

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 8, 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	2
1.1	What is 3D Reconstruction from Images?	2
1.2	Information Encoded in 2D Images	2
1.3	The Forward Projection: From 3D World to 2D Image	3
1.4	The Inverse Problem: From 2D Image to 3D World	4
1.5	Cues for Solving the Inverse Problem	5
2	Shadows as a Geometric Constraint	5
2.1	Light Ray and Shadow Geometry	5
2.2	Shadow Boundary and Surface Partitioning	5
2.3	Cast Shadow Regions	6
2.4	Shadow Regions in Images	6
2.5	Shadows as Cues for 3D Reconstruction	7

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 1.1.1 (3D Scene Representation).

A 3D scene is represented by a set of points $\mathbf{P} = [P_x, P_y, P_z]^\top \in \mathbb{R}^3$ in Euclidean space.

Definition 1.1.2 (Image Projection).

Each image I_n of the 3D scene records a set of pixel coordinates $\mathbf{p} = [p_x, p_y]^\top \in \mathbb{R}^2$.

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

$$\pi_n : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad [P_x, P_y, P_z]^\top \mapsto [p_x, p_y]^\top \quad (1)$$

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} :

$$\pi_n^{-1}(\mathbf{p}) = \{ \mathbf{P} \in \mathbb{R}^3 \mid \pi_n(\mathbf{P}) = \mathbf{p} \} \quad (2)$$

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:** $\mathbf{p} = [p_x, p_y]^\top \in \mathbb{R}^2$, represents the spatial location of each pixel
- (ii) **Color values:** $C_n(\mathbf{p}) = [r, g, b]^\top \in [0, 1]^3$, represents the RGB tristimulus values
- (iii) **RGB gradient matrix:**

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

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

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

These data structures result from projecting 3D geometry through camera optics, where $C_n(\mathbf{p})$ corresponds to visible surface reflectance and $\nabla C_n(\mathbf{p})$ 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 1.3.1 (Extrinsic Parameters).

The world-to-camera transformation is characterized by:

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

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

$$\text{Translation vector: } \mathbf{t} = -\mathbf{R}\mathbf{C} \in \mathbb{R}^3 \quad (6)$$

Definition 1.3.2 (Intrinsic Parameters).

The camera's internal geometry is encoded by:

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

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

Forward Projection Pipeline:

Proposition 1.3.1 (Perspective Projection Transform).

The complete forward projection involves three sequential transformations:

Step 1: World to camera coordinates

$$\mathbf{P}_{\text{cam}} = \mathbf{R}\mathbf{P} + \mathbf{t} \quad (8)$$

Step 2: Camera to image coordinates

$$\mathbf{P}_{\text{hom}} = \mathbf{K}\mathbf{P}_{\text{cam}} = \begin{bmatrix} p'_x \\ p'_y \\ z' \end{bmatrix} \quad (9)$$

Step 3: Perspective division

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

The complete transformation matrix can be expressed as:

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

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

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

Lemma 1.4.1 (Camera Ray Parametrization).

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

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

where the ray direction is:

$$\boxed{\mathbf{d} = R^{-1}K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}} \quad (13)$$

Remark 1.4.1 (Normalization).

The direction vector \mathbf{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):

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

$$R\mathbf{P} + \mathbf{t} = z' K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad (15)$$

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

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

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

□

Proposition 1.4.1 (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 \mathbf{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 λ
- **Shadow constraints:** Geometric relationships via light ray intersections

2 Shadows as a Geometric Constraint

We formalize how shadows encode geometric information to constrain 3D reconstruction. By analyzing light transport and occlusion, we derive conditions enabling depth recovery from single-view images, leveraging shadows as powerful geometric cues.

2.1 Light Ray and Shadow Geometry

Definition 2.1.1 (Light Ray).

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

$$r(t) = L + t(P - L), \quad t \in [0, 1] \quad (18)$$

Definition 2.1.2 (Shadow Occlusion Test).

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

$$r(t) \cap \mathcal{S} \neq \emptyset, \quad t \in (0, 1) \quad (19)$$

Remark 2.1.1 (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 L and P . This models physical light transport where occlusion by another surface causes a shadow.

2.2 Shadow Boundary and Surface Partitioning

Theorem 2.2.1 (Tangency Condition).

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

$$(Q - L) \cdot n(Q) = 0 \quad (20)$$

where $n(Q)$ is the outward unit normal at Q .

Proof. At the shadow boundary, the light ray $r(t)$ grazes the surface \mathcal{S} at Q , meaning the direction $Q - L$ lies in the tangent plane. Thus, it is perpendicular to the surface normal $n(Q)$, yielding $(Q - L) \cdot n(Q) = 0$. \square

Definition 2.2.1 (Shadow Boundary Set).

The 3D shadow boundary is:

$$\mathcal{B} = \{Q \in \mathcal{S} \mid (Q - L) \cdot n(Q) = 0\} \quad (21)$$

Proposition 2.2.1 (Surface Illumination Partition).

The surface \mathcal{S} is partitioned into three disjoint regions based on the light direction:

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

$$\mathcal{S}_{\text{shadow}} = \{\mathbf{P} \in \mathcal{S} \mid (\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P}) < 0\} \quad (\text{attached shadow}) \quad (23)$$

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

Remark 2.2.1 (Geometric Interpretation).

The sign of $(\mathbf{P} - \mathbf{L}) \cdot \mathbf{n}(\mathbf{P})$ reflects the angle between the light direction and the surface normal, determining whether a point is lit, shadowed, or on the boundary.

2.3 Cast Shadow Regions

Definition 2.3.1 (Cast Shadow Region).

For an occluding surface \mathcal{S}_1 and a receiving surface \mathcal{S}_2 , the cast shadow region is:

$$\mathcal{C}_{1 \rightarrow 2} = \{\mathbf{P} \in \mathcal{S}_2 \mid \exists t \in (0, 1) : \mathbf{L} + t(\mathbf{P} - \mathbf{L}) \in \mathcal{S}_1\} \quad (25)$$

Remark 2.3.1 (Cast Shadow Formation).

A point $\mathbf{P} \in \mathcal{S}_2$ is in the cast shadow if the light ray from \mathbf{L} to \mathbf{P} is blocked by \mathcal{S}_1 . For multiple occluders $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$, a point is shadowed if it lies in $\bigcup_{i=1}^k \mathcal{C}_{i \rightarrow \text{target}}$. Self-shadowing occurs when $\mathcal{S}_1 = \mathcal{S}_2$, as a surface may occlude itself.

Definition 2.3.2 (Cast Shadow Boundary).

The cast shadow boundary on \mathcal{S}_2 is:

$$\partial \mathcal{C}_{1 \rightarrow 2} = \{\mathbf{P} \in \mathcal{S}_2 \mid \mathbf{P} = \mathbf{Q} + s(\mathbf{Q} - \mathbf{L}), \mathbf{Q} \in \mathcal{B}_1, s > 0\} \quad (26)$$

where \mathcal{B}_1 is the shadow boundary on \mathcal{S}_1 (Equation (21)).

Proposition 2.3.1 (Multi-Surface Cast Shadows and Boundaries).

For k occluders, the total cast shadow region and its boundary on a target surface are:

$$\mathcal{C}_{\text{total}} = \bigcup_{i=1}^k \mathcal{C}_{i \rightarrow \text{target}} \quad (27)$$

$$\partial \mathcal{C}_{\text{total}} = \bigcup_{i=1}^k \partial \mathcal{C}_{i \rightarrow \text{target}} \quad (28)$$

where $\mathcal{C}_{i \rightarrow \text{target}}$ is the cast shadow region from \mathcal{S}_i and $\partial \mathcal{C}_{i \rightarrow \text{target}}$ is its boundary.

2.4 Shadow Regions in Images

Definition 2.4.1 (Image Shadow and Lit Regions).

Given the projection function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ (Equation (1)), the image I partitions into:

$$\Omega_{\text{lit}}^{\text{obs}} = \{\pi(\mathbf{P}) \mid \mathbf{P} \in \mathcal{S}_{\text{lit}}\} \quad (\text{lit region}) \quad (29)$$

$$\Omega_{\text{shadow}}^{\text{obs}} = \{\pi(\mathbf{P}) \mid \mathbf{P} \in \mathcal{S}_{\text{shadow}} \cup \mathcal{C}_{\text{total}}\} \quad (\text{shadow region}) \quad (30)$$

$$\partial \Omega_{\text{shadow}}^{\text{obs}} = \{\pi(\mathbf{P}) \mid \mathbf{P} \in \mathcal{B} \cup \partial \mathcal{C}_{\text{total}}\} \quad (\text{shadow boundary}) \quad (31)$$

where \mathcal{S}_{lit} , $\mathcal{S}_{\text{shadow}}$, \mathcal{B} , and $\mathcal{C}_{\text{total}}$ are defined in Equations (22), (23), (24), and (27).

Remark 2.4.1 (Relation between shadow region and image region).

The forward projection π maps 3D surface points to 2D image pixels. For example, a 3D point P on a lit surface projects to pixel $p = \pi(P)$ in the bright image region $\Omega_{\text{lit}}^{\text{obs}}$. Similarly, shadowed 3D points project to dark pixels in $\Omega_{\text{shadow}}^{\text{obs}}$. This creates a direct correspondence between 3D geometry and what we observe in the image.

Definition 2.4.2 (Shadow Consistency Loss).

The shadow consistency loss measures how well our reconstructed 3D geometry matches the observed shadows:

$$\mathcal{L}_{\text{shadow}} = w_1 \cdot \mathcal{L}_{\text{lit-mismatch}} + w_2 \cdot \mathcal{L}_{\text{shadow-mismatch}} + w_3 \cdot \mathcal{L}_{\text{boundary-mismatch}} \quad (32)$$

where $\mathcal{L}_{\text{lit-mismatch}}$, $\mathcal{L}_{\text{shadow-mismatch}}$, and $\mathcal{L}_{\text{boundary-mismatch}}$ penalize pixels that are misclassified as lit, shadow, or boundary regions respectively, and w_1, w_2, w_3 are weighting coefficients.

Remark 2.4.2 (Purpose of Shadow Consistency Loss).

This loss acts as a geometric constraint: if we observe a shadow at pixel p , then the 3D point $\pi^{-1}(p)$ must be occluded by some surface. The loss penalizes inconsistencies, forcing the reconstruction to respect shadow boundaries and create geometrically reasonable surfaces.

2.5 Shadows as Cues for 3D Reconstruction

Geometric Information Encoded in Shadows

The presence and structure of shadows encode four key geometric properties:

Proposition 2.5.1 (Shadow-Derived Geometric Cues).

From observed shadows, we can infer:

- (i) **Surface Orientation:** Points in attached shadows satisfy

$$(P - L) \cdot n(P) < 0$$

as in Equation (23), indicating that the surface at P is facing away from the light source.

- (ii) **Tangency Constraints at Boundaries:** For points on the shadow boundary, the incoming light direction lies in the surface's tangent plane:

$$(Q - L) \cdot n(Q) = 0$$

(Equation (20)). This geometric constraint localizes the transition between lit and shadowed regions.

- (iii) **Relative Depth from Cast Shadows:** If $P \in \mathcal{C}_{1 \rightarrow 2}$ (cast shadow region), then the occluding surface \mathcal{S}_1 must be geometrically in front of \mathcal{S}_2 along the light direction, as defined in Equation (25). This gives ordinal depth information.
- (iv) **Occlusion Structure:** The presence of a shadow implies the existence of an occluder blocking the light ray from L to P , indicating some intermediate geometry between the light and the shadowed surface point.

Remark 2.5.1 (Depth Ordering via Cast Shadows).

Cast shadows provide explicit ordering information. If a point P on surface \mathcal{S}_2 is shadowed due to surface \mathcal{S}_1 , then \mathcal{S}_1 must lie between the light source L and P along the ray. This insight can be directly encoded into optimization or inference schemes.

From Shadow Observation to Depth Estimation

When a pixel $p \in \partial\Omega_{\text{shadow}}^{\text{obs}}$ is observed to lie on the shadow boundary in the image, the corresponding 3D point P must satisfy the tangency condition in 3D space. Since P lies on the camera ray:

$$P(\lambda) = C + \lambda d$$

we can substitute this into the tangency condition to derive a nonlinear constraint on depth λ :

Theorem 2.5.1 (Single-View Depth Recovery from Shadow Boundaries).

Given a pixel $p \in \partial\Omega_{\text{shadow}}^{\text{obs}}$, the corresponding 3D point $P = C + \lambda d$ lies on the shadow boundary \mathcal{B} if and only if:

$$(C + \lambda d - L) \cdot n(C + \lambda d) = 0$$

This constraint defines a scalar equation in the unknown λ , assuming the surface normal is known or estimated.

Proof. Since $p \in \partial\Omega_{\text{shadow}}^{\text{obs}}$, it must correspond to a surface point P in \mathcal{B} or $\partial\mathcal{C}_{\text{total}}$ (Equation (31)). If $P \in \mathcal{B}$, then by the tangency condition (Equation (20)), we substitute $P = C + \lambda d$ into:

$$(P - L) \cdot n(P) = 0$$

to yield the boxed equation. □

Limitations and Computational Challenges

While shadows provide rich constraints, several technical difficulties must be addressed for practical reconstruction:

1. **Dependence on Surface Normals:** Equation (??) depends on $n(P)$, which in turn depends on the unknown surface geometry. This creates a circular dependency in the reconstruction process.
2. **Non-Differentiable Occlusion Tests:** The condition $r(t) \cap \mathcal{S} \neq \emptyset$ is inherently non-differentiable, which limits the applicability of gradient-based optimization.
3. **Nonlinear Equations:** Solving the system:

$$(P - L) \cdot n(P) = 0, \quad \pi(P) = p$$

is nonlinear and may have multiple or no valid solutions for λ .

4. **Light Source Uncertainty:** The location of L is often unknown or only coarsely estimated. Inaccuracies here affect all derived constraints, particularly the tangency condition.
5. **Shadow Detection in Images:** Extracting $\partial\Omega_{\text{shadow}}^{\text{obs}}$ from real images is sensitive to noise, soft shadow edges, and lighting variability, which may degrade the reliability of supervision signals like $\mathcal{L}_{\text{shadow}}$.
6. **Complex Multi-Object Interactions:** Overlapping cast shadows, self-occlusion, and translucent effects complicate the interpretation of image shadows, making exact modeling infeasible without approximations.

Remark 2.5.2 (Need for Differentiable and Learnable Representations).

These challenges expose the limitations of pure geometric modeling. In practice, neural representations such as signed distance fields (SDFs) and differentiable ray tracing frameworks enable continuous and learnable formulations of shadow cues. This motivates models like ShadowNeuS, which incorporate shadow consistency into a trainable optimization loop, bridging geometry and learning.

References

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