A Scale-space Approach for Surface Normal Vector Estimation from Depth Maps

Diclehan Ulucan[1*], Oguzhan Ulucan[1] and Marc Ebner[1]

[1*]Computer Science, University of Greifswald, Walther-Rathenau-Str., Greifswald, 17489, Mecklenburg-Vorpommern, Germany.

*Corresponding author(s). E-mail(s): diclehan.ulucan@uni-greifswald.de;
Contributing authors: oguzhan.ulucan@uni-greifswald.de;
marc.ebner@uni-greifswald.de;

Abstract

Surface normal vectors provide cues about the local geometric features of the scene which are utilized in many computer vision and computer graphics applications. Thus, the estimation of surface normals by utilizing structured range sensor data is an important research field. Thereupon, we propose a learning-free algorithm to estimate the surface normal vectors from depth maps. Our simple yet effective method relies on computations carried out in scale-space. Our main idea is to estimate the surface normals which cannot be properly computed in the finest scale from the coarser scales. Our method can estimate the surface normals even for images included in datasets that have challenging characteristics such as noisy real-world data or significantly large planar regions that either have a small or no gradient change. We analyze our algorithm's performance by utilizing 5 benchmarks, namely, the MIT-Berkeley Intrinsic Images dataset, the New Tsukuba Dataset, the SceneNet RGB-D dataset, the IID-NORD dataset, and the NYU Depth Dataset V2, and by using 2 different evaluation strategies. According to the experimental results, our method can estimate surface normals efficiently without requiring neither complex computations nor huge amounts of data.

1 Introduction

The surface normal vectors help us to measure the angle between the incoming light ray and the fragment, i.e. the point the ray hits the object [1]. For a differentiable manifold, the

[1]This is the preprint version of the article published in SN Computer Science which is accessible via https://link.springer.com/article/10.1007/s42979-024-03098-4 (https://doi.org/10.1007/s42979-024-03098-4).

normal space at a certain point consists of a set of vectors that are orthogonal to the tangent space at this point. We utilize the normals in computer graphics to find the orientation of a surface with respect to the light source and to determine the orientation of the surface vertices. Also, surface normal vectors serve as local descriptors in many applications such as terrain mapping, navigation, object segmentation, and 3D object recognition by providing us the light direction, curvature, and shape of the object [1–4]. They can either be used directly or as auxiliary information. Due to their wide range of usability in computer vision tasks, estimating the surface normal vectors accurately with a low computational cost is substantial.

As the requirement of utilizing surface normals in different tasks increases, the efficient computation of normal vectors becomes more desired. Over the years both traditional and data-dependent algorithms have been proposed in the field of surface normal estimation [4–16]. Traditional methods usually make use of point clouds, and depth maps or disparity maps [3, 9]. Point clouds are mostly unstructured and distorted by noise, therefore feature extraction can become troublesome and procedures with high computational cost might be needed [4, 17]. On the other hand, depth maps contain organized sensor measurements and have a close geometric relationship with surface normals. Thus, they have gained more attention in this field. Alongside a distance measurement, data-dependent models may also use an RGB input image to estimate the surface normals. Recently, these models are widely used for the task of surface normal estimation, yet they have certain shortcomings. They require a high amount of unbiased data to estimate the surface normals accurately, their performance can be affected by adversarial attacks, providing a test image having a small difference with respect to their training set can decrease their performance, and usually, their computational cost is higher than traditional algorithms due to their training phase [4, 18–21]. Furthermore, in the field of normal estimation, many benchmarks are collected by using certain devices whose sensor quality affects the ground truth information, i.e. the measured depth data can be incomplete and noisy [22–24]. In several studies, the missing information is filled by utilizing various techniques [3]. Hence, the labeled data used in the training phase of data-dependent methods can be biased, and even inaccurate which may be a reason why data-dependent models still perform poorer than desired, i.e. generally, the mean proportion of good pixels is less than 80% [4, 10, 11].

Based on these observations, recently, we introduced a learning-free simple yet effective algorithm that estimates the surface normals from both complete and incomplete depth maps [25]. While designing our method, we exploited the close geometric relationship between depth maps and surface normals, and between the shading element of images and surface normals. Our algorithm makes use of the scale-space to provide more accurate estimations, and to deal with large planar regions and noisy measurements. In our previous study, we investigated the effectiveness of our method by reporting the performance of our technique on 2 benchmarks. In this study, we extend our previous work by providing a deeper analysis and by presenting results on 3 additional datasets. We utilize challenging benchmarks which contain significantly large planar regions and noisy real-world depth measurements which are features that were not investigated in our previous work.

This paper is organized as follows. In Sec. 2 we provide a brief literature review on geometry-based surface normal estimation methods. In Sec. 3 we introduce the proposed method. In Sec. 4 we present our experimental setup and in Sec. 5 we discuss the results. In Sec. 6 we conclude our work with a brief summary.

2 Related Work

Over the decades, several surface normal estimation algorithms based on distinct approaches have been put forward [5, 13]. We provide a brief literature review on geometry-based algorithms which we utilize in our experiments for comparison.

Surface normals and depth maps have a close geometric relationship. The differentiation of depth maps provides us with surface normals. Thereupon, a straightforward method, which we call *baseline*, to estimate the normals of a scene is to compute the pixel-wise difference of the consecutive neighboring pixels in the depth map, and to assume that the z-axis points to the camera (Eqn. 1).

$$n_{i_x} = p_{(i+1)_x} - p_{i_x} \qquad n_{i_y} = p_{(i+1)_y} - p_{i_y} \qquad n_{i_z} = -1 \tag{1}$$

where, $\mathbf{n_i} = [n_{i_x}, n_{i_y}, n_{i_z}]$ is the surface normal, $p_{(i+1)}$ is the neighboring pixel of our pixel of interest p_i, and x and y are the horizontal and vertical directions, respectively.

The local neighborhood of pixels is frequently exploited in surface normal estimation algorithms [5]. The averaging-based techniques make use of the neighboring relationships as follows;

$$\mathbf{n}_i = \frac{1}{k} \sum_{j=1}^{k} w_{i_j} \frac{[h_{i_j} - p_i] \times [h_{i_{j+1}} - p_i]}{|[h_{i_j} - p_i] \times [h_{i_{j+1}} - p_i]|} \tag{2}$$

where, h_{i_j} and $h_{i_{j+1}}$ are neighbours of p_i, w_{i_j} is the weight, which is 1 in the classical method, and $\times$ represents the cross-product operation.

In the angle-weighted averaging technique, w_{i_j} is the angle between the crossed vectors which can be calculated as follows;

$$w_{i_j} = cos^{-1}\left(\frac{\langle h_{i_j} - p_i,\ \ h_{i_{j+1}} - p_i \rangle}{|h_{i_j} - p_i||h_{i_{j+1}} - p_i|}\right). \tag{3}$$

In the area-weighted averaging technique, the normal of each triangle is weighted according to the magnitude of its area as follows;

$$w_{i_j} = \frac{1}{2}\left([h_{i_j} - p_i] \times [h_{i_{j+1}} - p_i]\right). \tag{4}$$

Aside from averaging methods, interpolation techniques are also used to estimate the surface normals. In the bicubic interpolation algorithm, the surface normals are estimated by applying bicubic interpolation on the depth map and utilizing quadratic extrapolation on the boundaries. Subsequently, the diagonal vectors are acquired and crossed to estimate surface normals [26].

There are also other algorithms that utilize optimization methods and filters to compute the surface normals. The SIRFS algorithm formulates an optimization problem to estimate the shading, illumination, reflectance, depth map, surface normals, and shape from a single masked image [6]. The filter-based 3F2N algorithm utilizes a horizontal and a vertical gradient filter, and a mean/median filter to estimate the surface normals [4]. The 3F2N method requires the image principal point and the camera focal lengths as priors.

3 Proposed Method

We introduce a learning-free surface normal vector estimation algorithm that carries out computations in scale-space whose effectiveness has been proven in different studies related to depth map and surface normal vectors estimation [12–14, 27–30]. Contrary to these methods, we do not use any optimization methods and data-dependent models or perform complex computations. In the following, we present our simple yet effective algorithm that estimates the surface normals from complete depth maps.

Our algorithm performs surface normal estimation in scale-space to take into account possible planar regions in the depth maps, fine details, and sharp edges. We determine the number of levels in the image pyramid adaptively by considering the fact that local information degrades significantly in the coarser scales of the pyramid [31, 32]. Since we would like to preserve local information which is important for estimating the surface normals, we do not use the coarser scales in our algorithm. After practical analysis, we determined that utilizing half of the number of possible levels that can be reached in scale-space allows us to respect local information while preventing high computational costs.

We utilize the classical averaging method to calculate the surface normal vectors [33]. We assume that a depth map is formed by a set of d points $P = \{p_1, p_2, ..., p_d\}$, $p_i \in \mathbb{R}^2$. Then, we calculate the surface normal vector $\mathbf{n}_i = [n_{i_x}, n_{i_y}, n_{i_z}]$ for a certain pixel p_i, by using k triangles created by the spatially closest neighbouring pixels $Q = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}$, $q_{i_k} \in P$, $q_{i_k} \neq p_i$ (Eqn. 5). For each pixel p_i we utilize 4 triangles to calculate the surface normal. However, k may decrease when we estimate the surface normals on the boundaries of the depth map, and when Q contains non-informative elements.

$$\mathbf{n}_i = \frac{1}{k} \sum_{j=1}^{k} [q_{i_j} - p_i] \times [q_{i_{j+1}} - p_i] \tag{5}$$

where, $q_{i_{j+1}}$ is the neighbouring pixel in the counter-clockwise direction of q_{i_j}, and $\times$ demonstrates the cross-product operation. It is worth mentioning that each $\mathbf{n_i}$ is normalized by its Frobenius norm $\mathbf{n}_i = \mathbf{n}_i / \|\mathbf{n}_i\|$.

During surface normal estimation, it is important to respect image regions containing sharp depth changes and edges. Abrupt depth changes may cause ambiguities in these challenging image regions since multiple distinct manifolds might be considered while computing the surface normals [7]. In our algorithm, we calculate the surface normals at each level s only in areas where the measured distance changes slightly. Thereby, we avoid ambiguities that arise due to any noticeable change in depth information. In other words, we obtain smooth transitions at edges and regions having sharp depth changes. For a pixel p_i, we compute the surface normal $\mathbf{n}_i$ at any level only when the depth changes below a certain threshold and does not equal zero (Fig 1). We practically determined that for the finest scale a threshold of 0.9 is sufficient, whereas this value should be increased 4 times at each consecutive coarser scale to compute the surface normals accurately. The reason behind this parameter selection procedure can be explained by the nature of scale space, where 1 pixel is represented by 4 pixels in each consecutive finer scale. We provide our investigation on the threshold in Sec. **??**.

After computing the surface normals at each level, we need to look for the surface normals that could not be calculated at the finest scale. For a pixel whose surface normal could not be computed at the finest scale, we seek to find the corresponding value at a level where the

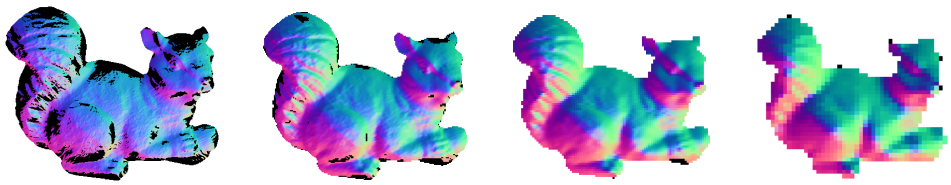

Fig. 1: The surface normal estimates at different scales are presented. The black areas correspond to the regions that do not satisfy the threshold. (Left-to-right) Surface normal estimates at the first scale, second scale, third scale, and fourth scale.

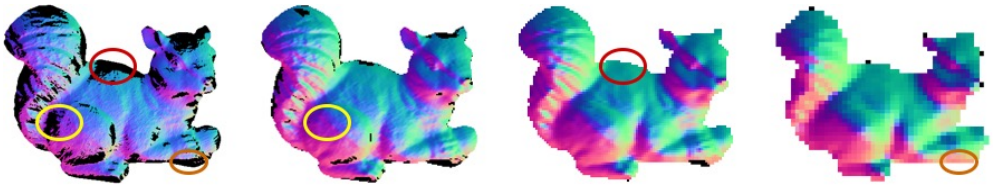

Fig. 2: The illustration of finding corresponding surface normals from different scales. The areas enclosed by yellow, red, and orange ellipses are filled from the second, third, and fourth scales, respectively.

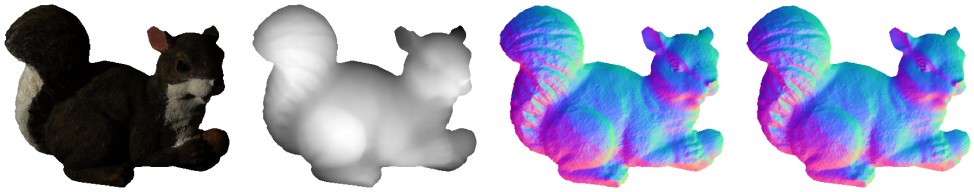

Fig. 3: Estimation of normals from a depth map. (Left-to-right) Scene, depth map, ground truth, and estimated surface normals.

normal could be calculated. We illustrate this procedure in Fig.2. We fill the areas which are encircled by a yellow ellipse by benefiting from the estimations at the second scale, while we recover the areas enclosed by an orange ellipse by utilizing the estimations at the coarsest level. In rare cases, we might have a pixel p_i whose surface normal could not be estimated at any level. For such pixels, we apply an averaging operation in the pixel's 3×3 neighborhood and assume that the calculated vector corresponds to $\mathbf{n}_i$.

As a final step, in order to refine our estimations, we perform a Gaussian smoothing operation with a small standard deviation σ obtained adaptively based on the image resolution. A proper choice is $\sigma = 0.08\gamma$ where $\gamma = max\{w, h\}/100$, and w and h are the width and height of the image, respectively. An example of estimating the surface normals via our method is shown in Fig. 3.

In case the measured sensor data includes missing pixels, we slightly modify our algorithm to avoid any ambiguities caused by the missing regions. In order to handle the missing pixels, one can fill the depth map by using various techniques, i.e., interpolation, before using

the data to estimate the normal vectors. However, filling the missing regions might corrupt the depth data [25]. Hence, utilizing the pre-processed measurements might prevent us to obtain accurate local descriptors of the scene. Therefore, in order to estimate the surface normal vectors without pre-processing the measured depth maps, we slightly modify our technique by taking the direct link between the shading element and the surface normal vectors of the scene into account [34]. Like the surface normals, the shading component contains information on the orientation of the light source and it can be obtained as follows;

$$\mathcal{S}_i = \phi_i \cdot \langle \mathbf{n}_i, \mathbf{L}_i \rangle \tag{6}$$

where ϕ is the light intensity, $\mathbf{L}$ is the light direction vector, and $\langle . \rangle$ represents the inner product.

As given in the study of Bonneel *et al.* [35], we assume that the grayscale illumination of the scene is the shading component $\mathcal{S}$ (Eqn. 7). We use this assumption since designing a method to form the shading image is outside the scope of this work.

$$\mathcal{S} = \sqrt{(r + g + b)/3} \tag{7}$$

where, r, g, b are the red, green, and blue color channels, respectively.

To estimate the surface normals of incomplete depth maps, first of all, we compute the normals at each level by utilizing Eqn. 5. Then, we discard the areas containing missing depth measurements. In order to obtain the estimations for the missing regions, we guide our computations in the direction of the gradient change of the shading element of the scene. We determine the gradient direction by utilizing the gradient angle [36].

For each missing pixel in the depth map, we estimate the surface normal by averaging the neighboring normals sharing the same gradient flow in its 3×3 neighborhood (Table 1). If there exists only one informative neighbor, we directly take it as the surface normal estimate. In case both neighbors are non-informative, we carry out the normal estimation at other scales. Subsequently, we perform the same procedure we applied to the complete depth maps. Lastly, we apply a Gaussian smoothing operation where $\sigma = 0.16\gamma$.

In order to carry out an analysis for the incomplete depth map scenario, in our previous work, we modified the MIT-Berkeley Intrinsic Images dataset [37] by removing a certain amount of pixels from the scenes, and we created 2 different sets, namely, easy and hard set [25]. While for the hard case, regions up to 30% of the pixels in the object are removed, for the easy case we discard regions up to 5% of the pixels. We provide examples of computing the surface normals in incomplete depth maps in Fig.4.

Table 1: Determining the gradient flow according to the angle between the neighboring pixels. x and y specify the location of the image elements.

Angles		**Neighboring spatial locations**	
$[0°, 45°)$	$[180°, 225°)$	$(x-1, y+1)$	$(x+1, y-1)$
$[45°, 90°)$	$[225°, 270°)$	$(x-1, y)$	$(x+1, y)$
$[90°, 135°)$	$[270°, 315°)$	$(x-1, y-1)$	$(x+1, y+1)$
$[135°, 180°)$	$[315°, 360°]$	$(x, y-1)$	$(x, y+1)$

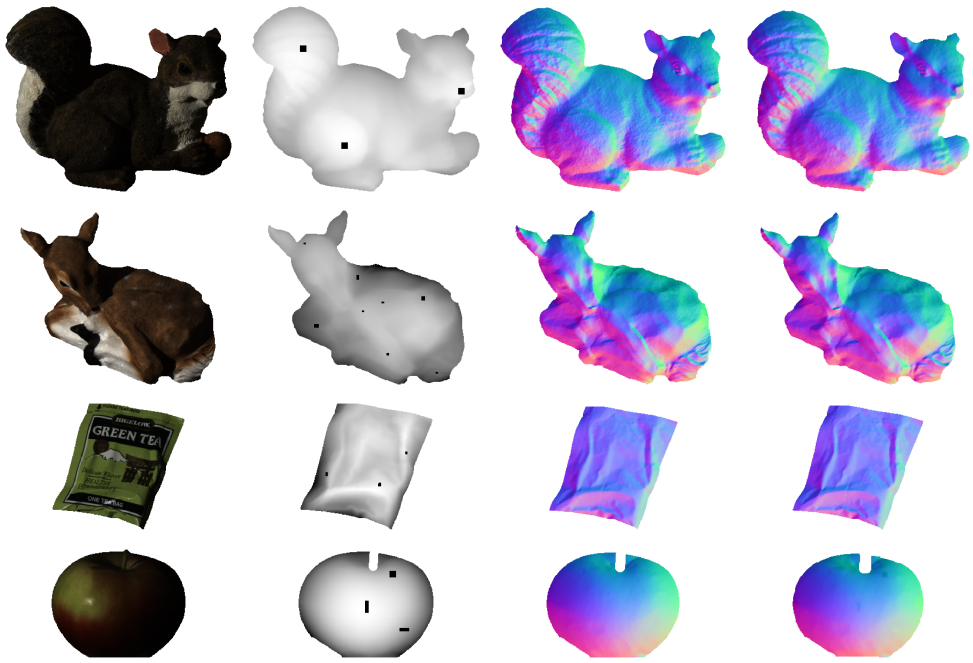

Fig. 4: Estimating the surface normals from an incomplete depth map. (Left-to-right) Shading, incomplete depth map, ground truth, and estimated surface normals.

4 Experimental Setup

In this section, we explain our experimental setup, while providing a brief summary of the datasets, and the evaluation strategies we utilize to discuss the performance of the proposed method.

4.1 Datasets

Even though surface normals are widely used in computer vision tasks and they have been extensively studied over the decades, investigating the efficiency of the designed algorithms is still troublesome since accurate ground truths for surface normals are lacking in many existing benchmarks [38]. Therefore, statistically evaluating the algorithms is not always possible which is one the reasons why researchers also provide visual comparisons in this field.

In our previous work, to investigate our method's performance, we utilized 2 benchmarks, namely, the MIT-Berkeley Intrinsic Images dataset [6], and the New Tsukuba dataset [39]. In this work, we are extending our discussion by demonstrating the performance of the proposed method on 3 additional datasets. In this subsection, we briefly introduce all the benchmarks we adopt for evaluating our method (Fig. 5).

Fig. 5: The benchmarks utilized in this work. (Top-to-bottom) Scene and depth map. (Left-to-right) The MIT-Berkeley Intrinsic Images dataset, the New Tsukuba Dataset, the SceneNet RGB-D dataset, the IID-NORD dataset, and the NYU Depth Dataset V2.

The MIT-Berkeley Intrinsic Images Dataset

The MIT-Berkeley Intrinsic Images dataset [6] is the extended version of the MIT Intrinsic Images dataset [37]. This benchmark consists of 20 different single-masked objects and contains ground truths of distinct intrinsic images including surface normals and depth maps.

The New Tsukuba Dataset

The New Tsukuba Dataset contains video sequences of real-world-like scenes that are rendered by computer graphics [39]. This dataset consists of 1800 frames of stereo pairs. The ground truth disparity maps are provided for each scene, while the ground truth surface normals are not included in the benchmark.

The SceneNet RGB-D Dataset

The SceneNet RGB-D dataset consists of 5 million of photorealistic indoor scenes generated by computer graphics [40, 41]. Alongside the RGB scenes, for each scene, the dataset also contains various intrinsics including the ground truth depth maps. In order to evaluate the performance of the algorithms, we utilize the validation set.

The IID-NORD Dataset

The IID-NORD dataset is a large-scale dataset which we created in one of our previous studies to assist the intrinsic image decomposition studies since in the field of intrinsic image decomposition, finding a benchmark that provides the actual pixel-wise ground truths of various intrinsics can be troublesome [38]. IID-NORD contains a total of 128000 synthetic indoor scenes, and provides the ground truth surface normals. In order to validate the algorithms, we use a subset of IID-NORD.

NYU Depth Dataset V2

The NYU Depth Dataset V2 is one of the well-known benchmarks for geometric computer vision problems [23]. This dataset contains 1449 RGB-D images obtained from 464 various real indoor scenes. The dataset does not provide the ground truths of surface normals. In our experiments we used the labeled set of this dataset.

4.2 Evaluation Metrics

Investigating the performance of the algorithms with a suitable error metric is crucial to report their efficiency and their shortcomings. It is known that the reported performance of an algorithm may differ with the usage of different evaluation strategies, hence utilizing several error metrics is beneficial [42]. In this part, we briefly introduce the evaluation strategies that we use to benchmark our algorithm.

Geodesic Distance

The geodesic distance is defined as the length of the shortest path between the points on the surfaces along the manifold. The geodesic distance is independent of the observer's viewing angle since it is responsive to small topology changes [43, 44]. The lower the geodesic distance the higher the similarity between the ground truth and the estimated surface normals. The geodesic distance $GDIS$ between the ground truth $\mathbf{n}_{gt}$ and the estimated surface normal $\mathbf{n}_{est}$ is computed as follows;

$$GDIS = \frac{1}{d}\sum_{i=1}^{d} cos^{-1}\left(\langle \mathbf{n}_{gt_i} \cdot \mathbf{n}_{esti} \rangle\right). \tag{8}$$

Root Mean Square Error

The root mean square error is a commonly used evaluation metric in the field of intrinsic image decomposition [45]. A lower root mean square error indicates better results. The root mean square error ($RMSE$) between $\mathbf{n}_{gt}$ and $\mathbf{n}_{est}$ can be calculated as follows;

$$RMSE = \sqrt{\frac{1}{d}\sum_{i=1}^{d} \left|\mathbf{n}_{gt_i} - \mathbf{n}_{esti}\right|^2}. \tag{9}$$

5 Discussion

We compare our algorithm with the following geometry-based surface normal estimation algorithms; baseline, Area-Weighted [5], Angle-Weighted [5], bicubic interpolation [26], SIRFS [6], and 3F2N [4]. We provide both statistical and visual comparisons on the MIT-Berkeley Intrinsic Images dataset and IID-NORD dataset, while we provide only visual evaluations on the New Tsukuba Dataset, SceneNet RGB-D dataset, and NYU Depth Dataset V2. We prefer this evaluation approach since the former datasets include ground truth surface normals, while the latter benchmarks do not. Moreover, for the 3F2N method which requires the camera specifics as input, we have used fixed parameters during the evaluation on the MIT-Berkeley Intrinsic Images dataset that does not provide the camera specifics. For more complex scenes the use of fixed camera parameters would affect the outcomes significantly. That is why we do not consider the 3F2N algorithm for the comparisons on more complex datasets. Also, we evaluate the SIRFS algorithm only on the MIT-Berkeley Intrinsic Images dataset since it requires single masked objects as input.

As shown in Table 2, our approach achieves the lowest errors for all metrics on the MIT-Berkeley Intrinsic Images dataset. This statistical achievement also coincides with the visual

Table 2: Statistical results on MIT-Berkeley Intrinsic Images dataset. The best scores are highlighted.

	GDIS					
	Min.	**Mean**	**Med.**	**B.25%**	**W.25%**	**Max.**
Baseline	0.996	0.999	0.999	0.998	1.000	1.000
Angle-Weighted	0.095	0.163	0.155	0.112	0.223	0.262
Area-Weighted	0.095	0.163	0.155	0.111	0.223	0.262
Bicubic Interpolation	0.148	0.256	0.246	0.172	0.359	0.504
SIRFS	0.145	0.256	0.242	0.166	0.366	0.438
3F2N	0.218	0.379	0.377	0.271	0.494	0.563
Proposed without scale-space	0.096	0.163	0.155	0.112	0.223	0.263
Proposed with scale-space	**0.091**	**0.160**	**0.153**	**0.108**	**0.220**	**0.259**
	RMSE					
	Min.	**Mean**	**Med.**	**B.25%**	**W.25%**	**Max.**
Baseline	0.783	0.968	0.953	0.854	1.094	1.270
Angle-Weighted	0.051	0.063	0.062	0.053	0.073	0.083
Area-Weighted	0.046	0.056	0.055	0.048	0.066	0.073
Bicubic Interpolation	0.189	0.342	0.337	0.238	0.447	0.516
SIRFS	0.130	0.246	0.241	0.164	0.324	0.333
3F2N	0.400	0.490	0.500	0.414	0.572	0.626
Proposed without scale-space	0.046	0.056	0.055	0.048	0.066	0.073
Proposed with scale-space	**0.037**	**0.049**	**0.048**	**0.039**	**0.061**	**0.064**

comparisons in Fig. 6. Our method is able to preserve the small details on edges where the gradient changes sharply. On this dataset, the performance difference between our algorithm, and its version where we do not employ the scale-space is statistically low, however, the advantage of the scale-space is clearly observable in the outcomes of more complex datasets containing larger planar regions.

After evaluating the performance on a dataset containing single objects, we analyze our method's effectiveness on a more complex synthetic dataset, namely IID-NORD. We provide the statistical outcomes in Table 3. For this table it is critical to note that the scores have to be considered together with the percentage of surface normals that the algorithm is able to predict, i.e. an algorithm might achieve a better score than others but estimate only half of the surface normals present in the scene. The missing normals in the estimations can be explained by the large planar regions where the algorithms are sometimes unable to compute the surface normals for each pixel of the scene as shown in Fig.7. Our proposed method is able to compute all the surface normals in the scenes, while achieving the lowest RMSE scores. In terms of GDIS, we do not present the lowest errors but we are able to estimate all the surface normals, while the methods achieving lower errors estimate the normals approximately for half of the pixels present in the image.

After providing statistical analysis on two datasets, we present further results of our method by providing visual comparisons on datasets containing either real-world scenes or real-world-like synthetic images. In Fig. 8, we present different scenes from the New Tsukuba Dataset which contrary to the MIT-Berkeley Intrinsic Images dataset contains real-world-like scenes having large planar regions that are challenging for most algorithms. For instance, the Angle-Weighted method fails to estimate the surface normals at planar regions, whereas

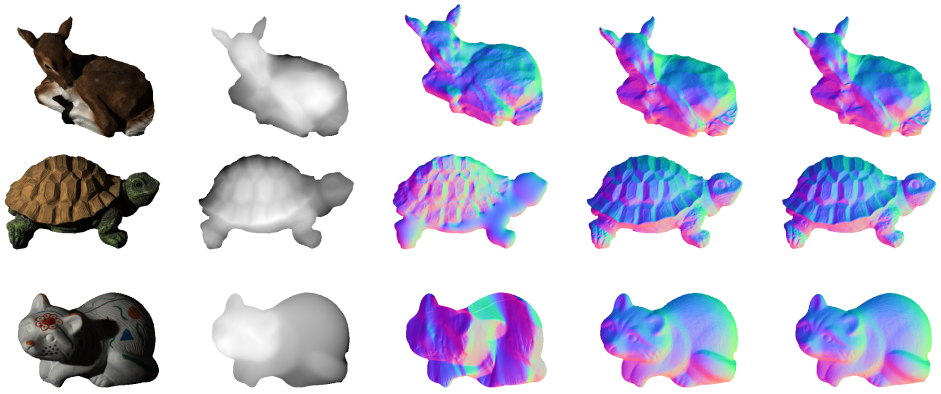

Fig. 6: Visual comparison on the MIT-Berkeley Intrinsic Images dataset. (Left-to-right) The scene, depth map, estimation of SIRFS, estimation of the proposed method, and the ground truth normals.

Table 3: Statistical results on the subset of the IID-NORD dataset. While analyzing this table it is important to consider the scores together with the percentage of filled pixels (Pix.%).

			GDIS				
	Pix.%	**Min.**	**Mean**	**Med.**	**B.25%**	**W.25%**	**Max.**
Baseline	99.53	0.972	0.994	0.999	0.977	0.999	0.999
Angle-Weighted	47.77	0.145	0.213	0.217	0.163	0.258	0.314
Area-Weighted	47.77	0.145	0.213	0.217	0.163	0.258	0.314
Bicubic Interpolation	100.00	0.529	0.651	0.657	0.545	0.724	0.734
Proposed without scale-space	52.01	0.138	0.289	0.289	0.159	0.399	0.433
Proposed with scale-space	100.00	0.263	0.321	0.315	0.284	0.366	0.403
			RMSE				
	Pix.%	**Min.**	**Mean**	**Med.**	**B.25%**	**W.25%**	**Max.**
Baseline	99.53	1.037	1.082	1.086	1.044	1.117	1.144
Angle-Weighted	47.77	0.195	0.281	0.305	0.221	0.324	0.374
Area-Weighted	47.77	0.195	0.281	0.305	0.221	0.324	0.374
Bicubic Interpolation	100.00	0.389	0.440	0.453	0.401	0.463	0.468
Proposed without scale-space	52.01	0.193	0.291	0.314	0.240	0.326	0.384
Proposed with scale-space	100.00	0.129	0.203	0.216	0.139	0.245	0.277

our algorithm provides accurate surface normals by benefiting from carrying out the computations in scale-space. If we have complex scenes rather than a single masked object, the nonuse of the scale-space causes discontinuities at edges and surface normals cannot be computed at planar regions. This observation is noteworthy since in the statistical results given in Table 2, the Area-Weighted, Angle-Weighted, and the proposed approach without scale-space provide similar scores as our algorithm, while in Fig. 8 we can clearly see the effectiveness of scale-space computations. In addition to the examples presented in Fig. 8, we provide a

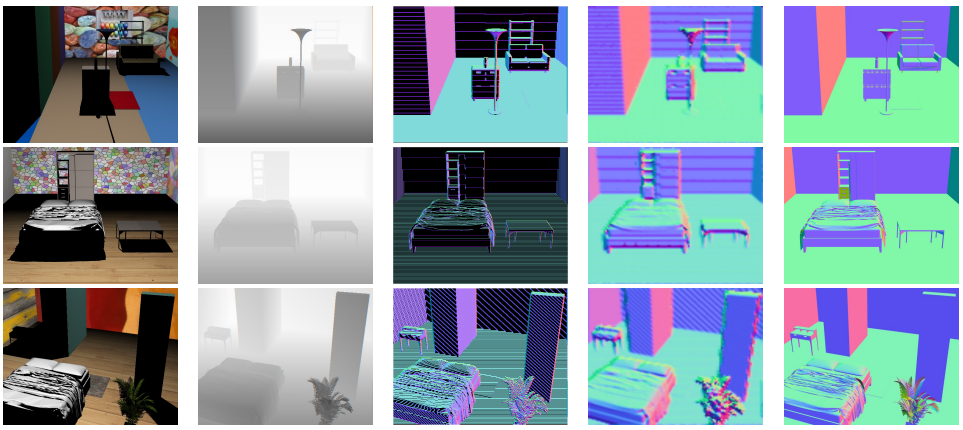

Fig. 7: Visual results on the IID-NORD dataset. (Left-to-right) Input scene, depth map, results of the Area-Weighted method, estimations of the proposed algorithm, and the ground truth surface normals.

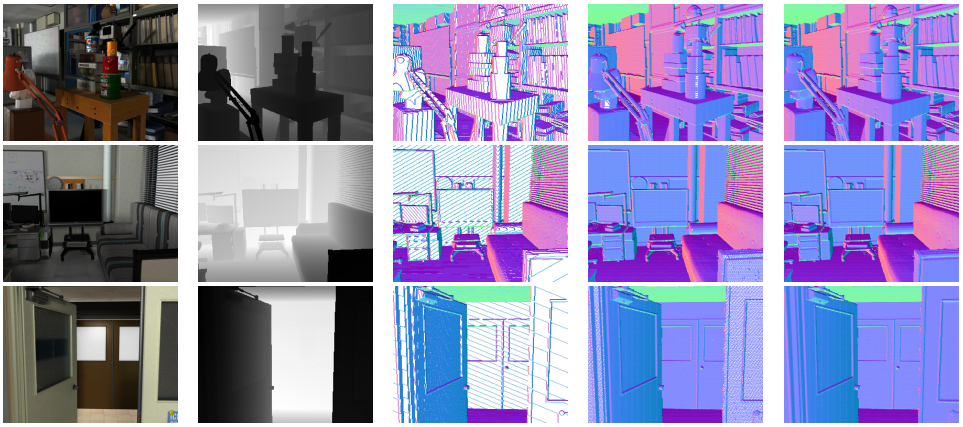

Fig. 8: Visual comparison on the New Tsukuba Dataset. (Left-to-right) The scene, depth map, and estimations of the Angle-Weighted method, the proposed algorithm without scale-space, and the proposed algorithm. The white regions present the pixels where surface normals could not be computed.

video sequence containing all the scenes of the New Tsukuba dataset which can be reached from the following page [46].

We provide our results on the SceneNet RGB-D dataset in Fig. 9. Like the New Tsukuba dataset, the SceneNet RGB-D dataset includes real-world-like synthetic scenes. On the first row of Fig. 9, we can see that the surface normals are estimated by the Angle-Weighted method at regions that appear to be planar in the scene, while the same method failed to compute the normals in planar areas in the New Tsukuba Dataset. The reason behind the difference of the outcomes might be the distinct rendering procedures applied during dataset

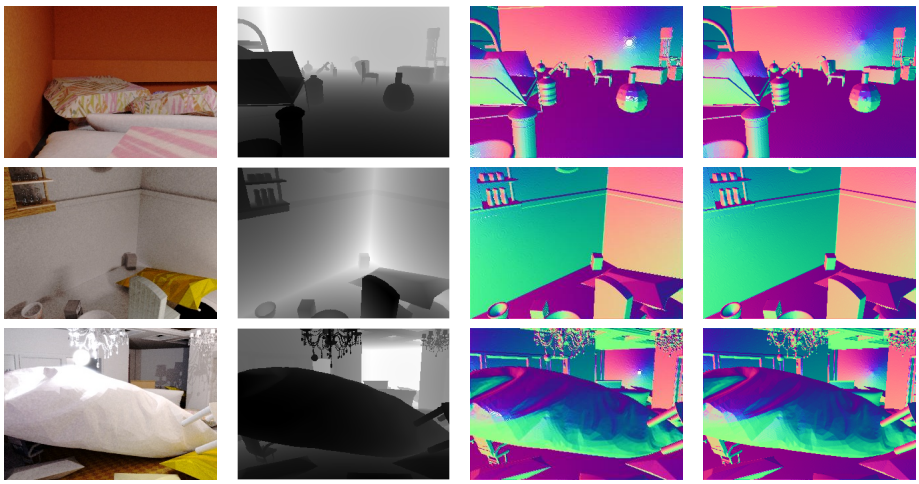

Fig. 9: Visual results on the SceneNet RGB-D dataset. (Left-to-right) Input scene, depth map, results of the Angle-Weighted method, and the proposed algorithm.

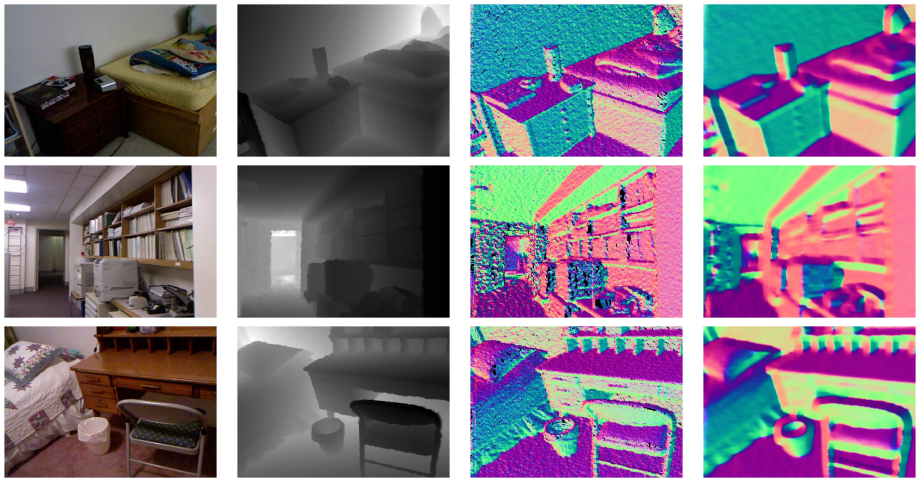

Fig. 10: Visual results on the NYU Depth Dataset V2. (Left-to-right) Input scene, depth map, results of the Area-Weighted method, and the proposed algorithm.

formation. As we can see from the estimated normals of the SceneNet RGB-D dataset, the depth maps contain small but frequent gradient changes even in areas that appear to be planar, i.e. back wall on the first row of Fig. 9. While the performance of the Angle-Weighted method increases on this dataset, it still faces difficulties, i.e. there are surface normals that could not be computed on the pillow and wall for the image on the third row of Fig. 9. On the other hand, our approach taking advantage of the scale-space computations provides accurate estimates without any missing regions.

In Fig. 10, we provide a visual analysis on the NYU Depth Dataset V2. As aforementioned this is a real-world dataset containing noisy depth measurements which challenges the algorithms during surface normal estimation. We observe that the Area-Weighted algorithm produces sharp edges yet it is severely affected by noise. On the other hand, our algorithm provides surface normals that do not contain severe ambiguities in planar regions arising due to noisy measurements, while greatly respecting the transitions between the manifolds. Obtaining entirely sharp estimations is a common challenge in the field of surface normal estimation, especially when noisy depth measurements are of interest. This challenge is observable even in the results of learning-based state-of-the-art studies requiring high amount of data and computationally expensive training phases [8, 14, 15, 47].

Lastly, we provide our investigation on the threshold setting. We have analyzed distinct threshold values by evaluating their performance on the MIT-Berkeley Intrinsic Images dataset based on the RMSE and GDIS scores. As given in Fig. 11, the errors do not significantly differ between the thresholds of 0.5 and 1.3. However, when we visually investigate the estimated surface normals based on distinct thresholds, we notice ambiguities in some of the outcomes. We observe that the lowest RMSE score obtained at the threshold value of 0.9 provides the best estimations. Therefore, we chose 0.9 as the threshold for the finest scale, while for each consecutive level, the threshold becomes 4 times larger. It is worth mentioning here that after making statistical investigations on the IID-NORD dataset, and carrying out visual analysis on the New Tsukuba Dataset, SceneNet RGB-D dataset, and NYU Depth Dataset V2, we determined that using approximately a 3 times larger threshold for these datasets results in better performance. We can explain the selection of a different threshold value for these datasets by two facts; *(i)* the measured depth range difference, and *(ii)* the complexity of the scenes. For the former we investigated the minimum and maximum depth measurements present in the datasets. While the depth range of the MIT-Berkeley Intrinsic Images dataset is between -327.87 and 286.77, the depth maps of the IID-NORD dataset are in integer numbers and their range is between 0 and 255, the New Tsukuba Dataset has a depth range between 23.63 and 1312.50, the SceneNet RGB-D dataset has a depth range between 0 and 10589, and the depth range of the NYU Depth Dataset V2 is between 713 and 9986. One could suggest to fit the depth information into a certain common range for all datasets, however this would distort the data, thus we do not prefer such an approach. For the latter we analyzed the depth difference between different objects in a scene, and during this analysis we noticed that the several scenes in the benchmarks (except of the MIT-Berkeley Intrinsic Images dataset) contain large depth measurement changes between objects which increases the required threshold. Since the MIT-Berkeley Intrinsic Images dataset contains only single objects, and the other datasets either provide depth maps in integer numbers or have a higher range, it is not surprising that for these datasets a larger threshold value is required.

6 Conclusion

Surface normals are local descriptors that enable us to extract features of the scenes. Therefore, they are utilized in various computer vision and computer graphics applications. Over the years many traditional and data-dependent algorithms have been proposed to estimate the surface normals, yet efficient low-cost methods are still needed in this field. We introduce a simple yet effective surface normal estimation algorithm that operates in scale-space. Our

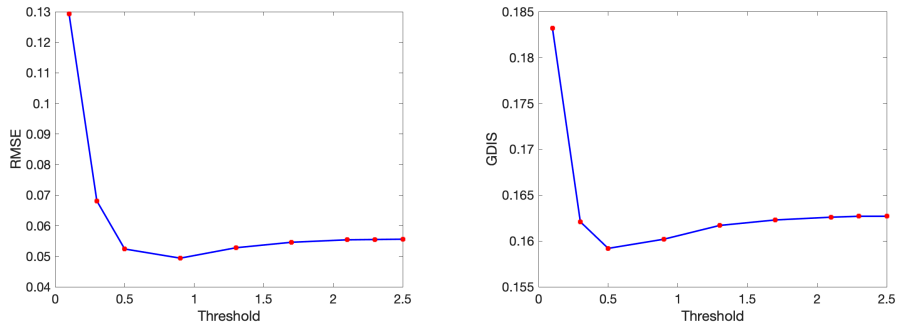

Fig. 11: The analysis of the threshold value. The best-performed parameter is selected based on the average RMSE and GDIS score on the MIT-Berkeley Intrinsic Images dataset.

method estimates the surface normals in the finest scale of the pyramid, then utilizes the estimations in coarser scales to fill the missing regions that could not be computed properly in the finest scale. By taking advantage of scale-space, our method is able to estimate surface normals even for scenes that contain large planar regions and noisy depth measurements. We provide in-depth analysis on 5 different benchmarks including challenging datasets such as the IID-NORD dataset and the NYU Depth Dataset V2. According to the experiments, our algorithm can compute the surface normals for both simple and complex scenes efficiently.

Declarations

Conflict of interest. The authors state that there are no conflicts of interest.

References

[1] Ebner, M.: Color Constancy, 1st Ed. Wiley Publishing, ISBN: 0470058299, Hoboken, NJ, USA (2007)

[2] Harms, H., Beck, J., Ziegler, J., Stiller, C.: Accuracy analysis of surface normal reconstruction in stereo vision. In: Intell. Vehicles Symp. Proc., Dearborn, MI, USA, pp. 730–736 (2014). IEEE

[3] Zhang, Y., Funkhouser, T.: Deep depth completion of a single RGB-D image. In: Conf. Comput. Vision Pattern Recognit., Salt Lake City, UT, USA, pp. 175–185 (2018). IEEE/CVF

[4] Fan, R., Wang, H., Xue, B., Huang, H., Wang, Y., Liu, M., Pitas, I.: Three-filters-to-normal: An accurate and ultrafast surface normal estimator. IEEE Robot. Automat. Letters **6**(3), 5405–5412 (2021)

[5] Klasing, K., Althoff, D., Wollherr, D., Buss, M.: Comparison of surface normal estimation methods for range sensing applications. In: Int. Conf. Robot. Automat., Kobe, Japan, pp. 3206–3211 (2009). IEEE

[6] Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. IEEE Trans. Pattern Anal. Mach. Intell. **37**(8), 1670–1687 (2014)

[7] Cao, J., Chen, H., Zhang, J., Li, Y., Liu, X., Zou, C.: Normal estimation via shifted neighborhood for point cloud. J. Comput. Appl. Math. **329**, 57–67 (2018)

[8] Chen, W., Xiang, D., Deng, J.: Surface normals in the wild. In: Int. Conf. Comput. Vision, Venice, Italy, pp. 1557–1566 (2017). IEEE

[9] Mitra, N.J., Nguyen, A.: Estimating surface normals in noisy point cloud data. In: Proc. Annu. Symp. Comput. Geometry, San Diego, CA, USA, pp. 322–328 (2003). ACM

[10] Li, B., Shen, C., Dai, Y., Van Den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In: Conf. Comput. Vision Pattern Recognit., Boston, MA, USA, pp. 1119–1127 (2015). IEEE

[11] Bansal, A., Russell, B., Gupta, A.: Marr revisited: 2D-3D alignment via surface normal prediction. In: Conf. Comput. Vision Pattern Recognit., Las Vegas, NV, USA, pp. 5965–5974 (2016). IEEE

[12] Li, K., Zhao, M., Wu, H., Yan, D.-M., Shen, Z., Wang, F.-Y., Xiong, G.: GraphFit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In: Eur. Conf. Comput. Vision, Tel Aviv, Israel, pp. 651–667 (2022). Springer

[13] Zeng, J., Tong, Y., Huang, Y., Yan, Q., Sun, W., Chen, J., Wang, Y.: Deep surface normal estimation with hierarchical RGB-D fusion. In: Conf. Comput. Vision Pattern Recognit., Long Beach, CA, USA (2019). IEEE/CVF

[14] Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Int. Conf. Comput. Vision, Santiago, Chile (2015). IEEE

[15] Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: Geonet: Geometric neural network for joint depth and surface normal estimation. In: Conf. Comput. Vision Pattern Recognit., Salt Lake City, UT, USA, pp. 283–291 (2018). IEEE/CVF

[16] Bae, G., Budvytis, I., Cipolla, R.: Irondepth: Iterative refinement of single-view depth using surface normal and its uncertainty. In: BMVC, London, UK (2022). BMVA Press

[17] Awwad, T.M., Zhu, Q., Du, Z., Zhang, Y.: An improved segmentation approach for planar surfaces from unstructured 3d point clouds. Photogrammetric Rec. **25**, 5–23 (2010)

[18] Lenssen, J.E., Osendorfer, C., Masci, J.: Deep iterative surface normal estimation. In: Conf. Comput. Vision Pattern Recognit., Virtual, pp. 11247–11256 (2020). IEEE/CVF

[19] Ulucan, O., Ulucan, D., Ebner, M.: Color constancy beyond standard illuminants. In: Int. Conf. Image Process., Bordeaux, France, pp. 2826–2830 (2022). IEEE

[20] Ulucan, O., Ulucan, D., Ebner, M.: BIO-CC: Biologically inspired color constancy. In: Brit. Mach. Vision Conf., London, UK, p. (2022). BMVA Press

[21] Ulucan, O., Ulucan, D., Ebner, M.: Block-based color constancy: The deviation of salient pixels. In: Int. Conf. Acoust. Speech Signal Process., Rhodes Island, Greece, pp. 1–5 (2023). IEEE

[22] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proc. Annu. ACM Symp. User Interface Softw. Technol., Santa Barbara, CA, USA, pp. 559–568 (2011). ACM

[23] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Eur. Conf. Comput. Vision, Florence, Italy, pp. 746–760 (2012). Springer

[24] Kwon, H., Tai, Y.-W., Lin, S.: Data-driven depth map refinement via multi-scale sparse representation. In: Conf. Comput. Vision Pattern Recognit., Boston, MA, USA, pp. 159–167 (2015). IEEE

[25] Ulucan, D., Ulucan, O., Ebner, M.: Multi-scale surface normal estimation from depth maps. In: Int. Conf. Image Process. Vision Eng., Prague, Czech Republic, pp. 47–56 (2023). INSTICC

[26] MATLAB: Surfnorm function. https://de.mathworks.com/help/matlab/ref/surfnorm.html (last access: 14.04.2024)

[27] Ioannou, Y., Taati, B., Harrap, R., Greenspan, M.: Difference of normals as a multi-scale operator in unorganized point clouds. In: Int. Conf. 3D Imag. Model. Process. Visualization Transmiss., Zurich, Switzerland, pp. 501–508 (2012). IEEE

[28] Saracchini, R.F.V., Stolfi, J., Leitão, H.C.G., Atkinson, G.A., Smith, M.L.: A robust multi-scale integration method to obtain the depth from gradient maps. Comput. Vision Image Understanding **116**(8), 882–895 (2012)

[29] Zhou, J., Huang, H., Liu, B., Liu, X.: Normal estimation for 3d point clouds via local plane constraint and multi-scale selection. Comput. Aided Des. **129**, 102916 (2020)

[30] Hsu, H., Su, H.-T., Yeh, J.-F., Chung, C.-M., Hsu, W.H.: SeqDNet: Improving missing value by sequential depth network. In: Int. Conf. Image Process., Bordeaux, France, pp. 1826–1830 (2022). IEEE

[31] Ulucan, O., Ulucan, D., Ebner, M.: Multi-scale block-based color constancy. In: Eur.

Signal Process. Conf., Helsinki, Finland, pp. 536–540 (2023). IEEE

[32] Ulucan, O., Ulucan, D., Ebner, M.: Multi-scale color constancy based on salient varying local spatial statistics. The Vis. Comput., 1–17 (2023)

[33] Gouraud, H.: Continuous shading of curved surfaces. IEEE Trans. Computers **100**(6), 623–629 (1971)

[34] Jeon, J., Cho, S., Tong, X., Lee, S.: Intrinsic image decomposition using structure-texture separation and surface normals. In: Eur. Conf. Comput. Vision, Zurich, Switzerland, pp. 218–233 (2014). Springer

[35] Bonneel, N., Kovacs, B., Paris, S., Bala, K.: Intrinsic decompositions for image editing. Comput. Graph. Forum **36**, 593–609 (2017)

[36] Karakaya, D., Ulucan, O., Turkan, M.: Image declipping: Saturation correction in single images. Digit. Signal Process. **127**, 103537 (2022)

[37] Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: Int. Conf. Comput. Vision, Kyoto, Japan, pp. 2335–2342 (2009). IEEE

[38] Ulucan, D., Ulucan, O., Ebner, M.: IID-NORD: A comprehensive intrinsic image decomposition dataset. In: Int. Conf. Image Process., Bordeaux, France, pp. 2831–2835 (2022). IEEE

[39] Martull, S., Peris, M., Fukui, K.: Realistic cg stereo image dataset with ground truth disparity maps. In: ICPR Workshop TrakMark2012, vol. 111, pp. 117–118 (2012)

[40] McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet RGB-D: 5m photorealistic images of synthetic indoor trajectories with ground truth. arXiv preprint arXiv:1612.05079 (2016)

[41] McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet RGB-D: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In: Int. Conf. Comput. Vision, Venice, Italy, pp. 2678–2687 (2017). IEEE

[42] Ulucan, D., Ulucan, O., Ebner, M.: Intrinsic image decomposition: Challenges and new perspectives. In: Int. Conf. Image Process. Vision Eng., Prague, Czech Republic, pp. 57–64 (2023). INSTICC

[43] Pizer, S.M., Marron, J.S.: Object statistics on curved manifolds. In: Statistical Shape Deformation Anal., pp. 137–164. Elsevier, ??? (2017)

[44] Antensteiner, D., Štolc, S., Pock, T.: A review of depth and normal fusion algorithms. Sensors **18**(2), 431 (2018)

[45] Fouhey, D.F., Gupta, A., Hebert, M.: Data-driven 3D primitives for single image understanding. In: Int. Conf. Comput. Vision, Sydney, NSW, Australia, pp. 3392–3399 (2013). IEEE

[46] Ulucan, D., Ulucan, O., Ebner, M.: Supplementary video for our algorithm. Video at https://grypstube.uni-greifswald.de/w/wAPLAX8PcuiiBF7hVw2S9w (2023)

[47] Chen, W., Xiang, D., Deng, J.: Supplementary materials for surface normals in the wild. In: Int. Conf. Comput. Vision, Venice, Italy, pp. 1557–1566 (2017). IEEE