SDA-SNE: Spatial Discontinuity-Aware Surface Normal Estimation via Multi-Directional Dynamic Programming

Nan Ming, Yi Feng, Rui Fan*
MIAS Research Group, Robotics & Artificial Intelligence Laboratory, Department of Control Science & Engineering, Tongji University, Shanghai 201804, China
{nan.ming, fengyi, rui.fan}@ieee.org

Abstract

The state-of-the-art (SoTA) surface normal estimators (SNEs) generally translate depth images into surface normal maps in an end-to-end fashion. Although such SNEs have greatly minimized the trade-off between efficiency and accuracy, their performance on spatial discontinuities, e.g., edges and ridges, is still unsatisfactory. To address this issue, this paper first introduces a novel multi-directional dynamic programming strategy to adaptively determine inliers (co-planar 3D points) by minimizing a (path) smoothness energy. The depth gradients can then be refined iteratively using a novel recursive polynomial interpolation algorithm, which helps yield more reasonable surface normals. Our introduced spatial discontinuity-aware (SDA) depth gradient refinement strategy is compatible with any depth-to-normal SNEs. Our proposed SDA-SNE achieves much greater performance than all other SoTA approaches, especially near/on spatial discontinuities. We further evaluate the performance of SDA-SNE with respect to different iterations, and the results suggest that it converges fast after only a few iterations. This ensures its high efficiency in various robotics and computer vision applications requiring real-time performance. Additional experiments on the datasets with different extents of random noise further validate our SDA-SNE’s robustness and environmental adaptability. Our source code, demo video, and supplementary material are publicly available at mias.group/SDA-SNE.

1. Introduction

Surface normal is an informative 3D visual feature used in various computer vision and robotics applications, such as object recognition and scene understanding [5, 9, 13]. To date, there has not been extensive research on surface normal estimation, as it is merely considered to be an auxiliary functionality for other computer vision and robotics applications [7]. As such applications are typically required to perform robustly and in real time, surface normal estimators (SNEs) must be sufficiently accurate and computationally efficient [7].

The state-of-the-art (SoTA) SNEs [2, 4, 7, 12–14, 16, 19] generally select a set of 3D points and compute surface normals via planar fitting, geometric transformation, or statistical analysis. However, such approaches are infeasible to estimate surface normals near/on spatial discontinuities, e.g., edges and ridges (see Fig. 1), as they generally introduce adjacent 3D points on different surfaces unintentionally [10]. Bormann et al. [3] proposed the first edge-aware SNE, capable of adaptively selecting reasonable adjacent 3D points on the same surface. It performs significantly better than prior arts near edges. Nonetheless, their method focuses only on edges and requires a manually-set discontinuity awareness threshold. This results in low adaptability to different scenarios and datasets. Hence, there is a strong
motivation to develop an SNE capable of tackling all types of spatial discontinuities to achieve greater robustness and environmental adaptability.

The major contributions of this work are summarized as follows:

1) Spatial discontinuity-aware surface normal estimator (SDA-SNE), a highly accurate SNE, significantly outperforming all other SoTA SNEs, especially near/on spatial discontinuities.

2) Multi-directional dynamic programming (DP) for depth gradient refinement. It iteratively introduces inliers (co-planar pixels) by minimizing a smoothness energy.

3) Path discontinuity (PD) norm, a semi-norm evaluating the discontinuity of a given path. It reflects the depth gradient estimation error generated by the Taylor expansion.

4) Recursive polynomial interpolation (RPI), an ultrafast polynomial interpolation algorithm. Compared to Lagrange and Newton polynomial interpolation, it iteratively refines the first-order derivative of depth with a more efficient polynomial interpolation strategy.

2. Related Work

The existing SoTA SNEs can be categorized into four classes: (1) optimization-based [13], (2) averaging-based [14], (3) affine-correspondence-based [2], and (4) depth-to-normal [23].

The optimization-based SNEs, e.g., PlaneSVD [14], PlanePCA [12], VectorSVD [14], and QuadSVD [17, 22], compute surface normals by fitting local planar or curved surfaces to an observed 3D point cloud, using either singular value decomposition (SVD) or principal component analysis (PCA). The averaging-based SNEs, e.g., AreaWeighted [13] and AngleWeighted [14], estimate surface normals by computing the weighted average of the normal vectors of the triangles formed by each given 3D point and its neighbors. However, these two categories of SNEs are highly computationally intensive and unsuitable for online robotics and computer vision applications [7]. The affine-correspondence-based SNEs exploit the relationship between affinities and surface normals [2, 4, 19]. Nevertheless, such SNEs are developed specifically for stereo or multi-view cases and cannot generalize to other cases where monocular depth images are used.

Recently, enormous progress has been made in end-to-end depth-to-normal translation [6, 7, 16]. Such SNEs have demonstrated superior performance in terms of both speed and accuracy. Fan et al. [7] proposed a fast and accurate SNE referred to as 3F2N, which can infer surface normal information directly from depth or disparity images, with two gradient filters and one central tendency measurement filter (a mean or median filtering operation), as follows:

$$n_x = \frac{\partial d}{\partial u}, \quad n_y = \frac{\partial d}{\partial v},$$

$$n_z = -\Phi \left\{ \frac{(x_i - x)n_x + (y_i - y)n_y}{z_i - z}, \right\}, \quad i = 1, \ldots, m, \quad (1)$$

where $n = [n_x, n_y, n_z]^\top$ is the surface normal of a given 3D point $p^C = [x, y, z]^\top$ in the camera coordinate system, $p^C$ is projected to a 2D pixel $p = [u, v]^\top$ via $z [p^\top, 1]^\top = K p^C$ (K is the camera intrinsic matrix), $p^C_{i} = [x_i, y_i, z_i]^\top$ is one of the $m$ adjacent pixels of $p^C$, $d$ is disparity (or inverse depth), and $\Phi \{ \cdot \}$ represents the central tendency measurement filtering operation for $n_z$ estimation. Nakagawa et al. [16] presented an SNE based on cross-product of two tangent vectors (CP2TV): $n(u, v) = t_u \times t_v$, where

$$t_u = \left[ \frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right]^\top, \quad (2)$$

$$t_v = \left[ \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right]^\top,$$

$$\begin{align*}
\frac{\partial x}{\partial u} &= z + \frac{u - c_u}{f_u} \frac{\partial z}{\partial u} \\
\frac{\partial y}{\partial u} &= v + \frac{v - c_v}{f_u} \frac{\partial z}{\partial u} \\
\frac{\partial x}{\partial v} &= z + \frac{u - c_u}{f_v} \frac{\partial z}{\partial v} \\
\frac{\partial y}{\partial v} &= v + \frac{v - c_v}{f_v} \frac{\partial z}{\partial v},
\end{align*} \quad (3)$$

$e = [c_u, c_v]^\top$ is the principal point (in pixels), and $f_u$ and $f_v$ are the camera focal lengths (in pixels) in the $u$ and $v$ directions, respectively. Since the depth or disparity gradient estimation has a direct impact on the surface normal quality, this paper focuses thoroughly on the strategy to improve the accuracy of depth gradient $\nabla d = \left[ \frac{\partial z}{\partial u}, \frac{\partial z}{\partial v} \right]^\top$. The proposed strategy can also be applied to improve disparity gradient estimation, as it is inversely proportional to depth.

3. Methodology

As discussed in Sec. 2, surface normal estimation was formulated as a depth-to-normal translation problem in SoTA approaches, which have demonstrated superior performances over other methods in terms of both speed and accuracy [7]. The accuracy of such approaches is subject to the depth gradient quality. Since depth gradients typically jump near/on discontinuities, e.g., edges and ridges, the estimated surface normals near/on such discontinuities are always significantly different from their ground truth. Therefore, we propose a multi-directional DP strategy to translate

---

1In this paper, the variables with and without hat symbols denote the estimated and theoretical values, respectively.
a depth or disparity image into a surface normal map in a coarse-to-fine manner. It first initializes \( \nabla \hat{z} \) using finite difference interpolation and then optimizes \( \nabla \hat{z} \) using our proposed multi-directional DP. Accurate surface normals can then be obtained by plugging the optimum depth gradients into either the central tendency measurement filtering operation used in 3F2N (see (1)) or the cross-product operation used in CP2TV (see (2)-(3)).

### 3.1. Multi-Directional Dynamic Programming

#### 3.1.1 Smoothness Energy and State Variable

As a general rule, only inliers (co-planar pixels) should be introduced when estimating the depth gradient of an observed pixel \( p \). Such pixels are determined by minimizing the (path) smoothness energy \( E = [E_u, E_v] \) of \( p \) using DP, where \( E_u \) and \( E_v \) represent the horizontal and vertical components of \( E \), respectively. To minimize the computational complexity, we recursively introduce adjacent co-planar pixels. In each iteration, two desired pixels \( p_u' \) and \( p_v' \) (corresponding to the minimum \( E_u \) and \( E_v \), respectively) are used to extend the DP path and update the depth gradient of \( p \). We define a state variable \( S = [s_u, s_v] \), where \( s_u = p_u' - p \) and \( s_v = p_v' - p \) denote the horizontal and vertical components of \( S \), respectively. In the \( k \)-th iteration, the desired pixels are determined by minimizing \( E^{(k)} \) as follows:

\[
E^{(k)} = \min_S \left\{ T \left( E^{(k-1)}, S \right) \right\}.
\]  

The state variable \( S^{(k)} \) is transferred to

\[
S^{(k)} = \arg\min_S \left\{ T \left( E^{(k-1)}, S \right) \right\},
\]

where \( T \) denotes the smoothness energy transfer function. Therefore, we can update the depth gradient with the desired pixels using:

\[
\nabla \hat{z}^{(k)} = R \left( \nabla \hat{z}^{(k-1)}, S^{(k)} \right),
\]

where \( R \) denotes the depth gradient update function.

As shown in Fig. 2, after the initialization of depth gradient \( \nabla \hat{z} \) and DP variables \( E \) and \( S \) (detailed in Sec. 3.1.2), inliers are iteratively introduced using the smoothness energy transfer function \( T \) (detailed in Sec. 3.2) and the depth gradients are iteratively updated using the depth gradient update function \( R \) (detailed in Sec. 3.3). The optimum depth gradients can be obtained when the smoothness energy converges to the minimum. The pseudo-code of multi-directional DP is given in Algorithm 1.

#### 3.1.2 Initialization of Depth Gradient, Smoothness Energy, and State Variable

In the initialization stage, the coarse depth gradient \( \nabla \hat{z}^{(0)} \) of a given pixel is obtained by computing the finite difference (FD) between adjacent pixels. The FDs of depth are divided into \( \Delta f_z = [\Delta_{f,z_u}, \Delta_{f,z_v}]^T \) and \( \Delta v_z = [\Delta_{v,z_u}, \Delta_{v,z_v}]^T \), where the subscripts \( u \) and \( v \) represent the horizontal and vertical directions, respectively, and \( \Delta f \) and \( \Delta v \) represent forward and backward FDs, respectively. In order to
Optimum depth gradient smoothness measured by the discontinuity awareness ability, we set differential derivatives of adjacent pixels, as follows:

\[ \eta = \arg \min_{[i,j]} \{ |\hat{z}_{uu}(u+i, v)| + |\hat{z}_{vv}(u, v+j)| \}, \]

(7)

where \( i, j \in \{-1, 0, 1\} \). We can then use the smoothest pixels to linearly interpolate \( \Delta_j z \) and \( \Delta_b z \):

\[ \nabla \hat{z}^{(0)} = \frac{1}{2}(1 + \eta)^\top \circ \Delta_j z + \frac{1}{2}(1 - \eta)^\top \circ \Delta_b z, \]

(8)

where \( \circ \) denotes the Hadamard product and \( \mathbf{1} = [1, 1]^\top \). Furthermore, we initialize the smoothness energy \( E^{(0)} \) as \[ \min\{ |\hat{z}_{uu}(u+i, v)|, \min\{ |\hat{z}_{vv}(u, v+j)| \} \} \] and the state variable \( S^{(0)} \) as \( \text{diag}(\eta_u, \eta_v) \).

3.2. Smoothness Energy Transfer Function

This section discusses the formulation of the smoothness energy transfer function \( \mathcal{F} \). Energy transfer can be divided into two categories: 1) collinear transfer \( (s_u \times e_1 = 0 \text{ or } s_v \times e_2 = 0) \), where \( e_1 = [1, 0]^\top \) and \( e_2 = [0, 1]^\top \) are the unit orthogonal base of 2D Euclidean space; 2) non-collinear transfer \( (s_u \times e_1 \neq 0 \text{ or } s_v \times e_2 \neq 0) \). As a general rule, the collinear pixels should be introduced to adapt to the convex surface if they are smooth enough; otherwise, non-collinear pixels should be introduced to deal with discontinuities.

3.2.1 Depth Gradient Calculation

Based on the gradient theorem [25], the relationship among \( z, z_u \), and \( z_v \) can be written as follows:

\[ z(p') - z(p) = \int_{\mathcal{L}} z_u du + z_v dv, \]

(9)

where \( p = [u_0, v_0]^\top \) is the given pixel, \( p' = [u_1, v_1]^\top \) is its adjacent pixel to be introduced (satisfying \( |u_0 - u_1| \leq 1, |v_0 - v_1| \leq 1 \)), and \( \mathcal{L} \) is the path from \( p \) to \( p' \). We can therefore obtain (i) the collinear transfer as follows:

\[ z(p') - z(p) = \int_{\mathcal{L}_1} z_u du, \]

(10)

and (ii) the non-collinear transfer as follows:

\[ z(p') - z(p) = \int_{\mathcal{L}_1} z_u du + z_v dv = \int_{\mathcal{L}_2} z_u du + z_v dv, \]

(11)

where \( \mathcal{L}_1 \) and \( \mathcal{L}_2 \) are two different paths from \( p \) to \( p' \).

In practice, however, the discrete \( \hat{z}_u \) and \( \hat{z}_v \) can result in errors in (10) and (11). The errors of \( \hat{z}_u \) and \( \hat{z}_v \) can be estimated using the path integral of \( z_{uu} \) and \( z_{vv} \) based on the Taylor expansion (more details are provided in the supplement). Hence, in order to measure these errors in a more convenient way, we propose the PD norm, which denotes the path integral (sum) of a series of \( |z_{uu}| \) and \( |z_{vv}| \).

3.2.2 Path Discontinuity Norm

Let \( z(u, v) \) be a function in \( \mathbb{R}^2 \) defined on an open set \( \Omega \) and \( \mathcal{L} \) be a specific path contained in \( \Omega \). The discontinuity extent of the path can be measured using our proposed PD norm as follows:

\[ ||z||_{PD} = \int_{\mathcal{L}} |z_{uu} du| + |z_{vv} dv|. \]

(12)

After computing the PD norm, we can judge whether an adjacent pixel \( p' \) should be introduced. Therefore, we define the smoothness energy component as the PD norm with respect to a given path. In each iteration, we introduce two adjacent pixels which respectively minimize the smoothness energy components as follows:

\[ s_u, s_v = \arg \min_{p' \in \mathcal{N}_i(p)} \{ ||z||_{PD} - p \}, \]

\[ \mathcal{L} : \begin{cases} p \rightarrow p', & \text{if } s_u \times e_1 = 0 \text{ or } s_v \times e_2 = 0; \\ p \rightarrow p', & \text{if } s_u \times e_1 \neq 0 \text{ or } s_v \times e_2 \neq 0. \end{cases} \]

(13)
where \( \mathcal{N}_1(p) = \{ p' : \| p' - p \|_\infty \leq 1 \} \), and the smoothness energy components are minimized accordingly:

\[
E_u(p), E_o(p) = \min_{p' \in \mathcal{N}_1(p)} \{ \| \hat{z} \|_{PD} \} \nonumber
\]

\[
L : \begin{cases} p \to p' & \text{if } s_u \times e_1 = 0 \lor s_u \times e_2 = 0 \\ p \to p' \to p & \text{if } s_u \times e_1 \neq 0 \lor s_u \times e_2 \neq 0 \end{cases} \nonumber
\]

(14)

The determination of co-planar pixels is, therefore, converted into an energy minimization problem.

### 3.2.3 Energy Minimization Strategy

Since depth is discrete, we formulate the energy transfer function \( \mathcal{T} \) in (4) as follows:

\[
E_u^{(k)}(p), E_o^{(k)}(p) = \min_{p' \in \mathcal{N}_1(p)} \{ E_p^{(k-1)}(p), \nonumber
\]

\[
| \hat{z}_{pp}(p) | + E_p^{(k-1)}(p) + \frac{1}{p^o_x(p_0)^2} | \hat{z}_{oo}(p) | + E_p^{(k-1)}(p) + \frac{1}{p^o_y(p_0)^2} | \hat{z}_{oo}(p) | + E_p^{(k-1)}(p), \nonumber
\]

(15)

where the subscripts \( p, o, \) and \( d \) respectively denote the variable which is parallel, orthogonal, and diagonal to the given axis (either \( u \)-axis or \( v \)-axis); the representations of \( E_p, E_o, \hat{z}_{pp}, \hat{z}_{oo}, p^p, p^o, \) and \( p^d \) are given in the supplement; and the indicator function

\[
\mathbb{I}(p, p') = \begin{cases} 0, & \text{if } (\hat{z}_{pp}(p + s_p) \cdot \hat{z}_{pp}(p + s_p) > 0) \\
\infty, & \text{otherwise} \end{cases} \nonumber
\]

(16)

add constraints to the monotonicity and convexity of the smoothness energy transfer function, improving the DP adaptivity to convex surfaces. The four terms in (15), in turn, denote the smoothness energy components when we introduce 1) no other pixels, 2) a parallel pixel \( p^p \), 3) an orthogonal pixel \( p^o \), and 4) a diagonal pixel \( p^d \).

### 3.3 Depth Gradient Update Function

This subsection discusses the formulation of the depth gradient update function \( \mathcal{R} \). Similar to \( \mathcal{T} \), depth gradient update can be divided into two categories: 1) collinear update \((s_u \times e_1 = 0 \lor s_u \times e_2 = 0)\) and 2) non-collinear update \((s_u \times e_1 \neq 0 \lor s_u \times e_2 \neq 0)\). For the collinear update, we use all the collinear pixels which satisfy the constraint in (16) to estimate depth gradients, while on the other hand, for the non-collinear update, we replace the depth gradient of each given pixel with the ones on other smoother paths using (11).

#### 3.3.1 Depth Gradient Collinear Update with Recursive Polynomial Interpolation

The optimum \( \nabla \hat{z} \) can be obtained by interpolating the depth of \( n \) collinear pixels into an \((n - 1)\)-th order polynomial and subsequently computing its derivative. Lagrange or Newton polynomial interpolation has redundant computations, as each pixel is repeatedly used for polynomial interpolation. Furthermore, it is incredibly complex to yield the closed-form solution of the polynomial’s derivative. To simplify the depth gradient collinear update process, we introduce a novel RPI algorithm, which can update \( \nabla \hat{z} \) recursively with only adjacent pixels until the farthest pixel is accessed.

As \( \hat{z}_u \) and \( \hat{z}_v \) can be computed in the same way, we only provide the details on \( \hat{z}_u \) computation in this paper. To compute \( \hat{z}_u \) of a given pixel \( p = [u_0, v_0] \) in the \( k \)-th iteration, we use a \((k+1)\)-th order polynomial to interpolate the depth of \((k + 2)\) pixels, which have the same vertical coordinates \( u_0 \). The \((k + 1)\)-th order polynomial \( f_0^{(k+1)}(u) \) interpolated by \( \{u_0, z(u_0), (u_0+1, z(u_0+1)), \ldots, (u_0+k+1, z(u_0+k+1))\} \) can be expanded into a recursive form as follows:

\[
f_0^{(k+1)}(u) = \frac{u_k + 1 - u}{k + 1} f_0^{(k+1)}(u_0) + \frac{u - u_0}{k + 1} f_1^{(k+1)}(u), \quad (17)
\]

where the \( k \)-th order polynomials \( f_0^{(k)} \) and \( f_1^{(k)} \) are respectively interpolated by the first and the last \( k + 1 \) pixels (the proof of the theorem is provided in the supplement).

Since \( \hat{z}_u^{(k-1)}(u_0) = \frac{df_0^{(k)}}{du}(u_0) \) and \( \hat{z}_u^{(k-1)}(u_0+1) = \frac{df_0^{(k)}}{du}(u_0+1) \) have been computed in the last iteration, we can update \( \hat{z}_u(u_0) \) using:

\[
\hat{z}_u^{(k)}(u_0) = \frac{df_0^{(k+1)}}{du}(u_0) \nonumber
\]

\[
= \hat{z}_u^{(k-1)}(u_0) - \frac{1}{k + 1} \left[ z(u_0) - f_1^{(k)}(u_0) \right]. \quad (18)
\]

Substituting \( f_1^{(k)}(u_0) = z(u_0+1) - \frac{df_0^{(k)}}{du}(u_0+1) = z(u_0+1) - z_u^{(k-1)}(u_0+1) \) into (18) yields

\[
\hat{z}_u^{(k)}(u_0) = \hat{z}_u^{(k-1)}(u_0) - \frac{1}{k + 1} \left[ z(u_0) - z(u_0 + 1) + z_u^{(k-1)}(u_0 + 1) \right]. \quad (19)
\]

#### 3.3.2 Depth Gradient Non-Collinear Update

Replacing the integral in (11) with summation yields:

\[
\hat{z}_p(p) \pm \hat{z}_o(p^p) = \hat{z}_p(p^o) \pm \hat{z}_o(p). \quad (20)
\]

As the pixels \( p^p \) are regarded as outliers in non-collinear update, we replace \( \hat{z}_o(p^p) \) with \( \hat{z}_o(p^o) \) or \( \hat{z}_o(p) \) based on...
the state variable. The depth gradient update function $\mathcal{R}$ in (6) can be rewritten as follows:

$$
\tilde{z}_{u}^{(k)}(p), \tilde{z}_{v}^{(k)}(p) = \\
\begin{cases}
\tilde{z}_{p}^{(k)}(p), & \text{if } s_{u}^{(k)}, s_{v}^{(k)} = 0 \\
\tilde{z}_{p}^{(k-1)}(p) + \frac{1}{\tau + 1} [z(p_{p}^{*}) - z(p)], & \text{if } s_{u}^{(k)}, s_{v}^{(k)} = p_{p}^{*} - p \\
\tilde{z}_{p}^{(k-1)}(p_{o}), & \text{if } s_{u}^{(k)}, s_{v}^{(k)} = p_{o}^{*} - p \\
\tilde{z}_{p}^{(k-1)}(p_{o}) + \frac{1}{\tau + 1} [z(p_{p}^{*}) - z(p_{p}^{*})], & \text{if } s_{u}^{(k)}, s_{v}^{(k)} = p_{o}^{*} - p \\
\end{cases}
$$

(21)

The representations of $\tilde{z}_{p}$ and $\tilde{z}_{v}$ are given in the supplement. The four terms in (21), in turn, represent 1) the unchanged depth gradient, 2) parallel update via RPI algorithm, 3) orthogonal update, and 4) diagonal update. The orthogonal and diagonal updates are based on the gradient theorem [25].

4. Experiments

4.1. Datasets and Evaluation Metrics

In this paper, we followed [7] and conducted comprehensive experiments to qualitatively and quantitatively evaluate the performance of our proposed SDA-SNE. More details on the 3F2N datasets are available in [7]. Moreover, since range sensor data are typically noisy, we add random Gaussian noise of different variances $\sigma$ to the original 3F2N datasets to compare the robustness between our proposed SDA-SNE and other Sota SNEs.

Two evaluation metrics are used to quantify the accuracy of SNEs: the average angular error (AAE) [15]:

$$
e_{\text{A}}(\mathcal{M}) = \frac{1}{P} \sum_{k=1}^{P} \phi_{k}, \quad \phi_{k} = \cos^{-1} \left( \frac{(\mathbf{n}_{k}, \mathbf{n}_{k}^{*})}{||\mathbf{n}_{k}|| \cdot ||\mathbf{n}_{k}^{*}||} \right),
$$

and the proportion of good pixels (PGP) [15]:

$$
e_{\text{P}}(\mathcal{M}) = \frac{1}{P} \sum_{k=1}^{P} \delta(\phi_{k}, \varphi_{k}), \quad \delta = \begin{cases} 0 & (\phi_{k} > \varphi) \\ 1 & (\phi_{k} \leq \varphi) \end{cases},
$$

where $P$ denotes the total number of pixels used for evaluation, $\varphi$ denotes the angular error tolerance, $\mathcal{M}$ represents the given SNE, and $\mathbf{n}_{k}$ and $\mathbf{n}_{k}^{*}$ represent the ground-truth and estimated surface normals, respectively.

In addition to the above-mentioned evaluation metrics, we also introduce a novel evaluation metric, referred to as cross accuracy ratio (CAR), to depict the performance comparison between two SNEs as follows:

$$
r_{\text{A}}(\mathcal{M}_{1}, \mathcal{M}_{2}) = \frac{e_{\text{A}}(\mathcal{M}_{1})}{e_{\text{A}}(\mathcal{M}_{2})}
$$

(24)

$\mathcal{M}_{2}$ outperforms $\mathcal{M}_{1}$ in terms of $e_{\text{A}}$ if $r_{\text{A}} > 1$, and vice versa if $0 < r_{\text{A}} < 1$.

4.2. Implementation Details

As discussed in Sec. 3, initial depth gradients can be estimated by convolving a depth image with image gradient filters, e.g., Sobel [21], Scharr [11], Prewitt [18], etc. The second-order filters, e.g., Laplace [20], can be used to estimate the local depth gradient smoothness $\hat{z}_{uu}$ and $\hat{z}_{uv}$. We utilize a finite forward difference (FFD) kernel, i.e., $[0, -1, 1]$, as well as a finite backward difference (FBD) kernel, i.e., $[-1, 1, 0]$ to initialize (coarse) $\hat{z}_{u}^{(0)}$ and $\hat{z}_{v}^{(0)}$, and use a finite Laplace (FL) kernel, i.e., $[-1, 2, -1]$ to estimate $\hat{z}_{uu}$ and $\hat{z}_{uv}$. We also conduct experiments w.r.t. different numbers of iterations to quantify the performance of our proposed SDA-SNE. By implementing the optimum maximum iteration, the trade-off between the speed and accuracy of our algorithm can be significantly minimized.

4.3. Algorithm Convergence and Computational Complexity

Our proposed SDA-SNE converges when the state variable and surface normal estimation remains stable. We can obtain the lower bound of the smoothness energy components as follows:

$$
e_{u}(p), e_{v}(p) \geq \min_{p \in \mathcal{N}(p)} \left\{ |\hat{z}_{uu}(p^*)|, |\hat{z}_{uv}(p^*)| \right\} \geq 0.
$$

(25)

The smoothness energy decreases monotonously after each iteration. Therefore, all state variables stop transferring after finite iterations (the smoothness energy converges to a global minimum).

The computational complexity of our proposed PRI algorithm is $O(n)$ (n denotes the number of interpolated pixels), when computing the depth gradient using (21). Therefore, PRI is much more efficient than Lagrange or Newton polynomial interpolation whose computational complexity is $O(n^2)$.

Furthermore, when the image resolution is $M \times N$ pixels, the computational complexity of our proposed SDA-SNE is $O(lMN)$, where $l$ denotes the number of iterations in DP. In most cases, $l \ll M, N$ because DP automatically stops when $|\hat{z}_{uu}|$ and $|\hat{z}_{uv}|$ no longer decrease. If we set a limit on $l$, SDA-SNE’s computational complexity can be reduced to $O(lMN)$, which is identical to the computational complexities of 3F2N [7] and CP2TV [16].

4.4. Performance Evaluation

As our proposed multi-directional DP algorithm mainly aims at improving the performance of depth gradient estimation, it is compatible with any depth-to-normal SNEs.
Figure 3. CAR comparison between CP2TV-based and 3F2N-based SDA-SNEs w.r.t. different maximum iterations.

The AAE scores of SoTA depth-to-normal SNEs and such SNEs using our depth gradient estimation strategy are given in Table 1. It can be observed that by using our proposed depth gradient estimation strategy, the AAE scores of such depth-to-normal SNEs decrease by about 20-60%. Since SDA-SNE based on CP2TV performs better than SDA-SNE based on 3F2N, we use CP2TV to estimate $n_z$ in the following experiments. The qualitative comparison among these SNEs is shown in Fig. 5. It can be observed that our proposed SDA-SNE significantly outperforms 3F2N and CP2TV near/on discontinuities.

Table 1. Comparison of $e_A$ (degrees) between 3F2N and CP2TV w/ and w/o our depth gradient estimation strategy leveraged.

| Datasets | 3F2N w/o SDA | 3F2N w/ SDA | CP2TV w/o SDA | CP2TV w/ SDA |
|----------|--------------|-------------|---------------|--------------|
| Easy     | 1.657        | **0.782**   | 1.686         | 0.677        |
| Medium   | 5.686        | 4.535       | 6.015         | 4.379        |
| Hard     | 15.315       | **9.237**   | 13.819        | **8.098**    |

Additionally, we compare the performance of our proposed SDA-SNE w.r.t. a collection of maximum iterations. Fig. 3 shows the CAR scores achieved by CP2TV-based and 3F2N-based SDA-SNEs on the 3F2N easy, medium, and hard datasets. It can be observed that the performance of SDA-SNE saturates after only several iterations. The accuracy increases by less than 5% when the maximum iteration is set to infinity. Therefore, the maximum iteration of multi-directional DP can be set to 3 to minimize its trade-off between speed and accuracy. Moreover, it can be observed that our proposed SDA-SNE converges (the CAR score reaches the maximum) with more iterations on the hard dataset than on the easy and medium datasets. This is probably due to the fact that the hard dataset possesses more discontinuities than the easy and medium datasets. As a result, more distant pixels are required to be introduced to yield better depth gradients.

Figure 4. AAE comparison among 3F2N, CP2TV, and SDA-SNE on the 3F2N datasets with different levels of Gaussian noise added.

Moreover, we compare our proposed SDA-SNE with all other SoTA SNEs presented in Sec. 2. $e_A$ and $e_P$ of all SNEs on the 3F2N easy, medium, and hard datasets are given in Table 2. The $e_A$ scores achieved by SDA-SNE are less than $1^\circ$ (easy), $5^\circ$ (medium), and $9^\circ$ (hard), respectively. The $e_P$ scores (tolerance: $10^\circ$) achieved by SDA-SNE are about 100% (easy), 90% (medium), and 80% (hard), respectively. These results suggest that our proposed SDA-SNE performs significantly better than all other SoTA SNEs, no matter whether an iteration limit is added or not.

Furthermore, we compare the robustness (to random Gaussian noise) of our proposed SDA-SNE with two SoTA depth-to-normal SNEs (3F2N [7] and CP2TV [16]), as shown in Fig. 4. The logarithms of AAE scores and Gaussian variances are used to include the results of different datasets into a single figure. It is evident that (1) the $e_A$ scores achieved by all SNEs increase monotonically with the increasing noise level, and (2) our proposed SDA-SNE outperforms 3F2N and CP2TV at different noise levels. In addition, the compared SNEs’ performance on the easy dataset degrades more dramatically than the medium and hard datasets, as the original easy dataset contains much fewer discontinuities. Although the performance of these depth-to-normal SNEs becomes remarkably similar, with the increase in noise level, our proposed SDA-SNE consistently outperforms 3F2N and CP2TV. This further demonstrates the superior robustness of our algorithm over others.

5. Conclusion

This paper presented a highly accurate discontinuity-aware surface normal estimator, referred to as SDA-SNE. Our approach computes surface normals from a depth image by iteratively introducing adjacent co-planar pixels using a novel multi-directional dynamic programming algorithm. To refine the depth gradient in each iteration, we introduced a novel recursive polynomial interpolation al-
| Method                    | $e_A$ (degrees) ↓ | Easy | Medium | Hard | $e_P$ ↑ | Easy | Medium | Hard |
|--------------------------|-------------------|------|--------|------|---------|------|--------|------|
| PlaneSVD [14]            | 2.07              | 6.07 | 17.59  |      | 0.9648  | 0.9792| 0.9855 |      |
| PlanePCA [12]            | 2.07              | 6.07 | 17.59  |      | 0.9648  | 0.9792| 0.9855 |      |
| VectorSVD [13]           | 2.13              | 6.27 | 18.01  |      | 0.9643  | 0.9777| 0.9846 |      |
| AreaWeighted [13]        | 2.20              | 6.27 | 17.03  |      | 0.9636  | 0.9753| 0.9819 |      |
| AngleWeighted [13]       | 1.79              | 5.67 | 13.26  |      | 0.9762  | 0.9862| 0.9893 |      |
| FALS [1]                 | 2.26              | 6.14 | 17.34  |      | 0.9654  | 0.9794| 0.9857 |      |
| SRI [1]                  | 2.64              | 6.71 | 19.61  |      | 0.9499  | 0.9713| 0.9798 |      |
| LINE-MOD [8]             | 2.64              | 6.71 | 19.61  |      | 0.8542  | 0.9085| 0.9343 |      |
| SNE-RoadSeg [6]          | 2.04              | 6.28 | 16.37  |      | 0.9693  | 0.9810| 0.9871 |      |
| 3F2N [7]                 | 1.66              | 5.69 | 15.31  |      | 0.9723  | 0.9829| 0.9889 |      |
| CP2TV [16]               | 1.69              | 6.02 | 13.82  |      | 0.9740  | 0.9843| 0.9899 |      |
| SDA-SNE (iteration = ∞)  | 0.68              | 4.38 | 8.10   |      | 0.9947  | 0.9982| 0.9991 |      |
| SDA-SNE (iteration = 3)  | 0.67              | 4.38 | 8.14   |      | 0.9947  | 0.9983| 0.9992 |      |

Figure 5. Examples of experimental results: Columns (i)-(v) show the depth images, surface normal ground truth, and the experimental results obtained using 3F2N, CP2TV, and our proposed SDA-SNE, respectively. Rows (a)-(c) show the results of the easy, medium, and hard datasets, respectively.

Algorithm with high computational efficiency. Our proposed depth gradient estimation approach is compatible with any depth-to-normal surface normal estimator, such as 3F2N and CP2TV. To evaluate the accuracy of our proposed surface normal estimator, we conducted extensive experiments on both clean and noisy datasets. Our proposed SDA-SNE achieves the highest accuracy on clean 3F2N datasets (0.68°, 4.38°, 8.10° on the easy, medium, and hard datasets, respectively), outperforming all other SoTA surface normal estimators. It also demonstrates high robustness to different levels of random Gaussian noise. Additional experimental results suggest that our proposed SDA-SNE can achieve a similar performance when reducing the maximum iteration of multi-directional dynamic programming to 3. This ensures the high efficiency of our proposed SDA-SNE in various computer vision and robotics applications requiring real-time performance.

6. Acknowledgements

This work was supported by the National Key R&D Program of China, under grant No. 2020AAA0108100, awarded to Prof. Qijun Chen. This work was also supported by the Fundamental Research Funds for the Central Universities, under projects No. 22120220184, No. 22120220214, and No. 2022-5-YB-08, awarded to Prof. Rui Fan.
References

[1] Hernan Badino, Daniel Huber, Yongwoon Park, and Takeo Kanade. Fast and accurate computation of surface normals from range images. In 2011 IEEE Int. Conf. on Robotics and Automation, pages 3084–3091. IEEE, 2011. 8

[2] Dániel Baráth, Ivan Eichhardt, and Levente Hajder. Optimal multi-view surface normal estimation using affine correspondences. *IEEE Transactions on Image Processing*, 28(7):3301–3311, 2019. 1, 2

[3] Richard Bormann, Joshua Hampp, Martin Hägele, and Markus Vincze. Fast and accurate surface normal estimation by efficient 3d edge detection. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3930–3937, 2015. 1

[4] Burath Daniel, Jozsef Molnar, and Levente Hajder. Optimal surface normal from affine transformation. In *International Conference on Computer Vision Theory and Applications*, volume 2, pages 305–316. SciTePress, 2015. 1, 2

[5] Rui Fan, Umar Ozgunalp, Brett Hosking, Ming Liu, and Ioannis Pitas. Pothole detection based on disparity transformation and road surface modeling. *IEEE Transactions on Image Processing*, 29:897–908, 2019. 1

[6] Rui Fan, Hengli Wang, Peide Cai, and Ming Liu. SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. In *European Conference on Computer Vision*, pages 340–356. Springer, 2020. 2, 8

[7] Rui Fan, Hengli Wang, Bohuan Xue, Huaiyang Huang, Yuan Wang, Ming Liu, and Ioannis Pitas. Three-Filters-to-Normal: An accurate and ultrafast surface normal estimator. *IEEE Robotics and Automation Letters*, 6(3):5405–5412, 2021. 1, 2, 6, 7, 8, 11

[8] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 34(5):876–888, 2011. 8

[9] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, 2012. 1

[10] Arvind V. Iyer and Johannes Burge. Depth variation and stereo processing tasks in natural scenes. *Journal of vision*, 18(6):4–4, 2018. 1

[11] Bernd Jähne, Hanno Scharr, Stefan Körkel, et al. Principles of filter design. *Handbook of computer vision and applications*, 2:125–151, 1999. 6

[12] Krzysztof Jordan and Philippos Mordohai. A quantitative evaluation of surface normal estimation in point clouds. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4220–4226. IEEE, 2014. 1, 2, 8

[13] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In 2009 IEEE International Conference on Robotics and Automation, pages 3206–3211. IEEE, 2009. 1, 2, 8

[14] Klaas Klasing, Dirk Wollherr, and Martin Buss. Realtime segmentation of range data using continuous nearest neighbors. In 2009 IEEE International Conference on Robotics and Automation, pages 2431–2436. IEEE, 2009. 1, 2, 8

[15] Feng Lu, Xiaowu Chen, Imari Sato, and Yoichi Sato. Symps: Brdf symmetry guided photometric stereo for shape and light source estimation. *IEEE trans on PAMI*, 40(1):221–234, 2017. 6

[16] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-Ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In 2015 International Conference on 3D Vision, pages 640–647, 2015. 1, 2, 6, 7, 8, 11

[17] D. Ouyang and Feng H. Y. On the normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design*, 37(10):1071–1079, 2005. 2

[18] Judith MS Prewitt et al. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970. 6

[19] Carolina Raposo and Joao P Barreto. Accurate reconstruction of oriented 3d points using affine correspondences. In *European Conference on Computer Vision*, pages 545–560. Springer, 2020. 1, 2

[20] Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patane, and Michela Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation, 2009. 6

[21] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 02 2014. 6

[22] W. Sun, C. Bradley, Y. F. Zhang, and H. T. Loh. Cloud data modelling employing a unified, non-redundant triangular mesh. *Computer-Aided Design*, 33(2):183–193, 2001. 2

[23] Hengli Wang, Rui Fan, Peide Cai, and Ming Liu. SNE-RoadSeg+: Rethinking depth-normal translation and deep supervision for freespace detection. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1140–1145. IEEE, 2021. 2

[24] Qiang Wang, Shizhen Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. In 2021 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6, 2021. 11

[25] Richard E Williamson and Hale F Trotter. *Multivariable mathematics*. Pearson, 2004. 4, 6

[26] Qiang-Jun Xie, Wen-Biao Jin, Li Ma, and Di-Bo Hou. Fast global segmentation based on the dual formulation of tv-norm. In 2010 3rd International Congress on Image and Signal Processing, volume 3, pages 1382–1385, 2010. 10
Supplementary Material

7. Details on Path Discontinuity Norm

Plugging the Taylor expansions of \( z_u \) and \( z_v \) into (10) and (11) results in the upper bound of the depth gradient estimation error w.r.t. collinear transfer and non-collinear transfer. The depth gradient estimation error can be associated with our introduced PD norm (detailed in Sec. 7.1 and 7.2). Furthermore, we compare PD norm with total variation (TV) norm [26], as detailed in Sec. 7.3.

7.1. Depth Gradient Error w.r.t. Collinear Transfer

In collinear transfer, (10) can be rewritten as follows:

\[
\begin{align*}
\varepsilon_C &= \left| \int_{u_0}^{u_1} (u - u_0)z_{uu} \, du \right|, \text{ or} \\
\varepsilon_C &= \int_{v_0}^{v_1} (v - v_0)z_{vv} \, dv. 
\end{align*}
\]

As \(|u_0 - u_1| \leq 1\) and \(|v_0 - v_1| \leq 1\), PD norm is the upper bound of the collinear transfer error \(\varepsilon_C\):

\[
\varepsilon_C \leq ||z||_{\text{PD}}, \quad \mathcal{L} : p \rightarrow p'.
\] (28)

7.2. Depth Gradient Error w.r.t. Non-Collinear Transfer

In non-collinear transfer, (11) can be revised as follows:

\[
\begin{align*}
(u_1 - u_0)z_u(p) + (v_1 - v_0)z_v(p') \\
+ \int_{\mathcal{L}_1} (u - u_0)z_{uu} \, du + (v - v_1)z_{vv} \, dv \\
= (v_1 - v_0)z_v(p) + (u_1 - u_0)z_u(p') \\
+ \int_{\mathcal{L}_2} (u - u_1)z_{uu} \, du + (v - v_0)z_{vv} \, dv.
\end{align*}
\] (29)

PD norm is also the upper bound of the non-collinear transferring error \(\varepsilon_N\) w.r.t. different DP paths:

\[
\begin{align*}
\varepsilon_N &= \left| \int_{\mathcal{L}_1} (u - u_0)z_{uu} \, du + (v - v_1)z_{vv} \, dv \right| \\
&\quad + \int_{\mathcal{L}_2} (u - u_1)z_{uu} \, du + (v - v_0)z_{vv} \, dv \\
&\leq \int_{\mathcal{L}_1} |z_{uu} \, du| + |z_{vv} \, dv| + \int_{\mathcal{L}_2} |z_{uu} \, du| + |z_{vv} \, dv| \\
&= ||z||_{\text{PD}}, \quad \mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 : p \rightarrow p' \rightarrow p.
\end{align*}
\] (30)

Therefore, the PD norm determines the upper bound of the depth gradient estimation error. Optimizing the PD norm in (14) is equivalent to minimizing the error of \(z_u\) and \(z_v\) in (27) and (30).

7.3. Comparison between PD Norm and TV Norm

TV norm [26] is widely used to measure depth discontinuity. Compared to TV norm, our proposed PD norm introduces the second-order partial derivative of depth, demonstrating better awareness of ridges. We conduct additional experiments on the 3F2N datasets to compare the accuracy of our method using PD and TV norms, respectively. As shown in Table 3, the accuracy achieved when using PD norm is higher than that achieved when using TV norm.

| Method                        | Easy | Medium | Hard |
|-------------------------------|------|--------|------|
| SDA-SNE (with TV norm)        | 1.37 | 5.14   | 10.05|
| SDA-SNE (with PD norm)        | 0.67 | 4.38   | 8.14 |

8. RPI Theorem

**Theorem:** If the \( n \)-th order polynomial \( f(u)^{(n)} \in P^n[u] \) is interpolated by \((u_0, z_0), (u_1, z_1), \ldots, (u_n, z_n)\), it can be expanded into a recursive form as follows:

\[
f^{(n)}(u) = \frac{u - u_n}{u_0 - u_n} f_0^{(n-1)}(u_0) + \frac{u - u_0}{u_n - u_0} f_1^{(n-1)}(u_1)\]

where \( f_0^{(n-1)} \) and \( f_1^{(n-1)} \) are respectively interpolated by the first and the last \( n \) points.

**Proof**

(a) Substituting \( u = u_0 \) into (31), we can obtain:

\[
\begin{align*}
\frac{u - u_n}{u_0 - u_n} &= 1, \quad \frac{u - u_0}{u_n - u_0} = 0, \\
f^{(n)}(u_0) &= f_0^{(n-1)}(u_0) = z_0.
\end{align*}
\] (32)

(b) Substituting \( u = u_n \) into (31), we can obtain:

\[
\begin{align*}
\frac{u - u_n}{u_0 - u_n} &= 0, \quad \frac{u - u_0}{u_n - u_0} = 1, \\
f^{(n)}(u_n) &= f_1^{(n-1)}(u_n) = z_n.
\end{align*}
\] (33)

(c) Substituting \( u = u_1, \ldots, u_{n-1} \) into (31), we can obtain:

\[
\begin{align*}
\frac{u - u_0}{u_n - u_0} + \frac{u - u_n}{u_0 - u_n} &= 1, \\
f^{(n)}(u_i) &= f_0^{(n-1)}(u_i) = f_1^{(n-1)}(u_i) = z_i,
\end{align*}
\] (34)

where \( i \) is an integer between 1 and \( n - 1 \).
9. Additional Experiments on Noisy Data

We also conduct additional experiments on the IRS dataset \(^4\) [24] to further validate our proposed SDA-SNE’s robustness to noise. The IRS dataset contains four indoor scenarios (home, office, restaurant, and store) [24]. The comparison among 3F2N [7], CP2TV [16], and our proposed SDA-SNE in terms of \(e_A\) is given in Table 4. It can be observed that our proposed SDA-SNE also outperforms 3F2N and CP2TV on the IRS dataset (\(e_A\) is lower by about 40% and 30%, respectively). These results suggest that SDA-SNE performs robustly on real noisy data under different illumination conditions. Qualitative comparison among 3F2N, CP2TV, and SDA-SNE is given in Fig. 6. It can be observed that compared to 3F2N and CP2TV, SDA-SNE performs much better near/on discontinuities.

Table 4. Comparison of \(e_A\) (degrees) among 3F2N, CP2TV, and our proposed SDA-SDA on the IRS dataset.

| Method      | \(e_A\) (degrees) | Home | Office | Restaurant | Store |
|-------------|-------------------|------|--------|------------|-------|
| 3F2N [7]    | 6.2665            | 6.9919 | 8.2611 | 7.4258     |       |
| CP2TV [16]  | 5.6835            | 6.2551 | 7.3373 | 6.2356     |       |
| SDA-SDA (ours) | 3.9073        | 4.4796 | 5.2782 | 4.4245     |       |

\(^4\)github.com/HKBU-HPML/IRS

10. Detailed Representations of Subscripts \(p, o, \) and \(d\) in Multi-Directional DP

The representations of \(E_p^{(k-1)}, E_o^{(k-1)}, \hat{z}_{pp},\) and \(\hat{z}_{oo}\) in (15) are shown as follows:

\[
E_p^{(k-1)} = \begin{cases} 
E_u^{(k-1)}, & \text{when computing } E_u^{(k)} \\
E_v^{(k-1)}, & \text{when computing } E_v^{(k)}
\end{cases}
\]

and

\[
\hat{z}_{pp} = \begin{cases} 
\hat{z}_{uu}, & \text{when computing } E_u^{(k)} \\
\hat{z}_{vv}, & \text{when computing } E_v^{(k)}
\end{cases}
\]

(36)

The representations of \(z_{p}^{(k-1)}\) and \(z_{o}^{(k-1)}\) in (21) are shown as follows:

\[
\hat{z}_{o}^{(k-1)} = \begin{cases} 
\hat{z}_{u}^{(k-1)}, & \text{when computing } \hat{z}_{u}^{(k)} \\
\hat{z}_{v}^{(k-1)}, & \text{when computing } \hat{z}_{v}^{(k)}
\end{cases}
\]

(37)

Given a pixel \(p = [u_0, v_0]^T\), its adjacent pixels \(p_p^t, p_o^t, \) and \(p_d^t\) are denoted as follows:

\[
p_p^t = \begin{cases} 
[u_0 \pm 1, v_0]^T, & \text{when computing } E_u^{(k)} \text{ or } \hat{z}_{u}^{(k)} \\
[u_0, v_0 \pm 1]^T, & \text{when computing } E_v^{(k)} \text{ or } \hat{z}_{v}^{(k)}
\end{cases}
\]

\[
p_o^t = \begin{cases} 
[u_0, v_0 \pm 1]^T, & \text{when computing } E_u^{(k)} \text{ or } \hat{z}_{u}^{(k)} \\
[u_0 \pm 1, v_0]^T, & \text{when computing } E_v^{(k)} \text{ or } \hat{z}_{v}^{(k)}
\end{cases}
\]

(38)
### 11. Nomenclature

| Notation | Description |
|----------|-------------|
| $p^c = [x, y, z]$ | Coordinate of a 3D point in the camera coordinate system |
| $p = [u, v]^T$ | Coordinate of a 2D pixel in the pixel coordinate system |
| $p'$ | Adjacent pixel of a given pixel $p$ |
| $K$ | Camera intrinsic matrix |
| $\nabla z$ | Theoretical depth gradient |
| $\hat{\nabla} z$ | Estimated depth gradient |
| $n$ | Theoretical surface normal |
| $\hat{n}$ | Estimated surface normal |
| $\cdot(k)$ | The $k$-th iteration |
| $\cdot^*$ | The optimal value |
| $E_u, E_v$ | Smoothness energy components on the $u$-axis and $v$-axis |
| $s_u, s_v$ | State variable components on the $u$-axis and $v$-axis |
| $\mathcal{T}$ | Smoothness energy transfer function |
| $\mathcal{R}$ | Depth gradient update function |
| $\Delta f z$ | Forward finite difference of depth on the $u$-axis and $v$-axis |
| $\Delta b z$ | Backward finite difference of depth on the $u$-axis and $v$-axis |
| $z_u, z_v$ | Theoretical first-order partial derivatives of depth on the $u$-axis and $v$-axis |
| $\hat{z}_u, \hat{z}_v$ | Estimated first-order partial derivatives of depth on the $u$-axis and $v$-axis |
| $z_{uu}, z_{vv}$ | Theoretical second-order partial derivatives of depth on the $u$-axis and $v$-axis |
| $\hat{z}_{uu}, \hat{z}_{vv}$ | Estimated second-order partial derivatives of depth on the $u$-axis and $v$-axis |
| $\parallel \cdot \parallel_{PD}$ | Path discontinuity norm |
| $f^{(n)}$ | The $n$-th order polynomial |
| $e_A$ | Average angular error |
| $e_P$ | Proportion of good pixels |
| $r_A$ | Cross accuracy ratio |
| $\varepsilon_C$ | Collinear transfer error of depth gradient estimation |
| $\varepsilon_N$ | Non-collinear transfer error of depth gradient estimation |

Table 5. Nomenclature