Bilateral Normal Integration
Supplementary Material

Xu Cao[1], Hiroaki Santo[1], Boxin Shi[2,3], Fumio Okura[1], and Yasuyuki Matsushita[1]

[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

This supplementary material contains more experimental analysis of the proposed bilaterally weighted functional (Section 1) and our unified normal integration equations (Section 2), and discusses the limitations of the proposed method (Section 3).

1 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" [1,5].

Hyperparameter setup: There is one hyperparameter k in our objective function Eq. (19), controlling the sharpness of the sigmoid function. Figure 1 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.

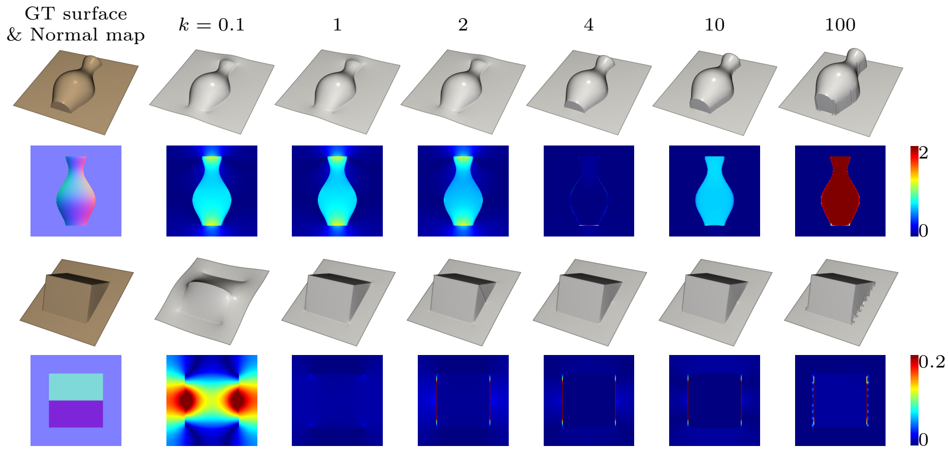

Fig. 1. The effects of the hyperparameter k, which controls the sharpness of the sigmoid function. Smaller k pushes our method toward a smooth surface recovery method, and an excessively large k introduces artifacts around discontinuities.

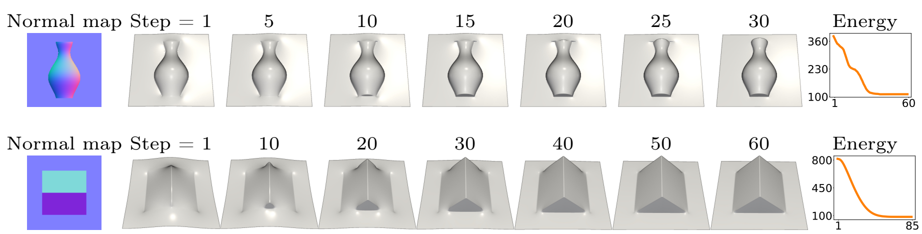

Fig. 2. Discontinuities are gradually preserved over the iteration as the energy of our objective function monotonically decreases. For figures in the last column, x-axis indicates the iteration step, and y-axis indicates the energy.

Convergence: As shown in Fig. 2, 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 3 shows the energy curves of the DiLiGenT objects presented in the main paper's 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 4 shows the discontinuity maps computed from the integrated surface after convergence. Our

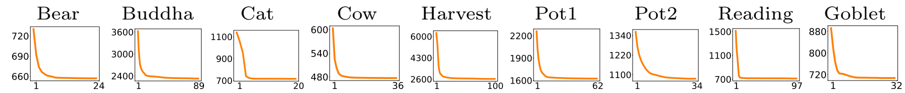

Fig. 3. Our method converges on all perspective DiLiGenT normal maps [6]. In all figures, x-axis indicates the iteration step, and y-axis indicates the energy. The integrated surfaces are visualized in the main paper's Fig. 7.

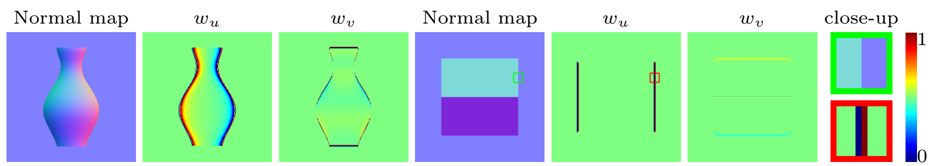

Fig. 4. Our weight functions w_u and w_v faithfully reflect the discontinuous side of each pixel in the horizontal and vertical directions, respectively. For example, in the w_u map, 1 indicates the left side is discontinuous, 0 indicates the right side is discontinuous, and 0.5 indicates both sides are continuous, as shown in the close-up images.

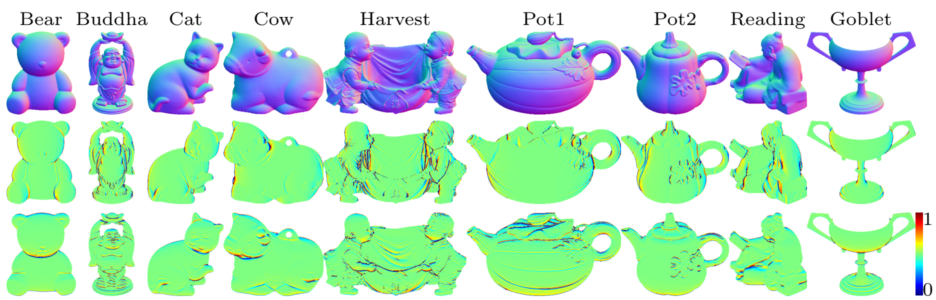

Fig. 5. Horizontal (2nd row) and vertical (3rd row) discontinuity maps by our method on DiLiGenT [6] normal maps (1st row). The integrated surfaces are visualized in the main paper's Fig. 7.

weight functions accurately reflect each pixel's discontinuous side in the horizontal and vertical directions. Figure 5 shows the discontinuity maps on DiLiGenT [6] objects, whose integrated surfaces are presented in the main paper's Fig. 7. In the perspective case, our weight functions still faithfully identify the side of discontinuity.

2 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.

In the main paper, we present our unified partial differential equations (PDEs) in the orthographic and perspective cases as

$$\begin{aligned} n_z \partial_u z + n_x = 0, \quad &\text{and} \quad n_z \partial_v z + n_y = 0 \quad \text{(Orthographic)},\\ \tilde{n}_z \partial_u \tilde{z} + n_x = 0, \quad &\text{and} \quad \tilde{n}_z \partial_v \tilde{z} + n_y = 0 \quad \text{(Perspective)}, \end{aligned} \tag{1}$$

which are derived from the orthogonal constraint

$$\mathbf{n}^\top \partial_u \mathbf{p} = 0 \quad \text{and} \quad \mathbf{n}^\top \partial_v \mathbf{p} = 0. \tag{2}$$

Unlike ours, the traditional unified PDEs [2,3,4] formulate the problem as depth from gradient:

$$\begin{aligned} \partial_u z + \frac{n_x}{n_z} = 0, \quad &\text{and} \quad \partial_v z + \frac{n_y}{n_z} = 0 \quad \text{(Orthographic)},\\ \partial_u \tilde{z} + \frac{n_x}{\tilde{n}_z} = 0, \quad &\text{and} \quad \partial_v \tilde{z} + \frac{n_y}{\tilde{n}_z} = 0 \quad \text{(Perspective)}, \end{aligned} \tag{3}$$

which are derived from the parallel constraint

$$\mathbf{n} \parallel \partial_u \mathbf{p} \times \partial_v \mathbf{p}. \tag{4}$$

Our unified PDEs (Eq. (1)) differ from the traditional ones (Eq. (3)) by a per-point n_z or $\tilde{n}_z$ scale.

In the main paper, based on our unified PDEs (Eq. (1)), we present the quadratic functional under the smooth surface assumption as

$$\begin{aligned} \min_z \iint_\Omega & 0.5 \left(n_z \partial_u^+ z + n_x\right)^2 + 0.5 \left(n_z \partial_u^- z + n_x\right)^2 \\ & +0.5 \left(n_z \partial_v^+ z + n_y\right)^2 + 0.5 \left(n_z \partial_v^- z + n_y\right)^2 \, du\, dv, \end{aligned} \tag{5}$$

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

$$\begin{aligned} \min_z \iint_\Omega & w_u \left(n_z \partial_u^+ z + n_x\right)^2 + (1 - w_u) \left(n_z \partial_u^- z + n_x\right)^2 \\ & +w_v \left(n_z \partial_v^+ z + n_y\right)^2 + (1 - w_v) \left(n_z \partial_v^- z + n_y\right)^2 \, du\, dv. \end{aligned} \tag{6}$$

For the traditional unified PDEs (Eq. (3)), we can consider the quadratic functional under the smooth surface assumption as

$$\begin{aligned} \min_z \iint_\Omega & 0.5 \left(\partial_u^+ z + \frac{n_x}{n_z}\right)^2 + 0.5 \left(\partial_u^- z + \frac{n_x}{n_z}\right)^2 \\ & +0.5 \left(\partial_v^+ z + \frac{n_y}{n_z}\right)^2 + 0.5 \left(\partial_v^- z + \frac{n_y}{n_z}\right)^2 \, du\, dv, \end{aligned} \tag{7}$$

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

$$\begin{aligned} \min_z \iint_\Omega & w_u \left(\partial_u^+ z + \frac{n_x}{n_z}\right)^2 + (1 - w_u) \left(\partial_u^- z + \frac{n_x}{n_z}\right)^2 \\ & +w_v \left(\partial_v^+ z + \frac{n_y}{n_z}\right)^2 + (1 - w_v) \left(\partial_v^- z + \frac{n_y}{n_z}\right)^2 \, du\, dv. \end{aligned} \tag{8}$$

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

To demonstrate the difference, Figure 6 compares the integration results by solving Eqs. (5) to (8) 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. (5) and (7) 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. 6, the functional Eq. (7) based on the traditional unified PDEs (Eq. (3)) 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. 6, the functional Eq. (5) based on our unified PDEs (Eq. (1)) is numerically more robust to outlier normal vectors; we can barely see spikes in the integrated surfaces.

Our proposed bilaterally weighed functional Eq. (6) inherits this numerical robustness. As shown in the fourth block in Fig. 6, 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. (3)) to our proposed bilaterally weighted functional Eq. (8), the spike artifacts are further amplified, as shown in the second block in Fig. 6.

Further, comparing the first column of Fig. 6, it can be seen that the proposed bilaterally weighted functional Eq. (6) can be more effective in preserving discontinuities using our PDEs (Eq. (1)) than the traditional ones (Eq. (3)).

To conclude, our unified PDEs (Eq. (1)) improve the numerical robustness to outlier normal vectors. Inheriting this numerical robustness, the proposed bilaterally weighted functional Eq. (6) can effectively preserve discontinuities from normal maps with outliers.

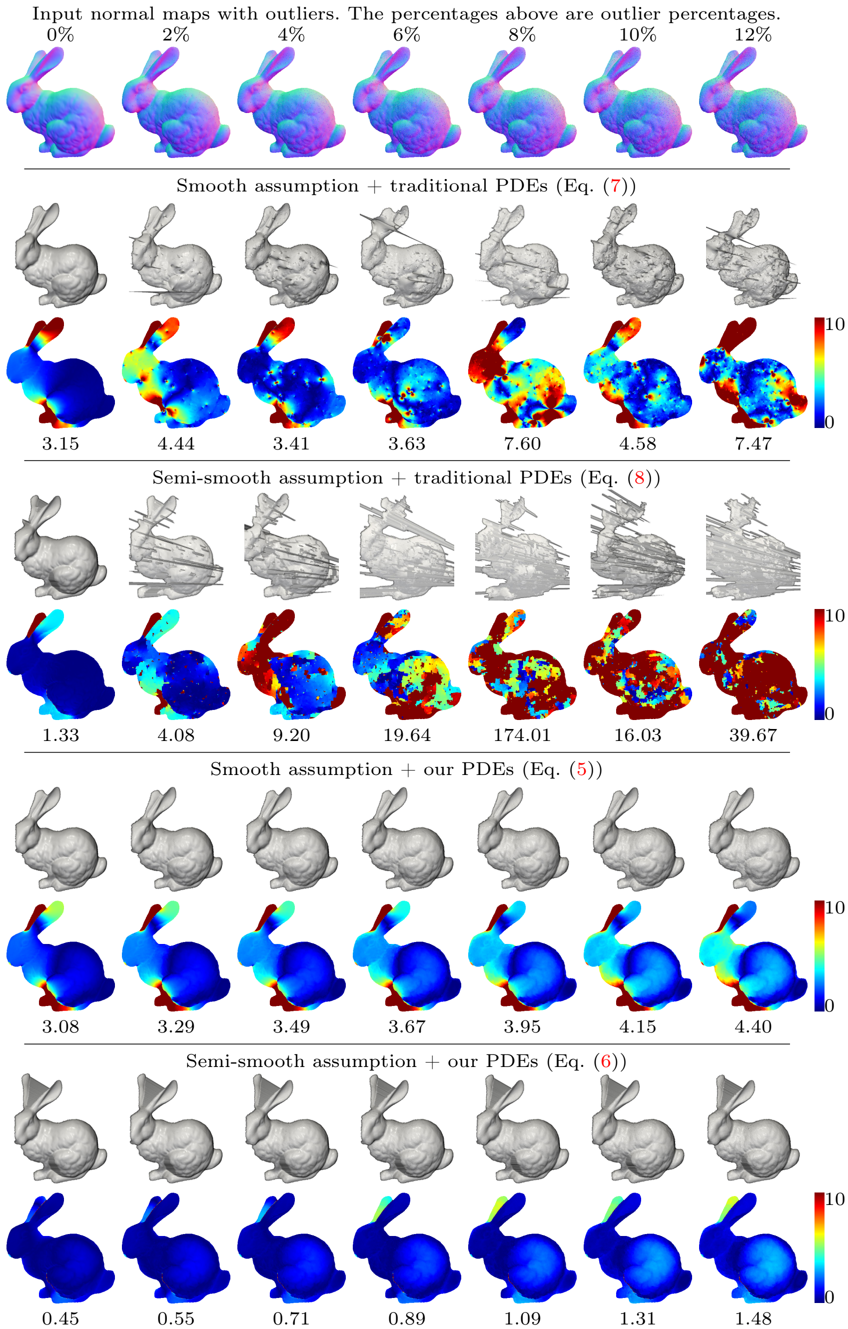

Fig. 6. Robustness comparison on perspective normal maps with increasing outliers. In each block, we show the integrated surfaces, corresponding absolute depth error maps, and MADEs. Compared to traditional unified PDEs (1st block), our unified PDEs are robust to outliers (3rd block), and our bilaterally weighted functional recovers discontinuities (4th block). As a whole, our method faithfully recovers surfaces with discontinuities from normal maps with outliers.

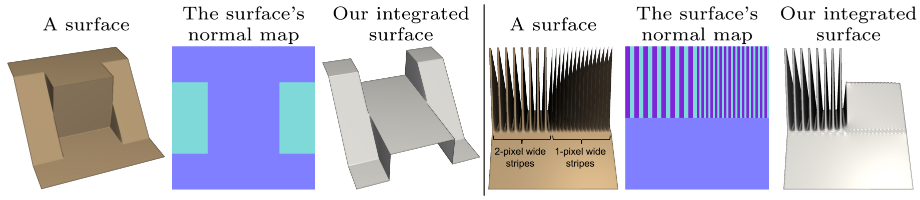

Fig. 7. Our method's limitations. **(Left)** We cannot correctly identify the discontinuity locations when normals happen to be continuous across the depth discontinuities. **(Right)** Our method breaks down when a pixel's both sides are discontinuous, as our semi-smooth assumption requires at least one side of each pixel, horizontally *and* vertically, to be continuous.

3 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 the main paper's Fig. 7. Our method cannot correctly estimate the relative depths between disjoint patches, which can be confirmed from the piecewise 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 7 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 7 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.

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) 1
2. Durou, J.D., Aujol, 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
3. Durou, J.D., Courteille, F.: Integration of a normal field without boundary condition. In: Workshops of the International Conference on Computer Vision (ICCVW) (2007) 4
4. Quéau, Y., Durou, J.D., Aujol, J.F.: Normal integration: A survey. Journal of Mathematical Imaging and Vision (2018) 4
5. Quéau, Y., Durou, J.D., Aujol, J.F.: Variational methods for normal integration. Journal of Mathematical Imaging and Vision (2018) 1
6. 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) 3