

Bilateral Normal Integration

Xu Cao¹, Hiroaki Santo¹, Boxin Shi^{2,3}, Fumio Okura¹, and Yasuyuki Matsushita¹

¹ Osaka University, Japan

{cao.xu, santo.hiroaki, okura, yasumat}@ist.osaka-u.ac.jp

² NERCVT, School of Computer Science, Peking University, China

³ Peng Cheng Laboratory, China

shiboxin@pku.edu.cn

Abstract. This paper studies the discontinuity preservation problem in recovering a surface from its surface normal map. To model discontinuities, we introduce the assumption that the surface to be recovered is *semi-smooth*, *i.e.*, the surface is one-sided differentiable (hence one-sided continuous) everywhere in the horizontal and vertical directions. Under the semi-smooth surface assumption, we propose a bilaterally weighted functional for discontinuity preserving normal integration. The key idea is to relatively weight the one-sided differentiability at each point’s two sides based on the definition of one-sided depth discontinuity. As a result, our method effectively preserves discontinuities and alleviates the under- or over-segmentation artifacts in the recovered surfaces compared to existing methods. Further, we unify the normal integration problem in the orthographic and perspective cases in a new way and show effective discontinuity preservation results in both cases¹.

Keywords: Normal integration, discontinuity preservation, semi-smooth surface, one-sided differentiability, photometric shape recovery

1 Introduction

Photometric shape recovery aims at high-fidelity three-dimensional (3D) surface reconstruction by exploiting the shading information. Representative methods include photometric stereo [29] and shape from polarization [17]. These methods typically estimates shape in the form of a surface normal map (Fig. 1(a)). To recover the 3D surface, it is needed to integrate the surface normals, which is called the normal integration problem [24]. Therefore, the normal integration plays a key role in photometric surface recovery.

Despite the importance, most normal integration methods are limited to recovering smooth surfaces. Namely, the target surface is assumed differentiable (hence continuous) everywhere [3,24,33]. However, this assumption is violated when the depths abruptly change at the occlusion boundaries, as shown in Fig. 1(b) and (c). In such a case, applying a method with the smooth surface assumption [3] yields distorted surfaces, as shown in Fig. 1(d).

¹ Source code is available at https://github.com/hoshino042/bilateral_normal_integration.

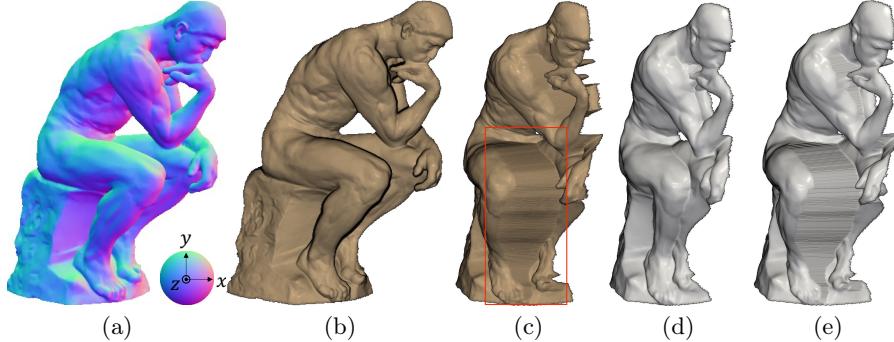


Fig. 1. The discontinuity preservation problem in normal integration. **(a)** RGB color-coded normal map rendered by a perspective camera². **(b,c)** Corresponding ground truth surface (front and side views). The red box highlights the depth gap at occlusion boundary. **(d)** Under the smooth surface assumption [3], the integrated surface is wrongly connected at occlusion boundary. **(e)** Under the semi-smooth surface assumption, our method preserves large depth gaps at occlusion boundary even in the perspective case.

Unfortunately, preserving the discontinuities in normal integration remains an open problem. So far, several assumptions on the discontinuities have been introduced to ease the problem. By assuming that discontinuities exist sparsely, robust estimators-based methods have been studied [2,7]. By assuming that discontinuity locations are short curves in the integration domain, the Mumford-Shah weighted approach has been proposed [25]. However, these methods can be fragile depending on scenes as they only statistically model the discontinuities, while the distribution of discontinuity locations of real surfaces can be arbitrary.

This paper instead introduces the *semi-smooth* surface assumption, which models the depth discontinuity directly from its definition. We assume that even if the surface is discontinuous at a point, it is discontinuous at only one side but not both sides of the point. To determine the discontinuous side for each point, we design a weight function based on the definition of one-sided depth discontinuity. Intuitively, if the depth gap at one side of a point is much larger than the other side, then the side with a larger depth gap is more likely to be discontinuous. We will show that this weight function design idea can be naturally derived from the semi-smooth surface assumption.

Under the semi-smooth surface assumption, we propose a bilaterally weighted functional for discontinuity preserving normal integration. In our functional, we approximate the normal vector observed at a point from that point's two sides in each of the horizontal and vertical directions. Using our weight functions, the unreliable approximation from the discontinuous side is ignored, while the reliable approximation from the continuous side is kept. In this way, the surface

² “The Thinker by Auguste Rodin” CC BY 4.0. <https://sketchfab.com/3d-models/the-thinker-by-auguste-rodin-08a1e693c9674a3292dec2298b09e0ae>.

can be accurately recovered without being affected by discontinuities. As our functional considers and compares the two sides of each point, we term our method as “bilateral” normal integration (BiNI).

Experimental results show that our method can faithfully locate and preserve discontinuities over the iterative optimization. Our method also reduces the number of hyperparameters that are needed in existing methods [2,25] to only one, which makes it easy to use in practice. In addition, we present a new unification of the normal integration problem in the orthographic and perspective cases, which allows us to treat the two cases in the same manner by our bilaterally weighted functional. While existing discontinuity preservation methods focus on the orthographic case [2,25,31], we confirm effective discontinuity preservation results for the first time in the perspective case, as shown in Fig. 1(e).

In summary, this paper’s contributions are

- a bilaterally weighted functional under the semi-smooth surface assumption,
- its numerical solution method, which can effectively preserve the discontinuities, and
- new unified normal integration equations that cover both orthographic and perspective cases.

2 Proposed method

The goal of normal integration is to estimate the depth or height map of a surface given its normal map and corresponding camera projection parameters. In this section, we first derive the unified partial differential equations (PDEs) relating the depth map to its normal map in the orthographic and perspective cases in Section 2.1. We then describe our semi-smooth surface assumption and present the bilaterally weighted functional in Section 2.2. Finally, we present in Section 2.3 the solution method for the proposed functional. We will discuss our method’s differences to the related work in Section 3.

2.1 Unified normal integration equations

Let $\mathbf{p} = [x, y, z]^\top \in \mathbb{R}^3$ be a surface point in a 3D space, and $\mathbf{n}(\mathbf{p}) = [n_x, n_y, n_z]^\top \in \mathcal{S}^2 \subset \mathbb{R}^3$ be the unit surface normal vector at the surface point \mathbf{p} . When the surface is observed by a camera, the surface point and its normal vector are projected in the image plane with coordinates $\mathbf{u} = [u, v]^\top \in \mathbb{R}^2$. Therefore, we can parameterize the surface and its normal map as vector-valued functions $\mathbf{p}(\mathbf{u}) = [x(\mathbf{u}), y(\mathbf{u}), z(\mathbf{u})]^\top$ and $\mathbf{n}(\mathbf{u}) = [n_x(\mathbf{u}), n_y(\mathbf{u}), n_z(\mathbf{u})]^\top$ respectively. By definition, the normal vector $\mathbf{n}(\mathbf{u})$ is orthogonal to the tangent plane to the surface at the point $\mathbf{p}(\mathbf{u})$. Hence, $\mathbf{n}(\mathbf{u})$ is orthogonal to the two tangent vectors in the tangent plane at $\mathbf{p}(\mathbf{u})$:

$$\mathbf{n}^\top \partial_u \mathbf{p} = 0 \quad \text{and} \quad \mathbf{n}^\top \partial_v \mathbf{p} = 0. \quad (1)$$

Here, ∂_u and ∂_v are partial derivatives with respect to u and v , and we omit the dependencies of \mathbf{p} and \mathbf{n} on \mathbf{u} for brevity.

We consider the normal maps observed on a closed and connected subset Ω in the image plane (*i.e.*, $\mathbf{u} \in \Omega \subset \mathbb{R}^2$) under orthographic or perspective projection. We now discuss the problem formulation in these two cases.

Orthographic case: Under orthographic projection,

$$\mathbf{p}(\mathbf{u}) = \begin{bmatrix} u \\ v \\ z(\mathbf{u}) \end{bmatrix}, \quad \partial_u \mathbf{p} = \begin{bmatrix} 1 \\ 0 \\ \partial_u z \end{bmatrix}, \quad \text{and} \quad \partial_v \mathbf{p} = \begin{bmatrix} 0 \\ 1 \\ \partial_v z \end{bmatrix}. \quad (2)$$

Inserting Eq. (2) into Eq. (1) results in a pair of PDEs

$$n_z \partial_u z + n_x = 0 \quad \text{and} \quad n_z \partial_v z + n_y = 0. \quad (3)$$

Perspective case: Let f be the camera's focal length and $[c_u, c_v]^\top$ be the coordinates of the principal point in the image plane, the surface is then $\mathbf{p}(\mathbf{u}) = z(\mathbf{u}) [(u - c_u)/f, (v - c_v)/f, 1]^\top$. The two tangent vectors are

$$\partial_u \mathbf{p} = \begin{bmatrix} \frac{1}{f}((u - c_u)\partial_u z + z) \\ \frac{1}{f}(v - c_v)\partial_u z \\ \partial_u z \end{bmatrix} \quad \text{and} \quad \partial_v \mathbf{p} = \begin{bmatrix} \frac{1}{f}(u - c_u)\partial_v z \\ \frac{1}{f}((v - c_v)\partial_v z + z) \\ \partial_v z \end{bmatrix}. \quad (4)$$

Similar to [6,7,24], we introduce a log depth map $\tilde{z}(\mathbf{u})$ satisfying $z(\mathbf{u}) = \exp(\tilde{z}(\mathbf{u}))$ to unify the formulations. By chain rule, we have

$$\partial_u z = z \partial_u \tilde{z} \quad \text{and} \quad \partial_v z = z \partial_v \tilde{z}. \quad (5)$$

Plugging Eq. (5) into Eq. (4) leads to

$$\partial_u \mathbf{p} = z \begin{bmatrix} \frac{1}{f}((u - c_u)\partial_u \tilde{z} + 1) \\ \frac{1}{f}(v - c_v)\partial_u \tilde{z} \\ \partial_u \tilde{z} \end{bmatrix} \quad \text{and} \quad \partial_v \mathbf{p} = z \begin{bmatrix} \frac{1}{f}(u - c_u)\partial_v \tilde{z} \\ \frac{1}{f}((v - c_v)\partial_v \tilde{z} + 1) \\ \partial_v \tilde{z} \end{bmatrix}. \quad (6)$$

Further plugging Eq. (6) into Eq. (1) cancels out z . Rearranging the remaining terms yields

$$\begin{cases} (n_x(u - c_u) + n_y(v - c_v) + n_z f) \partial_u \tilde{z} + n_x = 0 \\ (n_x(u - c_u) + n_y(v - c_v) + n_z f) \partial_v \tilde{z} + n_y = 0 \end{cases}. \quad (7)$$

Denoting $\tilde{n}_z = n_x(u - c_u) + n_y(v - c_v) + n_z f$ simplifies Eq. (7) as

$$\tilde{n}_z \partial_u \tilde{z} + n_x = 0 \quad \text{and} \quad \tilde{n}_z \partial_v \tilde{z} + n_y = 0, \quad (8)$$

which are in the same form as the orthographic counterpart Eq. (3). We can pre-compute \tilde{n}_z from the normal map and camera parameters. Once the log depth map \tilde{z} is estimated, we can exponentiate it to obtain the depth map z .

We have unified the PDEs in the orthographic and perspective cases as Eqs. (3) and (8). In Sections 2.2 and 2.3, we will not distinguish between orthographic and perspective cases.

2.2 Bilaterally weighted functional

We now describe our bilaterally weighted functional for discontinuity preserving normal integration. The proposed functional can be applied to either Eq. (3) or (8) depending on the camera projection model. Without loss of generality, we will use the notations in Eq. (3) hereafter. To pave the way for the bilaterally weighted functional under the semi-smooth surface assumption, we first discuss the traditional quadratic functional under the smooth surface assumption.

When assuming a smooth surface, the target surface is differentiable everywhere, *i.e.*, the partial derivatives $\partial_u z$ and $\partial_v z$ exist everywhere. We can therefore minimize the quadratic functional to find the depth map

$$\min_z \iint_{\Omega} (n_z \partial_u z + n_x)^2 + (n_z \partial_v z + n_y)^2 \, du \, dv. \quad (9)$$

When a function is differentiable at a point, it is also one-sided differentiable at the point's two sides. The one-sided partial derivatives hence exist at both sides of the point horizontally ($\partial_u^+ z$ and $\partial_u^- z$) and vertically ($\partial_v^+ z$ and $\partial_v^- z$), and are equal to the partial derivative, *i.e.*, $\partial_u^+ z = \partial_u^- z = \partial_u z$ and $\partial_v^+ z = \partial_v^- z = \partial_v z$. Therefore, the quadratic functional (9) under the smooth surface assumption is equivalent to

$$\begin{aligned} \min_z \iint_{\Omega} & 0.5(n_z \partial_u^+ z + n_x)^2 + 0.5(n_z \partial_u^- z + n_x)^2 \\ & + 0.5(n_z \partial_v^+ z + n_y)^2 + 0.5(n_z \partial_v^- z + n_y)^2 \, du \, dv, \end{aligned} \quad (10)$$

where the one-sided partial derivatives are defined as

$$\begin{aligned} \partial_u^+ z &= \lim_{h \rightarrow 0^+} \frac{z(u+h, v) - z(u, v)}{h}, & \partial_u^- z &= \lim_{h \rightarrow 0^-} \frac{z(u-h, v) - z(u, v)}{h}, \\ \partial_v^+ z &= \lim_{h \rightarrow 0^+} \frac{z(u, v+h) - z(u, v)}{h}, & \partial_v^- z &= \lim_{h \rightarrow 0^-} \frac{z(u, v-h) - z(u, v)}{h}. \end{aligned} \quad (11)$$

Considering the one-sided differentiability at a point's two sides leads to our semi-smooth surface assumption. We assume a semi-smooth surface can be indifferentiable (hence discontinuous) at and only at one side of a point in each of the horizontal and vertical directions. As illustrated in Fig. 2, this assumption contains three cases. At differentiable points, a semi-smooth surface is also guaranteed to be both left- and right-differentiable (Fig. 2 (a), both $\partial_u^+ z$ and $\partial_u^- z$ exist). Unlike a smooth surface, a semi-smooth surface can be one-sided indifferentiable at one-sided discontinuous points (Fig. 2 (b) and (c), either $\partial_u^+ z$ or $\partial_u^- z$ exists)³. On the other hand, a semi-smooth surface does not contain any point that is indifferentiable from its both sides in the horizontal or vertical direction (Fig. 2 (d) and (e), the case neither $\partial_u^+ z$ nor $\partial_u^- z$ exists is not allowed).

³ This requirement is stricter than jump discontinuity, which requires the one-sided limits exist but are unequal at a point's two sides. Fig. 2 (b), (c), and (e) are jump discontinuity examples, but a semi-smooth surface allows only Fig. 2 (b) and (c).

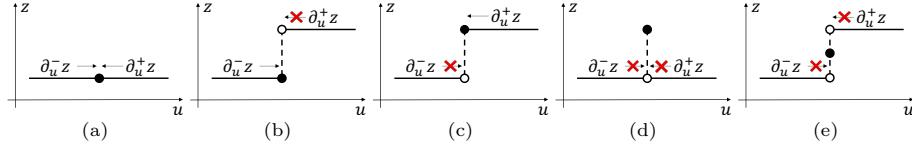


Fig. 2. A semi-smooth surface allows and only allows a point to be indifferentiable at one side. At a point, a semi-smooth surface can be (a) left and right differentiable, (b,c) only left or right differentiable, but *cannot* be (d,e) neither left nor right differentiable. Black dots indicate the function values; black circles indicate the one-sided limits; red cross indicate the non-existence of the one-sided partial derivatives.

Under the semi-smooth surface assumption, we propose the bilaterally weighted functional

$$\min_z \iint_{\Omega} w_u (n_z \partial_u^+ z + n_x)^2 + (1 - w_u)(n_z \partial_u^- z + n_x)^2 + w_v (n_z \partial_v^+ z + n_y)^2 + (1 - w_v)(n_z \partial_v^- z + n_y)^2 du dv, \quad (12)$$

where w_u and w_v indicate the one-sided differentiability at each point's two sides:

$$w_u = \begin{cases} 1 & (\text{only right diff.}) \\ 0.5 & (\text{left \& right diff.}) \\ 0 & (\text{only left diff.}) \end{cases} \text{ and } w_v = \begin{cases} 1 & (\text{only upper diff.}) \\ 0.5 & (\text{upper \& lower diff.}) \\ 0 & (\text{only lower diff.}) \end{cases}. \quad (13)$$

The bilaterally weighted functional (12) states that, for example, when the depth map is left but not right differentiable at a point, the data term is kept at the left side but ignored at the right side. When the depth map is differentiable at a point, the data terms at the two sides are equally weighted. This relative weighting thus covers all possible cases at every point in a semi-smooth surface.

Now, the problem is how to determine the surface's one-sided differentiability at each point. To this end, we use the fact that one-sided differentiability requires one-sided continuity. For example, a function being right (in)differentiable at a point must be right (dis)continuous at that point. A function being one-sided continuous at a point requires the function value at that point to be equal to the limit approached from the corresponding side. Formally, denote the differences between the function value and one-sided limits at a point as

$$\begin{aligned} \Delta_u^+ z &= z(u, v) - \lim_{h \rightarrow 0^+} z(u + h, v), & \Delta_u^- z &= z(u, v) - \lim_{h \rightarrow 0^-} z(u + h, v), \\ \Delta_v^+ z &= z(u, v) - \lim_{h \rightarrow 0^+} z(u, v + h), & \Delta_v^- z &= z(u, v) - \lim_{h \rightarrow 0^-} z(u, v + h). \end{aligned} \quad (14)$$

The function is right continuous if $\Delta_u^+ z = 0$ or right discontinuous if $\Delta_u^+ z \neq 0$; so for the left, upper, and lower continuity. To further judge which side of a point is discontinuous by one function, we can compare the one-sided continuity at the point's two sides. As detailed in Fig. 3, the surface is continuous at a point if

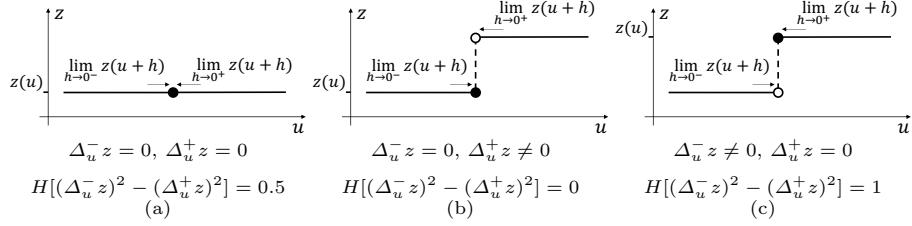


Fig. 3. Our weight function $w_u = H[(\Delta_u^- z)^2 - (\Delta_u^+ z)^2]$ compares the one-sided continuity at a point's two sides and can indicate all three cases in a semi-smooth surface. (a) $w_u = 0.5$ when left and right continuous. (b) $w_u = 0$ when left continuous and right discontinuous. (c) $w_u = 1$ when left discontinuous and right continuous.

$(\Delta_u^- z)^2 - (\Delta_u^+ z)^2 = 0$, *only* left continuous if $(\Delta_u^- z)^2 - (\Delta_u^+ z)^2 < 0$, and *only* right continuous if $(\Delta_u^- z)^2 - (\Delta_u^+ z)^2 > 0$. By wrapping this difference with a Heaviside step function $H(x) = \{1 \text{ if } x > 0; 0.5 \text{ if } x = 0; 0 \text{ if } x < 0\}$, we have our weight functions in the horizontal and vertical directions:

$$w_u = H((\Delta_u^- z)^2 - (\Delta_u^+ z)^2) \quad \text{and} \quad w_v = H((\Delta_v^- z)^2 - (\Delta_v^+ z)^2). \quad (15)$$

It can be verified that the weight functions Eq. (15) take values as Eq. (13).

2.3 Solution method

This section presents the discretization of the bilaterally weighted functional (12) and the weight functions Eq. (15) and derives a solution method given the normal map observed in the discrete pixel domain, *i.e.*, $\Omega \subset \mathbb{Z}^2$. In the following, we will not distinguish the symbols between the continuous and discrete cases.

Discretization: We first discretize the one-sided partial derivatives Eq. (11) by forward or backward differences, *i.e.*, plugging $h = 1$ or -1 into Eq. (11):

$$\begin{aligned} \partial_u^+ z &\approx z(u+1, v) - z(u, v), & \partial_u^- z &\approx z(u, v) - z(u-1, v), \\ \partial_v^+ z &\approx z(u, v+1) - z(u, v), & \partial_v^- z &\approx z(u, v) - z(u, v-1). \end{aligned} \quad (16)$$

We then approximate the one-sided limits by the depth values at adjacent pixels, and Eq. (14) becomes

$$\begin{aligned} \Delta_u^+ z &\approx n_z(z(u, v) - z(u+1, v)), & \Delta_u^- z &\approx n_z(z(u, v) - z(u-1, v)), \\ \Delta_v^+ z &\approx n_z(z(u, v) - z(u, v+1)), & \Delta_v^- z &\approx n_z(z(u, v) - z(u, v-1)). \end{aligned} \quad (17)$$

Here, the depth differences are scaled by n_z to measure the difference along the normal direction at the point. To avoid the step function always taking binary values in the discrete domain (*i.e.*, treating every pixel as one-sided discontinuous), we approximate the step function by a sigmoid function:

$$H(x) \approx \sigma_k(x) = \frac{1}{1 + e^{-kx}}, \quad (18)$$

where the parameter k controls the sharpness of the sigmoid function. Combining Eqs. (16) to (18) together, we have the discretized bilaterally weighted functional

$$\begin{aligned} \min_{z(u,v)} \sum_{\Omega} & \sigma_k ((\Delta_u^- z)^2 - (\Delta_u^+ z)^2) (n_z \partial_u^+ z + n_x)^2 \\ & + \sigma_k ((\Delta_u^+ z)^2 - (\Delta_u^- z)^2) (n_z \partial_u^- z + n_x)^2 \\ & + \sigma_k ((\Delta_v^- z)^2 - (\Delta_v^+ z)^2) (n_z \partial_v^+ z + n_y)^2 \\ & + \sigma_k ((\Delta_v^+ z)^2 - (\Delta_v^- z)^2) (n_z \partial_v^- z + n_y)^2. \end{aligned} \quad (19)$$

In (19), we use the sigmoid function's property $1 - \sigma_k(x) = \sigma_k(-x)$ to make it more compact. Intuitively, the optimization problem (19) states that if the depth difference at one side of a pixel is much larger than the other side, then the larger side is more likely to be discontinuous, and correspondingly the quadratic data term is less weighted at the discontinuous side.

Optimization: The optimization problem (19) is non-convex due to the weights being non-linear sigmoid functions of unknown depths. To solve (19), we use iteratively re-weighted least squares (IRLS) [12]. To describe the iteration process, we first prepare the matrix form of (19).

Let \mathbf{z} , \mathbf{n}_x , \mathbf{n}_y , and $\mathbf{n}_z \in \mathbb{R}^m$ be the vectors of $z(\mathbf{u})$, $n_x(\mathbf{u})$, $n_y(\mathbf{u})$, and $n_z(\mathbf{u})$ from all $|\Omega| = m$ pixels serialized in the same order. Let $\text{diag}(\mathbf{x})$ be the diagonal matrix whose i -th diagonal entry is the i -th entry of \mathbf{x} , and denote $\mathbf{N}_z = \text{diag}(\mathbf{n}_z)$. We can write the optimization problem (19) in the matrix form as

$$\min_{\mathbf{z}} (\mathbf{A}\mathbf{z} - \mathbf{b})^\top \mathbf{W}(\mathbf{z})(\mathbf{A}\mathbf{z} - \mathbf{b}) \quad (20)$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{N}_z \mathbf{D}_u^+ \\ \mathbf{N}_z \mathbf{D}_u^- \\ \mathbf{N}_z \mathbf{D}_v^+ \\ \mathbf{N}_z \mathbf{D}_v^- \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{n}_x \\ -\mathbf{n}_x \\ -\mathbf{n}_y \\ -\mathbf{n}_y \end{bmatrix}, \quad \text{and} \quad \mathbf{W}(\mathbf{z}) = \text{diag} \left(\begin{bmatrix} \mathbf{w}_u(\mathbf{z}) \\ \mathbf{1} - \mathbf{w}_u(\mathbf{z}) \\ \mathbf{w}_v(\mathbf{z}) \\ \mathbf{1} - \mathbf{w}_v(\mathbf{z}) \end{bmatrix} \right), \quad (21)$$

where

$$\begin{aligned} \mathbf{w}_u(\mathbf{z}) &= \sigma_k [(\mathbf{N}_z \mathbf{D}_u^- \mathbf{z})^{\circ 2} - (\mathbf{N}_z \mathbf{D}_u^+ \mathbf{z})^{\circ 2}], \\ \mathbf{w}_v(\mathbf{z}) &= \sigma_k [(\mathbf{N}_z \mathbf{D}_v^- \mathbf{z})^{\circ 2} - (\mathbf{N}_z \mathbf{D}_v^+ \mathbf{z})^{\circ 2}]. \end{aligned} \quad (22)$$

Here, the four matrices \mathbf{D}_u^+ , \mathbf{D}_u^- , \mathbf{D}_v^+ , and $\mathbf{D}_v^- \in \mathbb{R}^{m \times m}$ are discrete partial derivative matrices. The i -th row either contains only two non-zero entries -1 and 1 or is a zero vector if the adjacent pixel of i -th pixel is outside the domain Ω ; more details can be found in [25]. Besides, $\mathbf{1} \in \mathbb{R}^m$ is an all-one vector, $\sigma_k(\cdot)$ is now the element-wise version of the sigmoid function Eq. (18), and $(\cdot)^{\circ 2}$ is an element-wise square function on a vector.

At each step t during the optimization, we first fix the weight matrix $\mathbf{W}(\mathbf{z}^{(t)})$ and then solve for the depths \mathbf{z} :

$$\mathbf{z}^{(t+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} (\mathbf{A}\mathbf{z} - \mathbf{b})^\top \mathbf{W}(\mathbf{z}^{(t)})(\mathbf{A}\mathbf{z} - \mathbf{b}). \quad (23)$$

When $\mathbf{W}(\mathbf{z}^{(t)})$ is fixed, Eq. (23) boils down to a convex weighted least-squares problem. We can find $\mathbf{z}^{(t+1)}$ by solving the normal equation of Eq. (23)

$$\mathbf{A}^\top \mathbf{W}(\mathbf{z}^{(t)}) \mathbf{A} \mathbf{z} = \mathbf{A}^\top \mathbf{W}(\mathbf{z}^{(t)}) \mathbf{b}. \quad (24)$$

The matrix \mathbf{A} is rank 1 deficient; the 1D nullspace basis is an all-one vector, corresponding to the offset ambiguity in the result. In the perspective case, the offset ambiguity becomes scale ambiguity after exponentiating the result. In our implementation, we use a conjugate gradient method [11] to solve Eq. (24).

We initialize $\mathbf{z}^{(0)}$ as a plane, or equivalently, initialize all weights as 0.5. The estimated depth map at the first step $\mathbf{z}^{(1)}$ is then the same as the one estimated under the smooth surface assumption, *i.e.*, using the functional (10). Denoting the energy of the objective function at step t as $E_t = (\mathbf{A} \mathbf{z}^{(t)} - \mathbf{b})^\top \mathbf{W}(\mathbf{z}^{(t)}) (\mathbf{A} \mathbf{z}^{(t)} - \mathbf{b})$, we terminate the iteration once the relative energy $|E_t - E_{t-1}|/E_{t-1}$ is smaller than the user-provided tolerance or the maximum number of iterations is exceeded.

3 Related work

This section briefly reviews related works and discusses the differences between our method and existing methods.

Unified normal integration equations: Since the emergence of the normal integration problem [13,16], the majority of the methods estimates the depth map based on the PDEs

$$\partial_u z - p = 0 \quad \text{and} \quad \partial_v z - q = 0, \quad (25)$$

where $[p, q]^\top = [-\frac{n_x}{n_z}, -\frac{n_y}{n_z}]^\top$ is the gradient field computed from the normal map. By introducing the log depth map \tilde{z} , we can unify the PDEs in the perspective case as the same form as Eq. (25), with a different gradient field $[\tilde{p}, \tilde{q}]^\top = [-\frac{n_x}{\tilde{n}_z}, -\frac{n_y}{\tilde{n}_z}]^\top$ [6,7,24]. The normal integration problem is therefore also called shape/height/depth from gradient [2,8,9,10,14,19,21,26].

Equation (25) is derived from the constraint that a normal vector should be parallel to the cross product of the two tangent vectors $\mathbf{n} \parallel \partial_u z \times \partial_v z$. Alternatively, Zhu and Smith [33] derived the PDEs from the orthogonal constraint Eq. (1) and found it benefits numerical stability. However, Zhu and Smith [33] derived inconsistent PDEs in the orthographic and perspective cases. In the perspective case, Zhu and Smith [33] solves a homogeneous system using singular value decomposition, which can be more time consuming than solving an inhomogeneous system in the orthographic case.

We combine the strength of both derivations. Like Zhu and Smith [33], we derive the PDEs from the orthogonal constraint for numerical stability. Like [6,7,24], we introduce the log depth map to unify the formulations. In this way, we can solve the normal integration problem in the two cases in the same manner while being numerically more stable⁴.

⁴ See experiments in the supplementary material.

Discontinuity preserving surface recovery: We now discuss two strategies for discontinuity preservation: Robust estimator-based and weighted approaches.

As the residuals of Eq. (25) become large at discontinuous points, robust estimator-based methods apply robust functions ρ to the data terms as

$$\min_z \iint_{\Omega} \rho(\partial_u z - p) + \rho(\partial_v z - q) \, du \, dv. \quad (26)$$

Properly designed ρ -functions that can suppress the influence of large residuals are expected to preserve the discontinuities. Lorentzian function [6], total variation [23], and triple sparsity [2] have been studied.

Instead of applying robust functions, weighted approaches assign the weights to quadratic residuals of PDEs to eliminates the effects of discontinuities:

$$\min_z \iint_{\Omega} w_u (\partial_u z - p)^2 + w_v (\partial_v z - q)^2 \, du \, dv. \quad (27)$$

If the data terms at discontinuous points are appropriately assigned smaller weights, then the discontinuities are expected to be preserved.

A class of weighted approaches detects discontinuous points as a preprocessing step before optimizing (27). The major differences among these works are the clues used for discontinuity detection. Karacali and Snyder [18] detect the discontinuity based on the residuals of Eq. (25). Wu and Tang [30] use the expectation–maximization algorithm to estimate a discontinuity map from the normal map. Wang *et al.* [28] detect a binary discontinuity map using both photometric stereo images and the normal map. Xie *et al.* [31] handcraft features from the normal map. The one-time detection can be fragile as there is no scheme to correct possibly wrong detection in the optimization afterward.

A more effective type of weighted approach iteratively updates the weights. Alpha-surface method [1] first creates a minimal spanning tree from the integration domain, then iteratively adds to the spanning-tree the edges that are treated continuous. Anisotropic diffusion [1,25] applies diffusion tensors to the gradient field. Quéau *et al.* [25] design the diffusion tensors as functions of depths, and the diffusion tensors are iteratively updated during optimization. Mumford-Shah integrator [25] bypasses the detection by jointly optimizing for the weights and depths by assuming that discontinuities are short curves in the domain.

Our method can be categorized as the weighted approach. Unlike previous methods, we assume a semi-smooth target surface and *relatively weight* the one-sided differentiability at each point’s two sides. Our weights are iteratively updated during the optimization, which is different to the methods determining the weights once before the optimization [28,30,31]. Further, unlike most methods determining the weights without depth information, our weights are functions of unknown depths and thus are adaptively determined during the optimization.

4 Comparison

To verify our method’s effectiveness, this section compares our method to existing ones using synthetic and real-world normal maps in orthographic and per-

spective cases. Readers can find more experimental analysis and the discussion on the limitations of our method in the supplementary material.

4.1 Experimental settings

Baselines: We compare our method to six methods. The first one assumes smooth surfaces and uses inverse plane fitting (IPF) [3]. The remaining five methods all aim at discontinuity preservation using triple sparsity (TS) [2], total variation (TV) [25], robust estimator (RE)⁵ [25], Mumford-Shah (MS) [25], and anisotropic diffusion (AD) [25].

Implementation: We use the publicly available official implementations⁶ of IPF [3] and the four discontinuity preservation methods presented in [25]. We use the five-point version of IPF, and there is no hyperparameter. For the hyperparameters of TV, RE, MS, AD methods, we follow the suggestions in [25]: $\alpha = 0.1$ in TV, $\gamma = 0.5$ and $\beta = 0.8$ in RE, $\mu = 45$ and $\epsilon = 0.01$ in MS, and $\mu = 0.2$ and $\nu = 10$ in AD. We implement TS [2] by ourselves and following the hyperparameter setup in the paper [2]. For our method, there is one hyperparameter k in the objective function, and we set $k = 2$. The maximum iteration number and stopping tolerance of IRLS are set as 100 and 1×10^{-5} , respectively.

Metric: When the ground-truth (GT) surfaces are available, we show the absolute depth error maps and report the mean absolute depth error (MADE) between the integrated and GT depth maps. To remove the offset ambiguity in the orthographic case, we shift the integrated surfaces such that the L_1 norm between the shifted and the GT depth maps is minimal. We similarly remove the scale ambiguity by scaling the integrated surfaces in the perspective case.

4.2 Results in the orthographic case

Figure 4 shows quantitative comparisons on synthetic orthographic normal maps. The first normal map (top rows) is analytically computed, while the second one (bottom rows) is rendered from the object ‘‘Reading’’ in DiLiGenT-MV dataset [20] by the Mitsuba renderer [22]. Restricted by the smooth surface assumption, the IPF method [3] only recovers smooth surfaces as expected. When the target surface contains large depth gaps, the recovered surfaces by the IPF method [3] are heavily distorted. Compared to the IPF method [3], the TS [2] and TV method [25] marginally improve the result. The TS [2] and TV method [25] under-segments the surface and cannot faithfully locate all discontinuities. On the other hand, both the RE and MS methods [25] can identify unnecessary or incorrect discontinuity locations and thus introduce the over-segmentation artifacts into the recovered surfaces. The AD method [25] performs

⁵ The method we call robust estimator is called non-convex estimator in [25].

⁶ <https://github.com/hoshino042/NormalIntegration>
https://github.com/yqueau/normal_integration

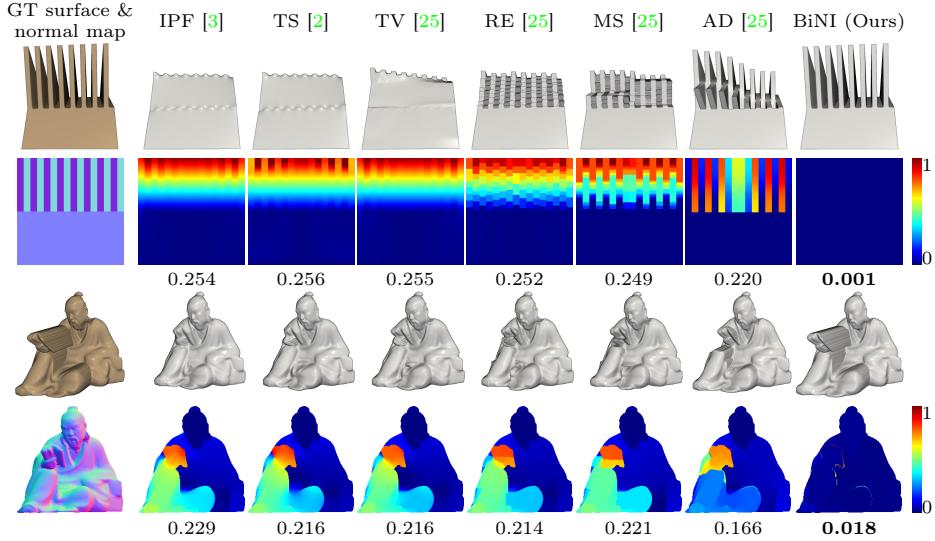


Fig. 4. Quantitative comparison using an analytically computed and a Mitsuba-rendered normal map as input. The odd and even rows display the integrated surfaces and absolute depth error maps, respectively. Numbers underneath are MADEs.

better at identifying discontinuity locations but suffers from the distortion problem. The distortion is clear in Fig. 4 (top). The ideally straight stripes recovered by the AD method [25] are still bent, although the discontinuities are well located. In contrast, our method locates discontinuity and reduces the under- or over-segmentation artifacts in the surfaces. As a result, our method achieve the smallest MADEs among all compared methods on both surfaces.

Figure 5 shows a qualitative comparison on real-world orthographic normal maps obtained in three applications. The first normal map is estimated by photometric stereo [15] on the real-world images from Light Stage Data Gallery [4]; the second one is estimated by shape from polarization [5]; and the third one is inferred from a single RGB human image by learning-based method [32]. Consistent with the trends for synthetic normal maps, the results by the baseline methods suffer from under- or over-segmentation artifacts. Our method can still preserves discontinuities reasonably well for noisy real-world normal maps. This experiment demonstrates the wide application of our method.

In addition, our method can be easier to use in practice. Compared to existing discontinuity preservation methods, our method reduce the number of hyperparameter from six [2] or two [25] to only one.

4.3 Results in the perspective case

Figure 6 shows the quantitative comparison on a Mitsuba-rendered normal map with a perspective camera model. The TV, RE, MS, and AD methods [25] all

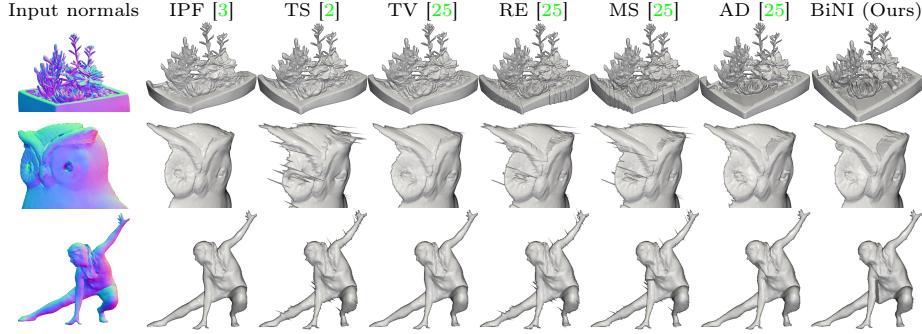


Fig. 5. Qualitative comparison of surfaces integrated from real-world noisy normal maps, which are estimated by (1st row) photometric stereo [15], (2nd row) shape from polarization [5], and (3rd row) a deep network inferred from a single RGB human image [32]. Best viewed on screen.

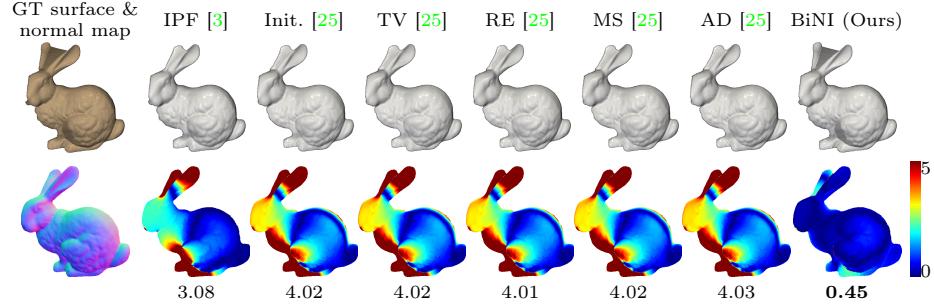


Fig. 6. Quantitative comparison using synthetic perspective normal maps rendered from Stanford Bunny. We additionally show the initialization (3rd column) for the TV, RE, MS, and AD methods [25]. The four methods do not improve the initialization and do not preserve discontinuities in the perspective case.

initialize the surface by solving the quadratic Poisson equation [25], as shown in the third column in Fig. 6. However, we can barely see the difference between the initialization and final results of the four methods. As the four methods [25] are all based on the traditional unified formulations Eq. (25), it is likely that the traditional formulation Eq. (25) is unsuitable for the discontinuity preservation methods in the perspective case. In contrast, based on our unified formulation Eq. (8), our bilaterally weighted functional still preserves discontinuities in the perspective case.

Figure 7 shows integration results from the GT perspective normal maps in DiLiGenT benchmark [27]. We again observe that the TV, RE, MS, and AD methods [25] do not improve the initialization; therefore, we only display the results from the IPF method [3] and our method. Compared to the IPF method [33], our method preserves the discontinuities and largely reduces the

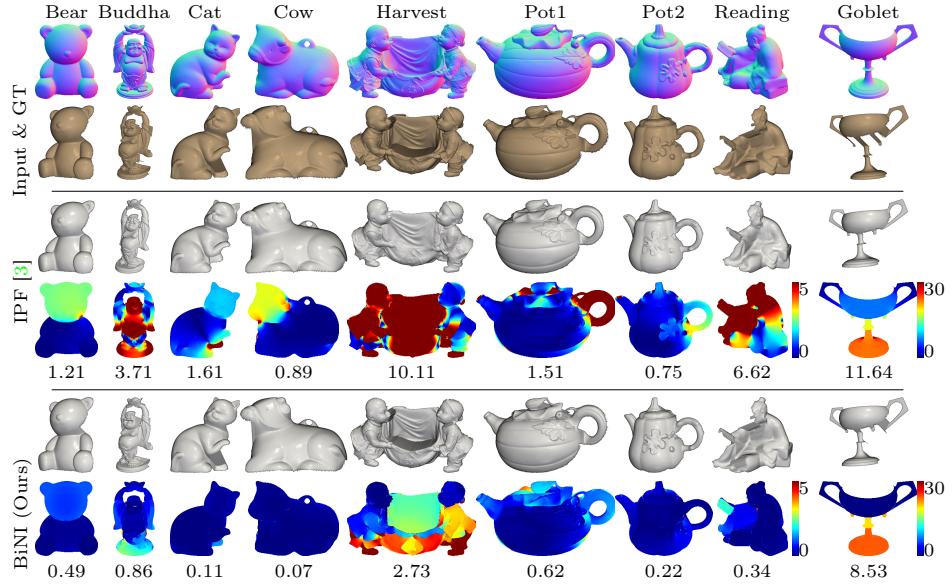


Fig. 7. Quantitative comparison on DiLiGenT benchmark [27]. **(1st & 2nd rows)** The input perspective normal maps and GT surfaces. Viewpoints of surfaces are adjusted to emphasize discontinuities. **(3rd & 4th rows)** Surfaces integrated by IPF method [3] and absolute depth error maps. The colormap scale is the same for the first eight objects. **(5th & 6th rows)** Surfaces integrated by our method and absolute depth error maps. Numbers underneath are MADEs [mm].

distortion. The MADEs are within 1 mm except for two objects (“Harvest” and “Goblet”). Especially, we achieve 0.07 mm MADE for the object “Cow.” To our knowledge, our discontinuity preservation method is the first to be effective in the perspective case.

5 Conclusions

We have presented and evaluated bilateral normal integration for discontinuity preserving normal integration. Compared to existing methods, our method preserves discontinuities more effectively and alleviates the under- or over-segmentation artifacts. The effectiveness of our method relies on the bilaterally weighted functional under the semi-smooth surface assumption. Further, we have unified the normal integration formulations in the orthographic and perspective cases appropriately. As a result, we have first shown effective discontinuity preservation results in the perspective case.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number JP19H01123, and National Natural Science Foundation of China under Grant No. 62136001, 61872012.

References

1. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: Proc. of European Conference on Computer Vision (ECCV) (2006) [10](#), [17](#)
2. Badri, H., Yahia, H., Aboutajdine, D.: Robust surface reconstruction via triple sparsity. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 2283–2290 (2014) [2](#), [3](#), [9](#), [10](#), [11](#), [12](#), [13](#)
3. Cao, X., Shi, B., Okura, F., Matsushita, Y.: Normal integration via inverse plane fitting with minimum point-to-plane distance. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 2382–2391 (2021) [1](#), [2](#), [11](#), [12](#), [13](#), [14](#)
4. Chabert, C.F., Einarsson, P., Jones, A., Lamond, B., Ma, W.C., Sylwan, S., Hawkins, T., Debevec, P.: Relighting human locomotion with flowed reflectance fields. In: Proc. of ACM SIGGRAPH (2006) [12](#)
5. Deschaintre, V., Lin, Y., Ghosh, A.: Deep polarization imaging for 3D shape and SVBRDF acquisition. In: CVPR. pp. 15567–15576 (2021) [12](#), [13](#)
6. Durou, J.D., Aujal, J.F., Courteille, F.: Integrating the normal field of a surface in the presence of discontinuities. In: Workshops of the Computer Vision and Pattern Recognition (CVPRW) (2009) [4](#), [9](#), [10](#), [19](#)
7. Durou, J.D., Courteille, F.: Integration of a normal field without boundary condition. In: Workshops of the International Conference on Computer Vision (ICCVW) (2007) [2](#), [4](#), [9](#), [19](#)
8. Harker, M., O’Leary, P.: Least squares surface reconstruction from measured gradient fields. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2008) [9](#)
9. Harker, M., O’Leary, P.: Least squares surface reconstruction from gradients: Direct algebraic methods with spectral, Tikhonov, and constrained regularization. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2011) [9](#)
10. Harker, M., O’leary, P.: Regularized reconstruction of a surface from its measured gradient field. Journal of Mathematical Imaging and Vision (2015) [9](#)
11. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards **49**(6), 409 (1952) [9](#)
12. Holland, P.W., Welsch, R.E.: Robust regression using iteratively reweighted least-squares. Communications in Statistics-theory and Methods **6**(9), 813–827 (1977) [8](#)
13. Horn, B.K., Brooks, M.J.: The variational approach to shape from shading. Computer Vision, Graphics, and Image Processing (1986) [9](#)
14. Horovitz, I., Kiryati, N.: Depth from gradient fields and control points: Bias correction in photometric stereo. Image and Vision Computing (2004) [9](#)
15. Ikehata, S.: CNN-PS: CNN-based photometric stereo for general non-convex surfaces. In: Proc. of European Conference on Computer Vision (ECCV). pp. 3–18 (2018) [12](#), [13](#)
16. Ikeuchi, K.: Constructing a depth map from images. Tech. rep., Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab (1983) [9](#)
17. Kadambi, A., Taamazyan, V., Shi, B., Raskar, R.: Polarized 3D: High-quality depth sensing with polarization cues. In: Proc. of International Conference on Computer Vision (ICCV) (2015) [1](#)
18. Karacali, B., Snyder, W.: Reconstructing discontinuous surfaces from a given gradient field using partial integrability. Computer Vision and Image Understanding (2003) [10](#)

19. Klette, R., Schluens, K.: Height data from gradient maps. In: Machine Vision Applications, Architectures, and Systems Integration V. vol. 2908, pp. 204–215. SPIE (1996) [9](#)
20. Li, M., Zhou, Z., Wu, Z., Shi, B., Diao, C., Tan, P.: Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Transactions on Image Processing* (2020) [11](#)
21. Ng, H.S., Wu, T.P., Tang, C.K.: Surface-from-gradients without discrete integrability enforcement: A gaussian kernel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **32**(11), 2085–2099 (2009) [9](#)
22. Nimier-David, M., Vicini, D., Zeltner, T., Jakob, W.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* **38**(6), 1–17 (2019) [11](#)
23. Quéau, Y., Durou, J.D.: Edge-preserving integration of a normal field: Weighted least-squares, TV and L1 approaches. In: Proc. of International Conference on Scale Space and Variational Methods in Computer Vision (2015) [10](#)
24. Quéau, Y., Durou, J.D., Aujol, J.F.: Normal integration: A survey. *Journal of Mathematical Imaging and Vision* (2018) [1](#), [4](#), [9](#), [19](#)
25. Quéau, Y., Durou, J.D., Aujol, J.F.: Variational methods for normal integration. *Journal of Mathematical Imaging and Vision* (2018) [2](#), [3](#), [8](#), [10](#), [11](#), [12](#), [13](#), [17](#)
26. Saracchini, R.F., Stolfi, J., Leitão, H.C., Atkinson, G.A., Smith, M.L.: A robust multi-scale integration method to obtain the depth from gradient maps. *Computer Vision and Image Understanding* **116**(8), 882–895 (2012) [9](#)
27. Shi, B., Mo, Z., Wu, Z., Duan, D., Yeung, S.K., Tan, P.: A benchmark dataset and evaluation for non-Lambertian and uncalibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2019) [13](#), [14](#), [18](#), [19](#)
28. Wang, Y., Bu, J., Li, N., Song, M., Tan, P.: Detecting discontinuities for surface reconstruction. In: Proc. of International Conference on Pattern Recognition (ICPR). pp. 2108–2111. IEEE (2012) [10](#)
29. Woodham, R.J.: Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In: Proc. of Image Understanding Systems and Industrial Applications I (1979) [1](#)
30. Wu, T.P., Tang, C.K.: Visible surface reconstruction from normals with discontinuity consideration. In: Proc. of Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 1793–1800. IEEE (2006) [10](#)
31. Xie, W., Wang, M., Wei, M., Jiang, J., Qin, J.: Surface reconstruction from normals: A robust DGP-based discontinuity preservation approach. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2019) [3](#), [10](#)
32. Xiu, Y., Yang, J., Tzionas, D., Black, M.J.: ICON: Implicit clothed humans obtained from normals. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 13296–13306 (June 2022) [12](#), [13](#)
33. Zhu, D., Smith, W.A.P.: Least squares surface reconstruction on arbitrary domains. In: Proc. of European Conference on Computer Vision (ECCV) (2020) [1](#), [9](#), [13](#)

Supplementary material. This supplementary material contains more experimental analysis of the proposed bilaterally weighted functional (Section A) and our unified normal integration equations (Section B), and discusses the limitations of the proposed method (Section C).

A Experimental analysis of bilaterally weighted functional

This section examines the behavior of our method with different hyperparameter setups, the convergence of IRLS, and our weight functions. We test our method using two classical toy surfaces “Vase” and “Tent” [1, 25].

Hyperparameter setup: There is one hyperparameter k in our objective function Eq. (19), controlling the sharpness of the sigmoid function. Figure 8 shows the integration results with different hyperparameters k . It can be seen that when k is small, the integrated surfaces tend to be smooth; when k is excessively large, artifacts are visible around discontinuities. This is because k controls the sensitivity to depth differences between adjacent pixels.

A smaller k leads to a smoother sigmoid function. Therefore, a pixel is easier to be treated continuous because even if the depth differences between adjacent pixels are significant, the sigmoid function still takes values close to 0.5. The extreme case is when $k = 0$, the sigmoid function takes the constant value 0.5, and our method degrades to a smooth surface recovery method.

In contrast, a larger k pushes the sigmoid function towards the step function. Therefore, a pixel is easier to be treated one-sided discontinuous because a tiny depth difference between adjacent pixels can be mapped to a value close to 0 or 1, which can cause numerical instability.

For surfaces presented in the main paper, we empirically find $k = 2$ suitable. Therefore, we recommend setting $k = 2$ initially, then slightly increasing or decreasing k depending on whether the integrated surface appears overly smooth or flawed.

Convergence: As shown in Fig. 9, the energy of our objective function monotonically decreases over iterations in these two tested surfaces. The same was true for all the surfaces presented in the paper. The discontinuities are gradually preserved as the energy steadily decreases. The iteration of Eq. (23) achieves the fixed point within 100 steps given the stopping tolerance 1×10^{-5} for the two surfaces, which means the major computation time spends on solving the inhomogeneous system of Eq. (24) dozens of times.

Figure 10 shows the energy curves of the DiLiGenT objects presented in Fig. 7. Iterations on all objects converge to the fixed points within dozens of steps. See the supplementary video for the iteration process of all tested surfaces.

Discontinuity maps: Our weight functions w_u and w_v in Eq. (15) can be viewed as horizontal and vertical discontinuity maps, respectively. Figure 11 shows the

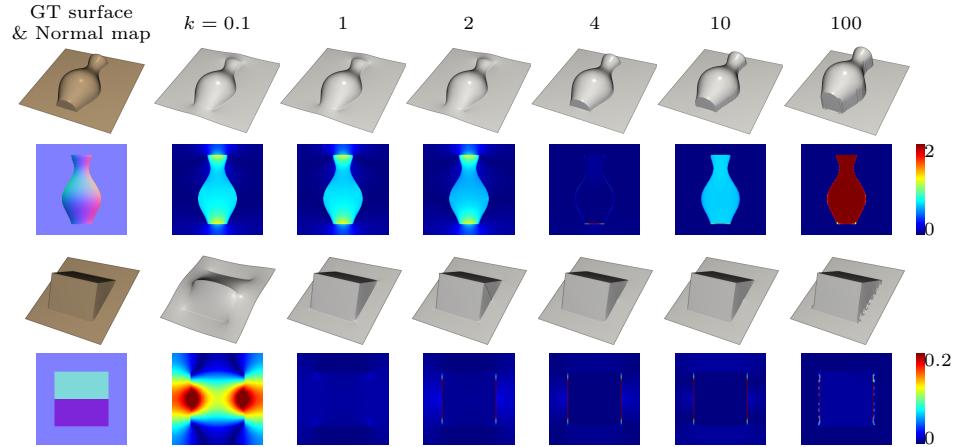


Fig. 8. The effects of the hyperparameter k , which controls the sharpness of the sigmoid function. Smaller k pushes our method toward a smooth surface recovery method, and an excessively large k introduces artifacts around discontinuities.

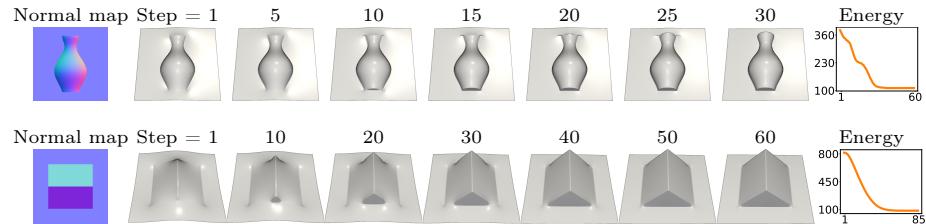


Fig. 9. Discontinuities are gradually preserved over the iteration as the energy of our objective function monotonically decreases. For figures in the last column, x -axis indicates the iteration step, and y -axis indicates the energy.

discontinuity maps computed from the integrated surface after convergence. Our weight functions accurately reflect each pixel's discontinuous side in the horizontal and vertical directions. Figure 12 shows the discontinuity maps on DiLi-GenT [27] objects, whose integrated surfaces are presented in Fig. 7. In the perspective case, our weight functions still faithfully identify the side of discontinuity.

B Comparison of unified normal integration equations

This section shows that the proposed bilaterally weighed functional is numerically more robust to outlier normal vectors and more effective in preserving discontinuities using our unified normal integration equations than the traditional unified ones.

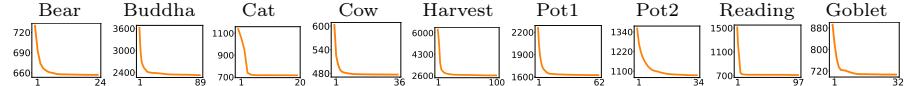


Fig. 10. Our method converges on all perspective DiLiGenT normal maps [27]. In all figures, x -axis indicates the iteration step, and y -axis indicates the energy. The integrated surfaces are visualized in Fig. 7.

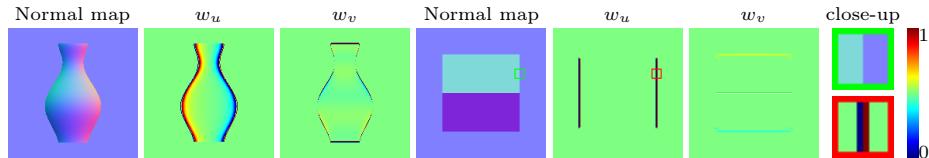


Fig. 11. Our weight functions w_u and w_v faithfully reflect the discontinuous side of each pixel in the horizontal and vertical directions, respectively. For example, in the w_u map, 1 indicates the left side is discontinuous, 0 indicates the right side is discontinuous, and 0.5 indicates both sides are continuous, as shown in the close-up images.

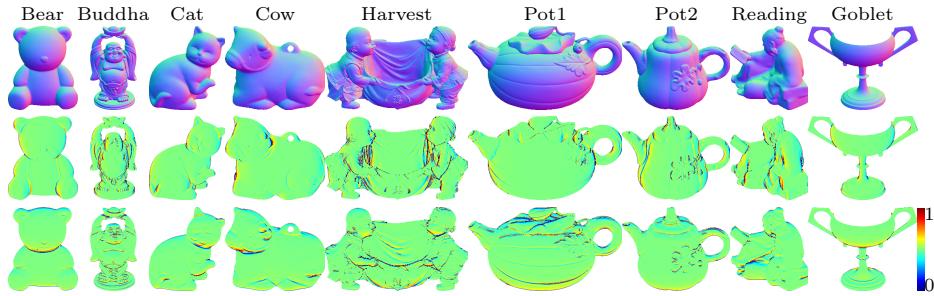


Fig. 12. Horizontal (2nd row) and vertical (3rd row) discontinuity maps by our method on DiLiGenT [27] normal maps (1st row). The integrated surfaces are visualized in Fig. 7.

In the main paper, we present our unified PDEs in the orthographic and perspective cases as

$$\begin{aligned} n_z \partial_u z + n_x = 0, \quad & \text{and} \quad n_z \partial_v z + n_y = 0 \quad (\text{Orthographic}), \\ \tilde{n}_z \partial_u \tilde{z} + n_x = 0, \quad & \text{and} \quad \tilde{n}_z \partial_v \tilde{z} + n_y = 0 \quad (\text{Perspective}), \end{aligned} \quad (28)$$

which are derived from the orthogonal constraint

$$\mathbf{n}^\top \partial_u \mathbf{p} = 0 \quad \text{and} \quad \mathbf{n}^\top \partial_v \mathbf{p} = 0. \quad (29)$$

Unlike ours, the traditional unified PDEs [6,7,24] formulate the problem as depth from gradient:

$$\begin{aligned} \partial_u z + \frac{n_x}{n_z} = 0, \quad & \text{and} \quad \partial_v z + \frac{n_y}{n_z} = 0 \quad (\text{Orthographic}), \\ \partial_u \tilde{z} + \frac{n_x}{\tilde{n}_z} = 0, \quad & \text{and} \quad \partial_v \tilde{z} + \frac{n_y}{\tilde{n}_z} = 0 \quad (\text{Perspective}), \end{aligned} \quad (30)$$

which are derived from the parallel constraint

$$\mathbf{n} \parallel \partial_u \mathbf{p} \times \partial_v \mathbf{p}. \quad (31)$$

Our unified PDEs (Eq. (28)) differ from the traditional ones (Eq. (30)) by a per-point n_z or \tilde{n}_z scale.

In the main paper, based on our unified PDEs (Eq. (28)), we present the quadratic functional under the smooth surface assumption as

$$\begin{aligned} \min_z \iint_{\Omega} & 0.5 (n_z \partial_u^+ z + n_x)^2 + 0.5 (n_z \partial_u^- z + n_x)^2 \\ & + 0.5 (n_z \partial_v^+ z + n_y)^2 + 0.5 (n_z \partial_v^- z + n_y)^2 du dv, \end{aligned} \quad (32)$$

and propose the bilaterally weighted functional under the semi-smooth surface assumption as

$$\begin{aligned} \min_z \iint_{\Omega} & w_u (n_z \partial_u^+ z + n_x)^2 + (1 - w_u) (n_z \partial_u^- z + n_x)^2 \\ & + w_v (n_z \partial_v^+ z + n_y)^2 + (1 - w_v) (n_z \partial_v^- z + n_y)^2 du dv. \end{aligned} \quad (33)$$

For the traditional unified PDEs (Eq. (30)), we can consider the quadratic functional under the smooth surface assumption as

$$\begin{aligned} \min_z \iint_{\Omega} & 0.5 \left(\partial_u^+ z + \frac{n_x}{n_z} \right)^2 + 0.5 \left(\partial_u^- z + \frac{n_x}{n_z} \right)^2 \\ & + 0.5 \left(\partial_v^+ z + \frac{n_y}{n_z} \right)^2 + 0.5 \left(\partial_v^- z + \frac{n_y}{n_z} \right)^2 du dv, \end{aligned} \quad (34)$$

or the bilaterally weighted functional under the semi-smooth surface assumption

$$\begin{aligned} \min_z \iint_{\Omega} & w_u \left(\partial_u^+ z + \frac{n_x}{n_z} \right)^2 + (1 - w_u) \left(\partial_u^- z + \frac{n_x}{n_z} \right)^2 \\ & + w_v \left(\partial_v^+ z + \frac{n_y}{n_z} \right)^2 + (1 - w_v) \left(\partial_v^- z + \frac{n_y}{n_z} \right)^2 du dv. \end{aligned} \quad (35)$$

As such, our unified PDEs leads to a different functional than the traditional one.

To demonstrate the difference, Figure 13 compares the integration results by solving Eqs. (32) to (35) from perspective normal maps containing outliers. We gradually increase the percentage of outlier normal vectors in the input normal map. To generate outliers, we replace original normal vectors at randomly chosen pixels by the vectors randomly sampled on a semi-sphere.

Both Eqs. (32) and (34) are based on the smooth surface assumption, so both methods cannot recover depth gaps as expected. However, as shown in the first block in Fig. 13, the functional Eq. (34) based on the traditional unified

PDEs (Eq. (30)) is likely to introduce spike artifacts to the integrated surface when the normal map contains outliers. On the method, as shown in the third block in Fig. 13, the functional Eq. (32) based on our unified PDEs (Eq. (28)) is numerically more robust to outlier normal vectors; we can barely see spikes in the integrated surfaces.

Our proposed bilaterally weighed functional Eq. (33) inherits this numerical robustness. As shown in the fourth block in Fig. 13, even in the existence of outlier normal vectors, our method can faithfully recover surfaces with discontinuities. On the other hand, if we apply the traditional PDEs (Eq. (30)) to our proposed bilaterally weighted functional Eq. (35), the spike artifacts are further amplified, as shown in the second block in Fig. 13.

Further, comparing the first column of Fig. 13, it can be seen that the proposed bilaterally weighted functional Eq. (33) can be more effective in preserving discontinuities using our PDEs (Eq. (28)) than the traditional ones (Eq. (30)).

To conclude, our unified PDEs (Eq. (28)) improve the numerical robustness to outlier normal vectors. Inheriting this numerical robustness, the proposed bilaterally weighted functional Eq. (33) can effectively preserve discontinuities from normal maps with outliers.

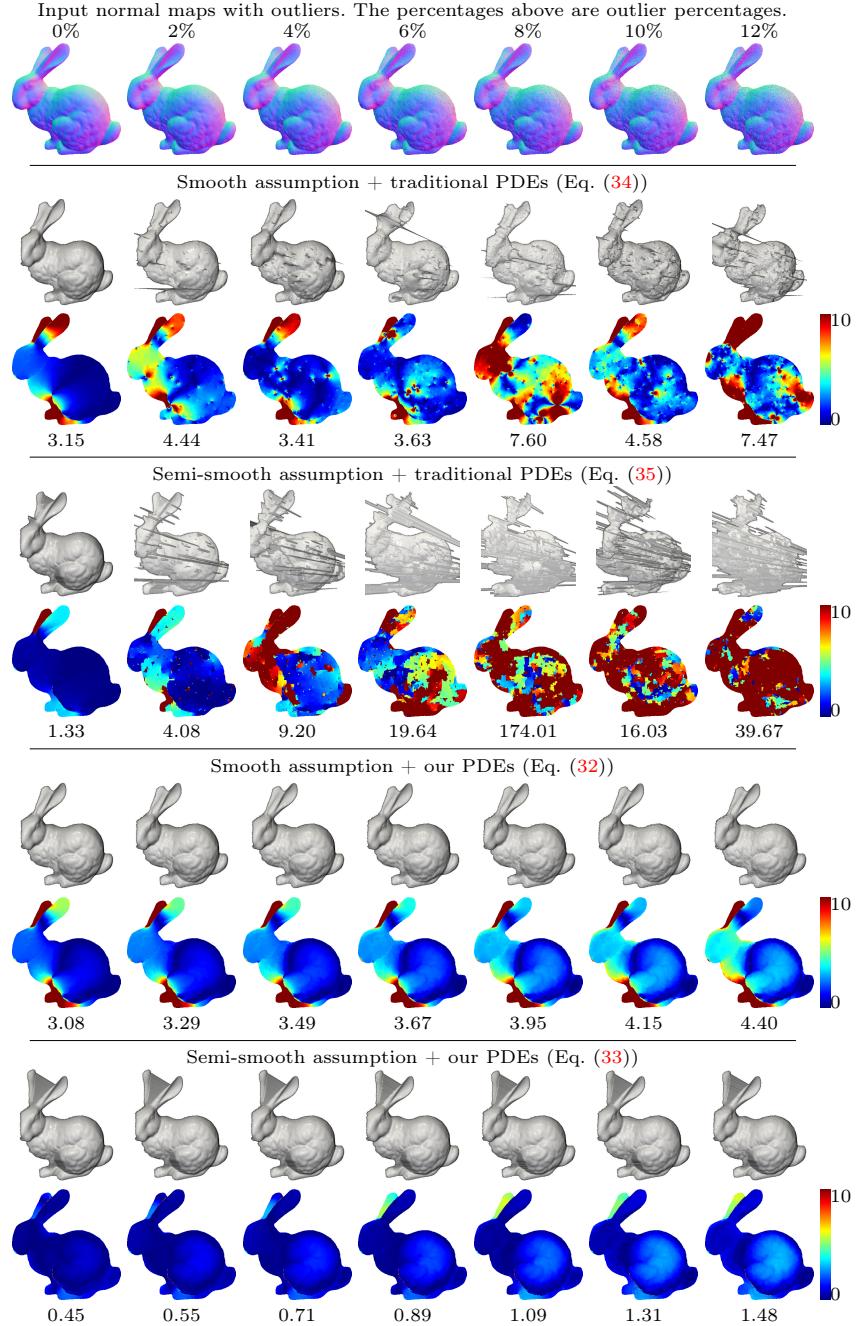


Fig. 13. Robustness comparison on perspective normal maps with increasing outliers. In each block, we show the integrated surfaces, corresponding absolute depth error maps, and MADEs. Compared to traditional unified PDEs (1st block), our unified PDEs are robust to outliers (3rd block), and our bilaterally weighted functional recovers discontinuities (4th block). As a whole, our method faithfully recovers surfaces with discontinuities from normal maps with outliers.

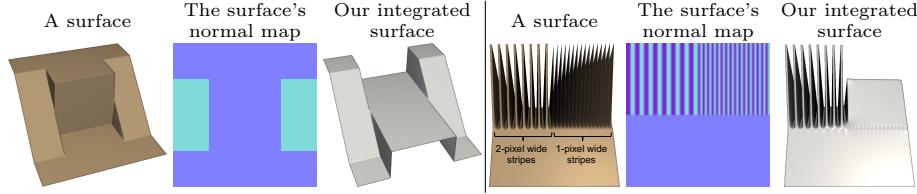


Fig. 14. Our method’s limitations. (**Left**) We cannot correctly identify the discontinuity locations when normals happen to be continuous across the depth discontinuities. (**Right**) Our method breaks down when a pixel’s both sides are discontinuous, as our semi-smooth assumption requires at least one side of each pixel, horizontally *and* vertically, to be continuous.

C Limitations

We have shown our method’s effectiveness and superiority; this section will discuss three limitations of our method.

First, when discontinuities separate the integration domain into multiple disjoint regions, multiple offset or scale ambiguities arise between disjoint regions of the surface. In principle, we cannot estimate the multiple offsets or scales given only a normal map. This phenomenon exists in the objects “Harvest” and “Goblet” in Fig. 7. Our method cannot correctly estimate the relative depths between disjoint patches, which can be confirmed from the piece-wise uniform depth error map of “Goblet.” Nevertheless, our method can be accurate up to scales between disjoint regions; the surface is barely distorted near discontinuities.

Second, our method cannot accurately identify discontinuity locations when the normals happen to be continuous across depth discontinuities. Figure 14 left shows a toy example where the normals appear planar on both sides of depth discontinuities, and the surface does not have disjoint regions. Our method fails at identifying correct discontinuity locations in the planar region of the surface. In such a case, additional information other than the normal map is required to identify the discontinuity locations.

Third, our method breaks down when a pixel’s both sides are discontinuous. Figure 14 right shows a surface consisting of multiple stripes, with stripes being two-pixel wide at the left half and one-pixel wide at the right half. Our method can recover the two-pixel wide stripes but cannot recover the one-pixel wide stripes because both sides of the one-pixel wide stripes are discontinuous.