pixel coordinates**: $p(p_x, p_y) \in \mathbb{R}^2$, represent the location of each pixel in the image
- **Color values**: $I_n(p) = [r, g, b] \in [0,1]^3$, represent the RGB color of each pixel in the image
- **Image gradient**:

$$\nabla I_n(p) = (\nabla r(p), \nabla g(p), \nabla b(p)) = \left(\underbrace{\left[\frac{\partial r}{\partial p_x}, \frac{\partial r}{\partial p_y}\right]^\top}_{\text{red channel}}, \underbrace{\left[\frac{\partial g}{\partial p_x}, \frac{\partial g}{\partial p_y}\right]^\top}_{\text{green channel}}, \underbrace{\left[\frac{\partial b}{\partial p_x}, \frac{\partial b}{\partial p_y}\right]^\top}_{\text{blue channel}} \right) \in \mathbb{R}^6$$

  It captures local changes in intensity, indicating edges or texture information in the image

- **Learned features**: $\phi(I_n)(p) \in \mathbb{R}^d$, represent a high-dimensional features extracted from the image using methods like convolutional neural networks (CNNs) or other feature extracters

These data result from projecting 3D structures through a camera. For example, the color $I_n(p)$ may correspond to the visible surface of a 3D object, while $\nabla I_n(p)$ may hint the edge of the shape of that 3D object, etc.

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

We formalize the perspective projection process that projects a 3D point $P = [P_x, P_y, P_z]^T \in \mathbb{R}^3$ to a pixel point $p = [p_x, p_y]^T \in \mathbb{R}^2$ using the camera parameters.

Camera parameters:

- **Extrinsic** (world-to-camera transformation)
  - **Camera center**: $C(C_x, C_y, C_z) \in \mathbb{R}^3$, represent the position of the camera in the world coordinate.
  - **Rotation matrix**: $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in SO(3)$, represent a 3x3 matrix that rotate the world to align with the orientation of the camera.
  - **Translation vector**: $t = -RC \in \mathbb{R}^3$, represent the translate to set the camera center be the world origin.
- **Intrinsic** (projection to image plane)
  - **Intrinsic matrix**:

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

    where f_x, f_y are the focal lengths in (pixels) and c_x, c_y is the principal point (the pixel coordinates where the camera's lens is optically centered)

Forward Projection Pipeline

$$P_{hom} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix} = K[R|t] \begin{bmatrix} P \\ 1 \end{bmatrix} \quad \& \quad \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix}, \qquad z' \neq 0$$

P_{hom} uses homogeneous coordinates that an extra coordinates is added to handle scaling. The process involves:

1. **World to camera coordinate**: Transform $P(P_x, P_y, P_z)$ to camera coordinate

$$P \to \underbrace{(P - C)}_{\text{set cemera to origin}} \to \underbrace{R(P - C)}_{\text{rotate the world for alignment}} = \underbrace{RP - RC = RP + t}_{\text{simplify}} = \underbrace{[R|t] \begin{bmatrix} P \\ 1 \end{bmatrix}}_{\text{matrix operation}}$$

2. **Perspective projection**: $P_{hom} = K[R|t] \begin{bmatrix} P \\ 1 \end{bmatrix} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix}$

3. **Normalization**: Convert to 2D pixel coordinates by the scaling fator z'

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

References

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