2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

D2NT: A High-Performing Depth-to-Normal Translator

Yi Feng, Bohuan Xue, Ming Liu, Qijun Chen, and Rui Fan✉

***Abstract*— Surface normal holds significant importance in visual environmental perception, serving as a source of rich geometric information. However, the state-of-the-art (SoTA) surface normal estimators (SNEs) generally suffer from an unsatisfactory trade-off between efficiency and accuracy. To resolve this dilemma, this paper first presents a superfast depth-to-normal translator (D2NT), which can directly translate depth images into surface normal maps without calculating 3D coordinates. We then propose a discontinuity-aware gradient (DAG) filter, which adaptively generates gradient convolution kernels to improve depth gradient estimation. Finally, we propose a surface normal refinement module that can easily be integrated into any depth-to-normal SNEs, substantially improving the surface normal estimation accuracy. Our proposed algorithm demonstrates the best accuracy among all other existing real-time SNEs and achieves the SoTA trade-off between efficiency and accuracy.**

Source Code, Demo Video, & Supplement

Our source code, demo video, and supplement are publicly available at mias.group/D2NT.

I. Introduction

Surface normal is an informative visual feature that has been widely used in a variety of robot environmental perception tasks, *e.g.*, visual odometry [1], [2], scene parsing [3]–[7], and depth estimation [8], [9]. Due to the requirement for real-time execution in such tasks, surface normal estimators (SNEs) should be both accurate and computationally efficient [10].

Early geometry-based SNEs compute surface normals via either plane fitting (solvable with energy minimization techniques) or weighted neighboring normal aggregation. However, these SNEs typically have an imbalance between accuracy and speed (see Fig. 1). In 2015, Nakagawa *et al.* [11] proposed an efficient SNE, which computes surface normals via the cross product of two orthogonal tangent vectors (hereafter called CP2TV). However, its performance on spatial discontinuities is unsatisfactory as a result of inaccurate observed tangent vectors. Recently, Fan *et al.* [10] introduced an efficient and accurate SNE, referred to as three-filters-to-normal (3F2N). Although 3F2N achieves state-of-the-art (SoTA) performance, the aggregation of neighboring surface normals with a mean or median filter is still computationally intensive.

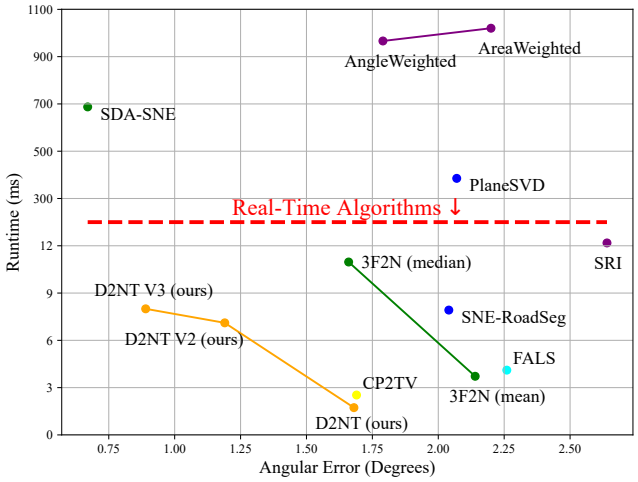

Fig. 1. Efficiency versus accuracy trade-off comparison among all SoTA geometry-based SNEs (on the 3F2N Easy dataset). D2NT has the highest computational efficiency, and D2NT V3 achieves the best trade-off between speed and accuracy.

Therefore, there is a strong necessity to develop an SNE that achieves a balance between rapid computation and high accuracy. In this paper, we present a high-performing depth-to-normal translator (D2NT), which significantly improves the efficiency and accuracy trade-off, and significantly refines the estimation results in and around discontinuities. The contributions of our work are summarized as follows:

1) **D2NT**, a cutting-edge **D**epth-**to**-**N**ormal **T**ranslator. In comparison to other geometry-based SNEs, D2NT computes surface normals directly from depth maps, demonstrating remarkable computational efficiency. Compared to existing SoTA SNEs, D2NT establishes the most direct relationship between depth and surface normal.
2) **D**iscontinuity-**A**ware **G**radient (**DAG**) filter, a depth gradient filter that selectively identifies discontinuities and eliminates outliers (non-coplanar points in relation to the reference point). Compared to traditional finite difference (FD) operators, our proposed DAG filter provides a significant improvement in terms of depth gradient estimation accuracy.

This work was supported by the National Key R&D Program of China under Grant 2020AAA0108100, the National Natural Science Foundation of China under Grant 62233013, the Science and Technology Commission of Shanghai Municipal under Grant 22511104500, the Fundamental Research Funds for the Central Universities under Grants 22120220184 and 22120220214, and the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100. *(Yi Feng and Bohuan Xue contributed equally to this work.) (Corresponding author: Rui Fan.)*

Yi Feng, Qijun Chen, and Rui Fan are with the Robotics & Artificial Intelligence Laboratory (RAIL), the College of Electronic & Information Engineering, the State Key Laboratory of Intelligent Autonomous Systems, and Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai 201804, P. R. China. (e-mails: fengyi@ieee.org, qjchen@tongji.edu.cn, rui.fan@ieee.org)

Bohuan Xue is with the Department of Computer Science & Engineering, the Hong Kong University of Science and Technology, Hong Kong SAR, P. R. China. (e-mail: bxueaa@ust.hk)

Ming Liu is with the Robotics & Autonomous Systems Thrust of the Systems Hub, the Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou 511400, P. R. China. (e-mail: eelium@ust.hk)

3) **M**arkov random field-based **N**ormal **R**efinement (**MNR**) module, which dramatically reduces surface normal estimation errors. It can also be integrated with any depth-to-normal SNEs to further enhance the quality of their estimated surface normals.

II. Related work

This section provides an overview of geometry-based surface normal estimators. As shown in Table I, the existing SNEs can be divided into three categories: energy minimization-based, averaging-based, and depth-to-normal.

Let $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n\}$ be the given 3D point set. For an arbitrary 3D point $\boldsymbol{p}_i \in \boldsymbol{P}$, its surface normal is represented as $\boldsymbol{n}_i = [n_{ix}, n_{iy}, n_{iz}]^\top$. To find the optimal $\boldsymbol{n}_i$, $\boldsymbol{Q}_i = \{\boldsymbol{q}_{i1}, \boldsymbol{q}_{i2}, ..., \boldsymbol{q}_{ik} \mid \boldsymbol{q}_{ik} \in \boldsymbol{P}\}$, the neighboring points of $\boldsymbol{p}_i$ are typically considered.

A. Energy Minimization-Based Methods

This category of methods computes surface normals by finding a best-fit plane from the augmented neighboring point set $\boldsymbol{Q}_i^+ = \{\boldsymbol{Q}_i, \boldsymbol{p}_i\}$ as follows:

$$\hat{\boldsymbol{n}}_i = \underset{\boldsymbol{n}_i}{\arg\min}\ E(\boldsymbol{Q}_i^+, \boldsymbol{n}_i), \tag{1}$$

where $\hat{\boldsymbol{n}}_i$ is obtained by minimizing the energy function E.

PlaneSVD [12] fits a local planar surface to $\boldsymbol{Q}_i^+$ by minimizing the least squares of the distances from the points to the surface using SVD. Similarly, PlanePCA [13] finds the minimum variance of $\boldsymbol{Q}_i^+$ with respect to the centroid $\bar{\boldsymbol{p}}_i = \frac{1}{k+1}(\boldsymbol{p}_i + \Sigma_{j=1}^{k} \boldsymbol{q}_{ij})$. VectorSVD [14] fits the local planar surface by minimizing the sum of the squared dot products between the surface normal and tangent vectors.

Recently, Ming *et al.* [15] proposed SDA-SNE, a highly accurate surface normal estimator based on multi-directional dynamic programming and iterative polynomial interpolation. Nevertheless, its demanding computational requirements and iterative nature result in subpar real-time performance. The computation-intensive nature of energy minimization and the calculation of 3D coordinates make these SNEs suffer from slow processing speed and noise.

B. Averaging-Based Methods

This category of methods estimates surface normals by averaging the normal vectors of the surrounding triangles:

$$\boldsymbol{n}_i = \frac{1}{k}\sum_{j=1}^{k} w_j \frac{\boldsymbol{r}_{ij} \times \boldsymbol{r}_{ij+1}}{\|\boldsymbol{r}_{ij} \times \boldsymbol{r}_{ij+1}\|_2}, \tag{2}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{q}_{ij} - \boldsymbol{p}_i$, $\boldsymbol{r}_{ik+1} = \boldsymbol{r}_{i1}$, and w_j is the weight calculated based on either the area (AreaWeighted [16]) or the angle (AngleWeighted [17]) of the triangles. Nonetheless, both of these methods necessitate an initial estimation of the normals and can only be utilized as a back-end optimization technique.

TABLE I
Taxonomy of the SoTA geometry-based SNEs.

Category	Algorithm	Expression
Energy Minimization-Based	PlaneSVD [12]	$\min \left\|\left\| \left[\boldsymbol{Q}_i^+, \boldsymbol{1}_k\right] \boldsymbol{b}_i \right\|\right\|_2$
	PlanePCA [13]	$\min \left\|\left\| \left[\boldsymbol{Q}_i^+ - \bar{\boldsymbol{p}}\right] \boldsymbol{n}_i \right\|\right\|_2$
	VectorSVD [14]	$\min \left\|\left\| \left[\boldsymbol{Q}_i - \boldsymbol{1}_k \boldsymbol{p}_i^\top\right] \boldsymbol{n}_i \right\|\right\|_2$
	SDA-SNE [15]	$\min \left\{ \boldsymbol{\mathcal{T}} \left(\boldsymbol{E}^{(k-1)}, \boldsymbol{S} \right) \right\}$
Averaging-Based	AreaWeighted [16]	$w_j = \frac{1}{2} \left\| \boldsymbol{r}_{ij} \times \boldsymbol{r}_{ij+1} \right\|_2$
	AngleWeighted [17]	$w_j = \cos^{-1}\left(\frac{\langle \boldsymbol{r}_{ij}, \boldsymbol{r}_{ij+1} \rangle}{\|\boldsymbol{r}_{ij}\|_2 \|\boldsymbol{r}_{ij+1}\|_2} \right)$
Depth-to-Normal	3F2N [10]	$n_x = f_x \frac{\partial 1/z}{\partial u}, \quad n_y = f_y \frac{\partial 1/z}{\partial v}$ $\hat{n}_z = -\Phi \left\{ \frac{\Delta x_{ij} n_x + \Delta y_{ij} n_y}{\Delta z_{ij}} \right\}$
	CP2TV [11]	$\boldsymbol{n}_i = \boldsymbol{t}_u \times \boldsymbol{t}_v$

C. Depth-to-Normal Methods

Fan *et al.* [10] proposed 3F2N, a fast and accurate surface normal estimator, which directly converts the structured range sensor data, such as depth or disparity images, into surface normal maps using two gradient filters and a mean/median filter. This category of methods typically assumes that the range sensor is a pinhole camera model as follows:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K}\boldsymbol{p}_i = \begin{bmatrix} f_x & 0 & u_{\mathrm{o}} \\ 0 & f_y & v_{\mathrm{o}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{3}$$

where $\boldsymbol{K}$ represents the camera intrinsic matrix, $\boldsymbol{p_0} = [u_0, v_0]^\top$ is the principal point in pixels, and f_x and f_y denote the camera's focal lengths in the x and y directions, respectively. This method achieves fast computational speed and high accuracy, but it still involves the calculation of 3D coordinates, which is redundant and computationally demanding.

Nakagawa *et al.* [11] presented CP2TV, an SNE that utilizes the cross-products of tangent vectors of local planar surfaces to directly estimate surface normals from depth maps. However, the accuracy of this method is inadequate in and around discontinuities, as it adopts a finite difference operator to estimate depth gradients. Inaccurate tangent vectors generated in these regions lead to substantial calculation errors in the estimated surface normals.

III. Methodology

In this section, we first introduce a highly efficient method for estimating surface normals from structured range sensor data in an end-to-end manner. Then, we present a novel approach to improve the accuracy of depth gradient estimation. Additionally, we propose an optimization strategy to refine surface normal estimation in and around discontinuities, which can be well embedded into any existing depth-to-normal SNEs. The pipeline of our algorithm is illustrated in Fig. 2.

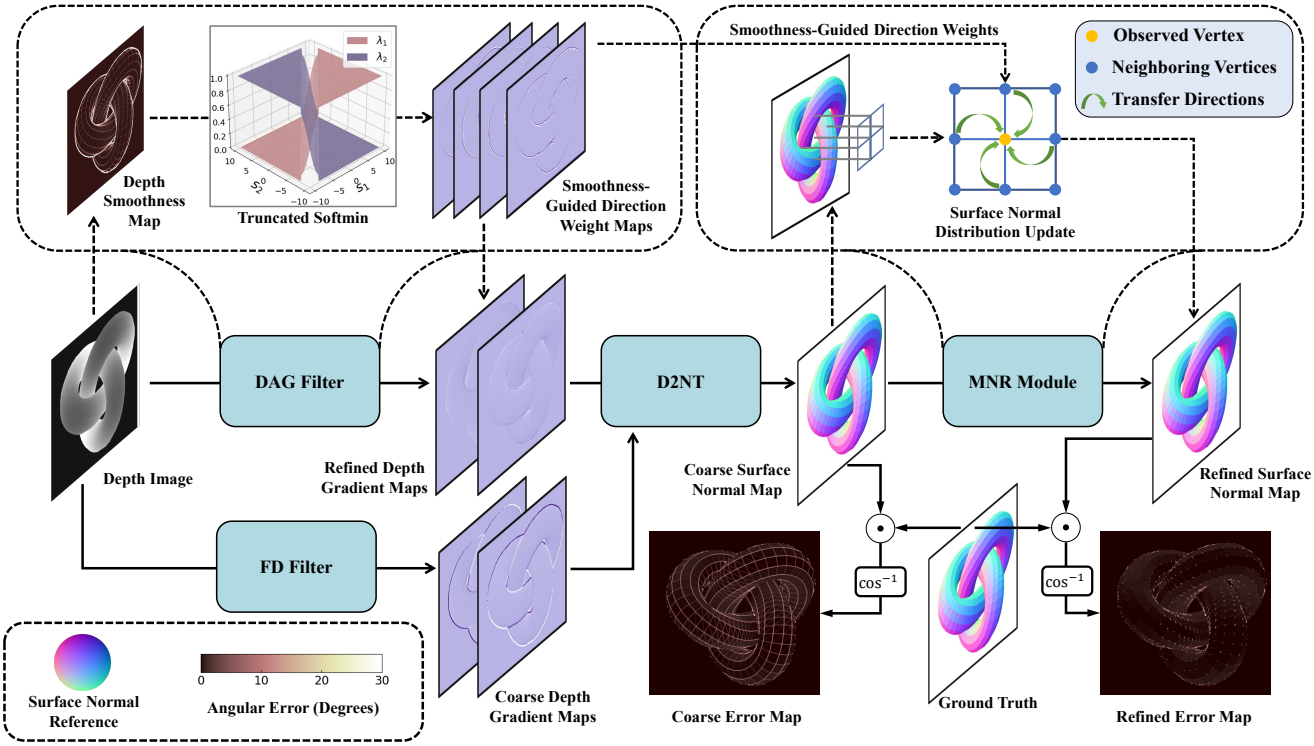

Fig. 2. The illustration of our proposed D2NT, DAG filter, and MNR module. D2NT translates depth images into surface normal maps in an end-to-end fashion; DAG filter adaptively generates smoothness-guided direction weights for improved depth gradient estimation in and around discontinuities; MNR module further refines the estimated surface normals based on the smoothness of neighboring pixels.

A. Depth-to-Normal Translation

An observed 3D point $\boldsymbol{p} = [x, y, z]^\top$ and its surface normal $\boldsymbol{n} = [n_x, n_y, n_z]^\top$ have the following relation:

$$n_x x + n_y y + n_z z + d = 0, \tag{4}$$

where d is the distance between the origin and the tangent plane. Combining (3) with (4) results in the following expression:

$$\frac{z(u-u_0)}{f_x} n_x + \frac{z(v-v_0)}{f_y} n_y + n_z z + d = 0. \tag{5}$$

(5) contains an implicit function $z(u, v)$. We compute the partial derivatives of z with respect to u and v, as follows:

$$\begin{aligned} z_u\left(\frac{u-u_0}{f_x} n_x + \frac{v-v_0}{f_y} n_y + n_z\right) + \frac{z}{f_x} n_x = 0, \\ z_v\left(\frac{u-u_0}{f_x} n_x + \frac{v-v_0}{f_y} n_y + n_z\right) + \frac{z}{f_y} n_y = 0, \end{aligned} \tag{6}$$

where $z_u = \frac{\partial z}{\partial u}$ and $z_v = \frac{\partial z}{\partial v}$. n_x and n_y can then be obtained by plugging (5) into (6):

$$n_x = \frac{f_x d}{z^2} z_u, \quad n_y = \frac{f_y d}{z^2} z_v. \tag{7}$$

n_z can therefore be computed by plugging (7) into (5):

$$n_z = -\frac{d}{z^2}(z + (u-u_0) z_u + (v-v_0) z_v). \tag{8}$$

Removing the common factor $\frac{d}{z^2}$ results in a simplified expression for surface normal:

$$\boldsymbol{n} = \begin{bmatrix} -f_x & 0 & 0 \\ 0 & -f_y & 0 \\ u-u_0 & v-v_0 & z \end{bmatrix} \begin{bmatrix} z_u \\ z_v \\ 1 \end{bmatrix}. \tag{9}$$

(9) describes an end-to-end translation from a given depth image to its surface normal map. Compared with other SoTA SNEs, such as 3F2N [10] and SNE-RoadSeg [3], D2NT eliminates the need to calculate 3D coordinates by leveraging the explicit relationship between depth and normal, demonstrating remarkable computational efficiency.

B. Discontinuity-Aware Gradient Filtering

As (9) demonstrates, surface normals can be directly calculated from structured range sensor data when the camera parameters are known. The accuracy of the partial derivatives directly affects the accuracy of the surface normal estimation. This subsection introduces an improved depth gradient computation approach.

The existing depth-to-normal methods generally utilize regular image gradient filters, such as FD[1], to approximate depth gradients. However, these filters tend to yield poor results on discontinuities, such as ridges, ditches, and edges, as outliers (non-coplanar adjacent points) are involved in depth difference computation.

To address this issue, we define a horizontal gradient filter G_h and a vertical gradient filter G_v as follows:

$$G_h = \lambda_l \Delta_b + \lambda_r \Delta_f, \quad G_v = \lambda_u \Delta_b^\top + \lambda_d \Delta_f^\top, \tag{10}$$

[1]Horizontal FD kernel: $\Delta = [-1, 0, 1]$.

where $\Delta_b = [-1, 1, 0]$ and $\Delta_f = [0, -1, 1]$ are the backward and forward difference operators, respectively. λ denotes the weight distribution along four different directions. Hereafter the subscripts l, r, u, and d denote left, right, up, and down directions, respectively.

To obtain more accurate G_u and G_v in areas with discontinuities, we must distinguish between distinct continuous surfaces and assign appropriate weights to Δ_f and Δ_b based on the smoothness of the neighboring pixels' surfaces. The local surface smoothness $s_{\boldsymbol{p}}$ can be reflected by:

$$s_{\boldsymbol{p}} = |\nabla^2 z_{\boldsymbol{p}}|, \tag{11}$$

where ∇^2 is a second-order Discrete Laplace Filter (DLF), and $z_{\boldsymbol{p}}$ is the depth at $\boldsymbol{p}$. The weights of the difference operator along four directions can then be assigned as follows:

$$\lambda_l, \lambda_r = \mathscr{M}(s_l, s_r), \quad \lambda_u, \lambda_d \quad = \mathscr{M}(s_u, s_d), \tag{12}$$

where s_l, s_r, s_u, and s_d represent the smoothness of four neighboring points, respectively, and $\mathscr{M}$ is the $\mathrm{softmin}$ function:

$$\mathscr{M}(s_i) = \frac{e^{-s_i/\tau}}{\sum_{j=1}^{n} e^{-s_j/\tau}}, \quad i = 1 \text{ or } 2, \tag{13}$$

where τ is the coefficient that regulates the "softness" of the $\mathrm{softmin}$ function. Additionally, we observe that significant estimation errors generally occur on the boundaries of surfaces which have small depth gradients (*i.e.*, surfaces that are nearly parallel to the XOY plane). This is due to the fact that the adjacent plane typically exhibits a much larger difference in depth gradient magnitude when compared to the reference surface (*i.e.*, the plane where the reference point is located). As a result, the calculated depth gradient is bound to differ from the depth gradient of the reference plane, even if the weight assigned to the reference plane's depth gradient is high, according to (13). To tackle this problem, when the smoothness of neighboring points differs greatly from each other, the weights for the depth gradients of adjacent and reference planes should be automatically assigned to 0 and 1, respectively. Therefore, we introduce $\mathrm{truncated\ softmin}$

$$\mathscr{M}_t(s_i) = \begin{cases} \mathscr{M}(s_i) & (|s_2 - s_1| \leq 1) \\ \mathbb{1}_{\mathbb{R}^+}(s_i - 1) & (|s_2 - s_1| > 1) \end{cases} \tag{14}$$

to further improve surface normal accuracy, where $\mathbb{1}_{\mathbb{R}^+}(\cdot)$ is the indicator function mapping weight to either 0 or 1 based on the difference in adjacent pixel's surface smoothness. Our proposed DAG filter adaptively generates gradient filters based on surface smoothness, resulting in more accurate estimations of depth gradients, as outliers are effectively filtered out. In summary, the gradient filter of a given point $\boldsymbol{p}_i$ can be represented by the following expression[2]:

$$\boldsymbol{G}_i = \begin{bmatrix} G_h \\ G_v^\top \end{bmatrix}_i = \mathscr{M}_t(|\nabla^2 \boldsymbol{z}_i|) \begin{bmatrix} \Delta_b \\ \Delta_f \end{bmatrix}, \tag{15}$$

where $\boldsymbol{z}_i = \left[\begin{smallmatrix} z_l & z_r \\ z_u & z_d \end{smallmatrix}\right]$ is the neighborhood depth matrix of $\boldsymbol{p}_i$.

[2]Here $\nabla^2(\cdot)$, $|\cdot|$, and $\mathscr{M}_t(\cdot)$ are element-wise operators.

C. MRF-Based Surface Normal Refinement

Our observation reveals that the surface normals of pixels near/on discontinuities are generally incorrect. This is due to the fact that non-coplanar points are used for local planar surface fitting, causing incorrect depth gradients (discussed in the previous subsection). To resolve this issue, we propose a fast and effective MRF-based optimization (post-processing) method, which significantly improves surface normal accuracy while having minimal impact on the processing speed.

The depth image can be modeled as an undirected graph $\mathcal{G} = (\mathcal{P}, \mathcal{E})$, where each node represents a pixel in the depth map, and each edge describes the connection between adjacent pixels. Let $\boldsymbol{N} = \{\boldsymbol{n_p} \,|\boldsymbol{p} \in \mathcal{P}\}$ be a random variable set, where $\boldsymbol{n_p}$ represents the estimated surface normal of point $\boldsymbol{p}$. $\boldsymbol{n_p}$ is conditionally independent of all other variables in $\boldsymbol{N}$:

$$\boldsymbol{n_p} \perp\!\!\!\perp \boldsymbol{n}_{\mathcal{P}\setminus\boldsymbol{Q_p^+}} \mid \boldsymbol{n}_{\boldsymbol{Q_p}}. \tag{16}$$

Let $P(\boldsymbol{N} = n)$ be the joint probability distribution of $\boldsymbol{N}$, representing the probability of a particular field configuration n in surface normal field $\boldsymbol{N}$. Specifically, we set the size of the maximum clique to 2 to model our pairwise MRF. According to the Hammersley-Clifford theorem [18], $P(\boldsymbol{N} = n)$ is represented as follows:

$$P(\boldsymbol{N} = n) = \frac{1}{Z} \prod_{\boldsymbol{p} \in \mathcal{P}} \phi(\boldsymbol{p}) \prod_{(\boldsymbol{p}, \boldsymbol{q_i}) \in \mathcal{E}} \psi(\boldsymbol{p}, \boldsymbol{q_i}), \tag{17}$$

where Z is the partition function. (17) is mathematically equivalent to the energy minimization problem as follows:

$$\boldsymbol{E} = \sum_{\boldsymbol{p} \in \mathcal{P}} \Phi(\boldsymbol{p}) + \sum_{(\boldsymbol{p}, \boldsymbol{q}_i) \in \mathcal{E}} \Psi(\boldsymbol{p}, \boldsymbol{q}_i), \tag{18}$$

where the data term

$$\Phi(\boldsymbol{p}) = ||\hat{\boldsymbol{n}}_{\boldsymbol{p}} - \boldsymbol{n_p}||_2, \tag{19}$$

enforces the consistency between the estimated surface normals $\hat{\boldsymbol{n}}_{\boldsymbol{p}}$ and the observed surface normals $\boldsymbol{n_p}$, and the smoothness term

$$\Psi(\boldsymbol{p}, \boldsymbol{q}) = s_{\boldsymbol{p}} \sum_i \mathscr{M}(\boldsymbol{s}_i) ||\hat{\boldsymbol{n}}_{\boldsymbol{p}} - \boldsymbol{n}_{\boldsymbol{q}_i}||_2, \tag{20}$$

smoothens the surface normal distribution between reference point $\boldsymbol{p}$ and its neighboring point $\boldsymbol{q}_i$, where $\mathscr{M}(\boldsymbol{s}_i)$ is the weight of the neighboring point $\boldsymbol{q}_i$ generated by the $\mathrm{softmin}$ function (13), $\boldsymbol{n}_{\boldsymbol{q}_i}$ is the observed surface normal of $\boldsymbol{q}_i$, and $s_{\boldsymbol{p}}$ is the local surface smoothness that decides the weight between data term and smoothness term.

(19) suggests that the difference between the observed and estimated surface normals should be insignificant, while (20) implies that adjacent points on the same local planar surface should have consistent normal distributions.

IV. Experiments

This section evaluates the performance of our proposed surface normal estimator and compares it with SoTA geometry-based SNEs. To simplify the presentation, we refer

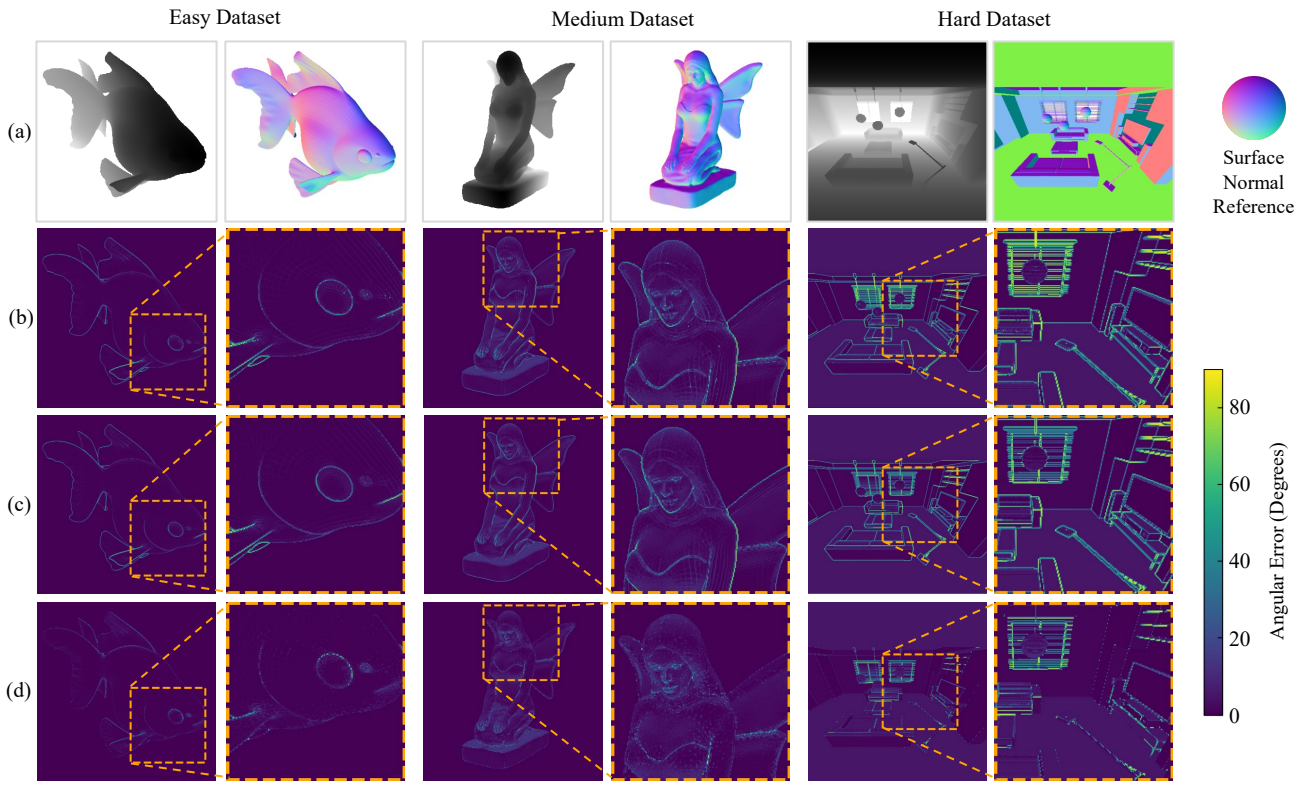

Fig. 3. Comparison of our proposed SNE with other SoTA geometry-based SNEs on the 3F2N [10] dataset: (a) depth maps and ground-truth surface normal maps; (b) error maps obtained using 3F2N (median filter); (c) error maps obtained using CP2TV; (d) error maps obtained using our proposed D2NT V3.

TABLE II
EVALUATION OF THE PROPOSED SNE USING FOUR DISCRETE LAPLACIAN FILTERS ON THE 3F2N DATASETS.

Filter	D2NT+DAG			D2NT+MNR		
Config	Easy	Medium	Hard	Easy	Medium	Hard
1D DLF	**1.19**	**4.87**	12.84	1.19	5.08	11.87
DLF-α	1.40	5.17	12.85	**0.79**	**4.80**	**9.86**
DLF-β	1.30	4.99	**12.46**	0.93	4.84	10.44
DLF-γ	1.36	5.05	13.03	1.36	5.05	13.03

to the basic depth-to-normal translator introduced in Sec. III-A as **D2NT**, the version that includes the DAG filter only as **D2NT V2**, and the version that includes both the DAG filter and the MNR module as **D2NT V3**.

Accurately determining surface normals from real-world range sensor data is infeasible due to the presence of noise. Although public datasets, such as NYUv2 [19] and DIODE [20], provide surface normal "ground truth", it is often obtained through the interpolation of point sets into local planar surfaces, making the evaluation of SNEs with such "ground truth" unreliable. As a result, we conduct experiments on our previously published synthetic dataset [10].

A. Implementation Details and Evaluation Metrics

As discussed in Sec. III, local surface smoothness is computed through the convolution of the depth map with Laplacian kernels. To find the best convolution kernel, four DLFs are used, including 1D DLF (horizontal kernel: $[1, -2, 1]$, and vertical kernel: $[1, -2, 1]^\top$), DLF-α, DLF-β, and DLF-γ[3] [21]. The execution time of the four DLFs is comparable, as the optimization only occupies a minor portion of the overall process. As demonstrated in Table II, the best results on the 3F2N easy and medium datasets are achieved when using the 1D DLF for D2NT+DAG. Additionally, D2NT+MNR shows the best performance across all three 3F2N datasets when using the DLF-α for computing local surface smoothness. Therefore, we use the 1D DLF for the DAG filter and the DLF-β for the MNR module.

Moreover, to meet the real-time requirement, we simplified the implementation of our proposed MNR module. Specifically, when the level of discontinuity in the local surface is assessed to be low according to (11), we exclude the smoothness term in (18). Similarly, if a point is identified to be in and around discontinuities, we omit the data term and instead use the surface normal of the neighboring point with the highest smoothness.

Following [10], we use the average angular error e_{A} to quantitatively evaluate the performance of SNEs:

$$e_{\mathrm{A}} = \frac{1}{N}\sum_{k=1}^{N} \cos^{-1}\left(\frac{\langle \boldsymbol{n}_k, \hat{\boldsymbol{n}}_k \rangle}{||\boldsymbol{n}_k||_2 ||\hat{\boldsymbol{n}}_k||_2}\right), \tag{21}$$

[3] DLF-α: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, DLF-$\beta$: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, DLF-$\gamma$: $\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

TABLE III

SPEED, ACCURACY, AND TRADE-OFF COMPARISONS AMONG SOTA GEOMETRY-BASED SNES ON THE 3F2N DATASET.

Real-Time	Method	t (ms) ↓	e_A (degrees) ↓			π (degrees/kHz) ↓		
			Easy	Medium	Hard	Easy	Medium	Hard
N	PlaneSVD [13]	**393.69**	2.07	6.07	17.59	813.87	**2389.73**	6923.18
	PlanePCA [14]	631.88	2.07	6.07	17.59	1306.29	3835.59	11111.92
	VectorSVD [16]	563.21	2.13	6.27	18.01	1199.63	3529.11	10142.34
	AreaWeighted [16]	1092.24	2.20	6.27	17.03	2407.74	6843.56	18600.68
	AngleWeighted [16]	1032.88	1.79	5.67	13.26	1850.00	5855.62	13693.24
	SDA-SNE [15]	726.18	**0.68**	**4.38**	**8.10**	**493.8**	3180.67	**5882.06**
Y	SNE-RoadSeg [3]	7.92	2.04	6.28	16.37	16.16	49.74	129.65
	3F2N [10]	10.97	1.66	5.69	15.31	18.18	62.38	168.03
	CP2TV [11]	2.23	1.69	6.01	13.82	3.75	13.39	30.76
	D2NT (ours)	**1.82**	1.54	5.64	15.32	**3.05**	**10.25**	**27.84**
	D2NT V2 (ours)	7.80	1.19	4.87	12.84	8.44	34.67	91.33
	D2NT V3 (ours)	7.99	**0.89**	**4.78**	**9.86**	7.09	38.28	78.91

TABLE IV

COMPARISON BETWEEN 3F2N AND CP2TV WITH AND WITHOUT OUR PROPOSED MNR MODULE EMBEDDED.

Module	3F2N			CP2TV		
Config	Easy	Medium	Hard	Easy	Medium	Hard
w/o MNR	1.66	5.69	15.32	1.69	6.02	13.82
w/ MNR	**0.82**	**4.89**	**10.33**	**0.91**	**4.80**	**9.86**
Improvement	50.7%	14.0%	32.5%	40.8%	15.0%	35.6%

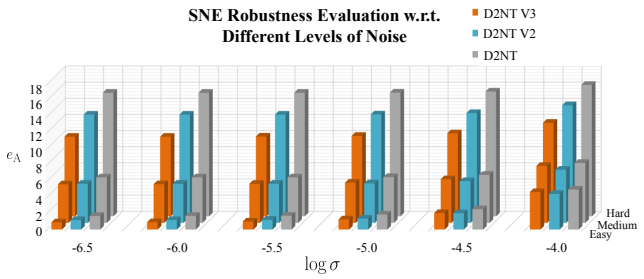

Fig. 4. Comparison among the three D2NT versions on the 3F2N datasets with different levels of Gaussian noise added.

where N is the number of valid pixels, and $\boldsymbol{n}_k$ and $\hat{\boldsymbol{n}}_k$ are the ground truth and estimated surface normals, respectively.

In addition to accuracy evaluation, we adopt the metric

$$\pi = e_A t \ (\text{degrees/kHz}) \tag{22}$$

proposed in [10] to quantify the trade-off between efficiency and accuracy of a given SNE. A high-performing (fast and accurate) SNE achieves a low π score.

B. Performance Comparison

As shown in Table III, our proposed surface normal estimators demonstrate superior performance compared to all other SoTA SNEs. D2NT achieves the highest computational efficiency and the optimum trade-off between speed and accuracy, while D2NT V3 achieves the highest accuracy (the e_A scores achieved by D2NT are less than $1°$, $5°$, and $9°$ on the 3F2N easy, medium, and hard datasets, respectively). Furthermore, as illustrated in Fig. 3, our D2NT outperforms 3F2N and CP2TV, particularly in and around discontinuities.

We also conducted supplementary experiments to demonstrate the compatibility of our proposed MNR module with other depth-to-normal SNEs, as shown in Table. IV. When incorporating the MNR module with 3F2N and CP2TV, the quality of their estimated surface normals is greatly improved, with a drop in 3F2N's e_A scores by 51%, 14%, and 33% on the 3F2N easy, medium, and hard datasets respectively and a decrease in CP2TV's e_A scores by 41%, 15%, and 36% on the same datasets. These results suggest that our proposed MNR module can be utilized in conjunction with other depth-to-normal SNEs and serves as an effective back-end optimization technique to enhance surface normal estimation in and around discontinuities.

As the used synthetic datasets are clean, we further evaluate the robustness of our methods in the presence of random Gaussian noise on the same datasets. As shown in Fig. 4, all three D2NT versions are stable with respect to different levels of Gaussian noise, and D2NT V3 is the most robust compared to the other two versions. In addition, it can be observed that our methods exhibit greater stability on the 3F2N medium and hard datasets, as compared to the 3F2N easy dataset. This is likely due to the added discontinuities caused by the Gaussian noise on the 3F2N easy dataset.

V. CONCLUSION

This paper presented an end-to-end depth-to-normal translator, a discontinuity-aware gradient filter, and an MRF-based surface normal refinement module. Extensive experimental results demonstrate that 1) our proposed depth-to-normal translator achieves the fastest execution speed and the best balance between computational efficiency and accuracy, and 2) the discontinuity-aware gradient filter and MRF-based surface normal refinement module can further improve its performance in and around discontinuities. Furthermore, our proposed MRF-based surface normal refinement module is also compatible with other depth-to-normal SNEs.

REFERENCES

[1] Y. Li *et al.*, "Structure-SLAM: Low-drift monocular SLAM in indoor environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6583–6590, 2020.

[2] Z. Liu *et al.*, "LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2831–2840.

[3] R. Fan *et al.*, "SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 340–356.

[4] H. Wang *et al.*, "Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10 750–10 760, 2022.

[5] H. Wang *et al.*, "Applying surface normal information in drivable area and road anomaly detection for ground mobile robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2706–2711.

[6] R. Fan *et al.*, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Transactions on Image Processing*, vol. 29, pp. 897–908, 2019.

[7] H. Wang *et al.*, "SNE-RoadSeg+: Rethinking depth-normal translation and deep supervision for freespace detection," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1140–1145.

[8] X. Qi *et al.*, "GeoNet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 283–291.

[9] X. Qi *et al.*, "GeoNet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 969–984, 2020.

[10] R. Fan *et al.*, "Three-filters-to-normal: An accurate and ultrafast surface normal estimator," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5405–5412, 2021.

[11] Y. Nakagawa *et al.*, "Estimating surface normals with depth image gradients for fast and accurate registration," in *2015 International Conference on 3D Vision (3DV)*. IEEE, 2015, pp. 640–647.

[12] C. Wang *et al.*, "Comparison of local plane fitting methods for range data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001.

[13] K. Klasing *et al.*, "Realtime segmentation of range data using continuous nearest neighbors," in *2009 International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2431–2436.

[14] K. Jordan *et al.*, "A quantitative evaluation of surface normal estimation in point clouds," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 4220–4226.

[15] N. Ming *et al.*, "SDA-SNE: Spatial discontinuity-aware surface normal estimation via multi-directional dynamic programming," in *2022 International Conference on 3D Vision (3DV)*, 2022, pp. 486–494.

[16] K. Klasing *et al.*, "Comparison of surface normal estimation methods for range sensing applications," in *2009 International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3206–3211.

[17] S. Jin *et al.*, "A comparison of algorithms for vertex normal computation," *The Visual Computer*, vol. 21, no. 1, pp. 71–82, 2005.

[18] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," *Unpublished manuscript*, vol. 46, 1971.

[19] N. Silberman *et al.*, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 746–760.

[20] Vasiljevic *et al.*, "DIODE: A Dense Indoor and Outdoor DEpth dataset," *CoRR*, 2019.

[21] M. Wardetzky, "Discrete laplace operators," *An Excursion Through Discrete Differential Geometry: AMS Short Course, Discrete Differential Geometry, January 8-9, 2018, San Diego, California*, vol. 76, p. 1, 2020.