# 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 7, 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 $\boldsymbol{P} = [P_x, P_y, P_z]^\mathsf{T} \in \mathbb{R}^3$ in Euclidean space.

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

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

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

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** $\pi_n^{-1}$:

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

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**: $\boldsymbol{p} = [p_x, p_y]^\mathsf{T} \in \mathbb{R}^2$, represents the spatial location of each pixel

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

(iii) **RGB gradient matrix**:

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

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 like CNNs

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

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

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

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

**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 $\boldsymbol{p} = [p_x, p_y]^\mathsf{T}$ 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 $\lambda$ is undetermined

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

(c) Additional constraints are required for unique reconstruction

## 1.5 Cues for Solving the Inverse Problem

To achieve unique reconstruction, we require additional information such as:

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

# 2 Shadows as a Geometric Constraint

We now introduce how shadows, often considered a nuisance in image understanding, can instead be leveraged as powerful geometric constraints. By formalizing light transport and occlusion, we derive conditions that allow recovery of 3D structure from single images.

## 2.1 Light Ray and Shadow Geometry

**Definition 2.1** (Light Ray)**.**
Given a light source $\boldsymbol{L} \in \mathbb{R}^3$ and a point $\boldsymbol{P} \in \mathbb{R}^3$, the light ray from $\boldsymbol{L}$ to $\boldsymbol{P}$ is the segment:

$$\boxed{r(t) = \boldsymbol{L} + t(\boldsymbol{P} - \boldsymbol{L}), \quad t \in [0,1]} \tag{18}$$

**Definition 2.2** (Shadow Occlusion)**.**
A point $\boldsymbol{P}$ is in shadow if there exists some $t \in (0,1)$ such that $r(t)$ intersects a surface $\mathscr{S}$:

$$\boxed{\exists t \in (0,1) : r(t) \cap \mathscr{S} \neq \emptyset} \tag{19}$$

**Remark 2.1** (Physical Interpretation)**.**
The open interval $(0,1)$ corresponds to obstructions between the light source and the point. Intersection in this interval implies occlusion and shadowing.

## 2.2 Shadow Boundary and Surface Partitioning

**Theorem 2.1** (Tangency Condition)**.**
A point $\boldsymbol{Q} \in \mathscr{S}$ lies on the shadow boundary if and only if the light ray is tangent to the surface at that point:

$$\boxed{(\boldsymbol{Q} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{Q}) = 0} \tag{20}$$

where $\boldsymbol{n}(\boldsymbol{Q})$ is the unit surface normal at $\boldsymbol{Q}$.

**Remark 2.2.** The dot product condition expresses that the vector from the light source to the point is orthogonal to the surface normal—indicating grazing incidence.

**Proposition 2.1** (Shadow Boundary Set)**.**
The 3D shadow boundary is defined as:

$$\boxed{\mathscr{B} = \{\boldsymbol{Q} \in \mathscr{S} \mid (\boldsymbol{Q} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{Q}) = 0\}} \tag{21}$$

**Proposition 2.2** (Surface Illumination Partition)**.**
The surface $\mathscr{S}$ is partitioned into:

$$\mathscr{S}_{\text{lit}} = \{\boldsymbol{P} \in \mathscr{S} \mid (\boldsymbol{P} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{P}) > 0\} \quad \text{(illuminated)} \tag{22}$$

$$\mathscr{A} = \{\boldsymbol{P} \in \mathscr{S} \mid (\boldsymbol{P} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{P}) < 0\} \quad \text{(attached shadow)} \tag{23}$$

$$\mathscr{B} = \{\boldsymbol{P} \in \mathscr{S} \mid (\boldsymbol{P} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{P}) = 0\} \quad \text{(shadow boundary)} \tag{24}$$

**Remark 2.3.** This partition reflects the angular relationship between the surface normal and the light direction, encoding geometric visibility information.

## 2.3 Cast Shadows on Secondary Surfaces

**Definition 2.3** (Cast Shadow Region)**.**
Given an occluding surface $\mathscr{S}_1$ and a receiving surface $\mathscr{S}_2$, the cast shadow region is:

$$\boxed{\mathscr{C}_{1\to 2} = \{\boldsymbol{P} \in \mathscr{S}_2 \mid \exists t \in (0,1) \text{ such that } \boldsymbol{L} + t(\boldsymbol{P} - \boldsymbol{L}) \in \mathscr{S}_1\}} \tag{25}$$

**Definition 2.4** (Cast Shadow Boundary)**.**
The boundary of the cast shadow on $\mathscr{S}_2$ is given by:

$$\boxed{\partial\mathscr{C}_{1\to 2} = \{\boldsymbol{P} = \boldsymbol{Q} + s(\boldsymbol{Q} - \boldsymbol{L}) \mid \boldsymbol{Q} \in \mathscr{B}_1, s > 0\}} \tag{26}$$

where $\mathscr{B}_1$ is the shadow boundary on the occluding surface $\mathscr{S}_1$.

**Remark 2.4.** Cast shadow boundaries are formed by rays tangent to the occluder, extending toward the receiver.

## 2.4 Shadows as Cues for 3D Reconstruction

Shadows, especially their boundaries, encode constraints that can be leveraged to resolve ambiguities in monocular depth estimation.

**Proposition 2.3** (Geometric Information Encoded in Shadows)**.**
Shadows provide four types of geometric cues:

(i) **Surface orientation**: Attached shadows satisfy $(\boldsymbol{P} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{P}) < 0$

(ii) **Boundary tangency**: Shadow boundaries satisfy $(\boldsymbol{Q} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{Q}) = 0$

(iii) **Relative depth**: Cast shadows reveal spatial relationships between objects

(iv) **Occluded structure**: Shadowed areas imply presence of obstructing geometry

**Theorem 2.2** (Single-View Depth Recovery via Shadow Constraints)**.**
Let a pixel $\boldsymbol{p}$ on the image lie on the projected shadow boundary. Its corresponding 3D point must lie on the camera ray:

$$\boldsymbol{P} = \boldsymbol{C} + \lambda \boldsymbol{d} \tag{27}$$

Imposing the tangency condition from equation (20) gives:

$$\boxed{(\boldsymbol{C} + \lambda \boldsymbol{d} - \boldsymbol{L}) \cdot \boldsymbol{n}(\boldsymbol{C} + \lambda \boldsymbol{d}) = 0} \tag{28}$$

This equation expresses that the unknown 3D point lies on both the shadow boundary and the camera ray, enforcing a tangency condition that geometrically constrains its depth. By solving this nonlinear equation, we can obtain a unique value $\lambda^*$ that determines the 3D point $\boldsymbol{P}^*$.

**Proposition 2.4** (Cast Shadow Consistency Check)**.**
To validate the reconstruction $\boldsymbol{P}^*$, extend the light ray from $\boldsymbol{P}^*$ and check whether its shadow projection matches the observed image:

$$\boxed{\pi(\boldsymbol{P}^* + s(\boldsymbol{P}^* - \boldsymbol{L})) \in \Omega^{\text{obs}}_{\text{shadow}}, \quad s > 0} \tag{29}$$

where $\Omega^{\text{obs}}_{\text{shadow}} \subset \mathbb{R}^2$ denotes the observed shadow region in the image.

**Remark 2.5** (Geometric Paradigm Shift)**.**
This framework reinterprets shadows not as photometric noise, but as reliable geometric constraints—enabling single-view 3D reconstruction without requiring texture, stereo pairs, or depth sensors.

## 2.5 Limitations of Shadow-Based Reconstruction

Shadows provide valuable geometric constraints but face challenges in certain scenarios. We suggest some key limitations below.

1. **Diffuse Lighting**: Diffuse or ambient light eliminates distinct shadow boundaries. The tangency condition $(\boldsymbol{Q}-\boldsymbol{L})\cdot\boldsymbol{n}(\boldsymbol{Q})=0$ (equation (20)) requires a point light source $\boldsymbol{L}$. Without it, the boundary set $\mathscr{B}$ is undefined, making equation (28) unsolvable.

2. **Self-Shadowing**: In complex scenes, multiple occluders $\mathscr{S}_1,\mathscr{S}_2$ create overlapping shadows. For a point $\boldsymbol{P}\in\mathscr{S}_2$, the light ray $r(t)=\boldsymbol{L}+t(\boldsymbol{P}-\boldsymbol{L})$ may have multiple intersections $t_1,t_2\in(0,1)$, leading to ambiguous cast shadow regions $\mathscr{C}_{1\to 2}$ and under-constrained depth $\lambda$ in equation (28).

3. **Unknown Light Position**: Accurate $\boldsymbol{L}$ is critical. An error $\Delta\boldsymbol{L}$ shifts the tangency condition to:

$$(\boldsymbol{Q}-(\boldsymbol{L}+\Delta\boldsymbol{L}))\cdot\boldsymbol{n}(\boldsymbol{Q})=0,$$

   yielding incorrect $\mathscr{B}$ and erroneous $\boldsymbol{P}^*=\boldsymbol{C}+\lambda^*\boldsymbol{d}$. Estimating $\boldsymbol{L}$ from a single image is ill-posed.

4. **Non-Lambertian Surfaces**: Specular surfaces distort shadow boundaries, misaligning observed $C_n(\boldsymbol{p})$ with geometric boundaries. Errors in $\nabla C_n(\boldsymbol{p})$ (equation (3)) invalidate the shadow constraint:

$$(\boldsymbol{C}+\lambda\boldsymbol{d}-\boldsymbol{L})\cdot\boldsymbol{n}(\boldsymbol{C}+\lambda\boldsymbol{d})=0.$$

5. **Computational Complexity**: Solving equation (28) requires minimizing:

$$|(\boldsymbol{C}+\lambda\boldsymbol{d}-\boldsymbol{L})\cdot\boldsymbol{n}(\boldsymbol{C}+\lambda\boldsymbol{d})|$$

   which is computationally intensive. Noisy shadow edges lead to local minima, producing incorrect $\boldsymbol{P}^*$.

**Remark 2.6.** Shadow-based reconstruction remains challenging due to the above limitations. However, exploring and extracting information encoded by shadows is highly useful for 3D reconstruction.

# 3

# References

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