Adaptive Surface Normal Constraint for Depth Estimation

Xiaoxiao Long\textsuperscript{1}  Cheng Lin\textsuperscript{1}  Lingjie Liu\textsuperscript{2}  Wei Li\textsuperscript{3}  
Christian Theobalt\textsuperscript{2}  Ruigang Yang\textsuperscript{3}  Wenping Wang\textsuperscript{1,4}  
\textsuperscript{1}The University of Hong Kong  \textsuperscript{2}Max Planck Institute for Informatics  \textsuperscript{3}Inceptio  \textsuperscript{4}Texas A\&M University

Abstract

We present a novel method for single image depth estimation using surface normal constraints. Existing depth estimation methods either suffer from the lack of geometric constraints, or are limited to the difficulty of reliably capturing geometric context, which leads to a bottleneck of depth estimation quality. We therefore introduce a simple yet effective method, named Adaptive Surface Normal (ASN) constraint, to effectively correlate the depth estimation with geometric consistency. Our key idea is to adaptively determine the reliable local geometry from a set of randomly sampled candidates to derive surface normal constraint, for which we measure the consistency of the geometric contextual features. As a result, our method can faithfully reconstruct the 3D geometry and is robust to local shape variations, such as boundaries, sharp corners and noises. We conduct extensive evaluations and comparisons using public datasets. The experimental results demonstrate our method outperforms the state-of-the-art methods and has superior efficiency and robustness\textsuperscript{1}.

1. Introduction

Estimating depth from a single RGB image, one of the most fundamental computer vision tasks, has been extensively researched for decades. With the recent advances of deep learning, depth estimation using neural networks has drawn increasing attention. Earlier works [6, 22, 9, 34, 31] in this field directly minimize the pixel-wise depth errors, of which results cannot faithfully capture the 3D geometric features. Therefore, the latest efforts incorporate geometric constraints into the network and show promising results.

Among various geometric attributes, surface normal is predominantly adopted due to the following two reasons. First, surface normal can be estimated by the 3D points converted from depth. Second, surface normal is determined by a surface tangent plane, which inherently encodes the local geometric context. Consequently, to extract normals from depth maps as geometric constraints, previous works propose various strategies, including random sampling [46], Sobel-like operator [13, 16] and differentiable least square [28, 26].

Despite the improvements brought about by the existing efforts, a critical issue remains unsolved, i.e., how to determine the reliable local geometry to correlate the normal constraint with the depth estimation. For example, at shape boundaries or corners, the neighboring pixels for a point could belong to different geometries, where the local plane assumption is not satisfied. Due to this challenge, these methods either struggle to capture the local features [46], or sensitive to local geometric variations (noises or boundaries) [13, 16], or computationally expensive [28, 26].

Given the significance of local context constraints, there is a multitude of works on how to incorporate shape-ware regularization in monocular reconstruction tasks, rang-
ing from sophisticated variational approaches for optical flow [33, 41, 32, 1] to edge-aware filtering in stereo [36] and monocular reconstruction [14, 29]. However, these methods have complex formulations and only focus on 2D feature edges derived from image intensity variation, without considering geometric structures of shapes in 3D space.

In this paper, we introduce a simple yet effective method to correlate depth estimation with surface normal constraint. Our formulation is much simpler than any of the aforementioned approaches, but significantly improves the depth prediction quality, as shown in Fig. 1. Our key idea is to adaptively determine the faithful local geometry from a set of randomly sampled candidates to support the normal estimation. For a target point on the image, first, we randomly sample a set of point triplets in its neighborhood to define the candidates of normals. Then, we determine the confidence score of each normal candidate by measuring the consistency of the learned latent geometric features between the candidate and the target point. Finally, the normal is adaptively estimated as a weighted sum of all the candidates.

Our simple strategy has some unique advantages: 1) the random sampling captures sufficient information from the neighborhood of the target point, which is not only highly efficient for computation, but also accommodates various geometric context; 2) the confidence scores adaptively determine the reliable candidates, making the normal estimation robust to local variations, e.g., noises, boundaries and sharp changes; 3) we measure the confidence using the learned contextual features, of which representational capacity is applicable to complex structures and informative to correlate the normal constraint with the estimated depth. More importantly, our method achieves superior results on the public datasets and considerably outperforms the state-of-the-art methods.

Our main contributions are summarized as follows:

- We introduce a novel formulation to derive geometric constraint for depth estimation, i.e., adaptive surface normal.
- Our method is simple, fast and effective. It is robust to noises and local variations and able to consistently capture faithful geometry.
- Our method outperforms the state-of-the-art method on the public datasets by a large margin.

2. Related Work

Monocular depth estimation As an ill-posed problem, monocular depth estimation is challenging, given that minimal geometric information can be extracted from a single image. Recently, benefiting from the prior structural information learned by the neural network, many learning-based works [6, 22, 43, 9, 34, 31, 11, 10, 24, 23] have achieved promising results. Eigen et al. [6] directly estimate depth maps by feeding images into a multi-scale neural network. Laina et al. [18] propose a deeper residual network and further improve the accuracy of depth estimation. Liu et al. [22] utilize a continuous conditional random field (CRF) to smooth super-pixel depth estimation. Xu et al. [43] propose a sequential network based on multi-scale CRFs to estimate depth. Fu et al. [9] design a novel ordinal loss function to recover the ordinal information from a single image. Unfortunately, the estimated depth maps of these methods always fail to recover important 3D geometric features when converted to point clouds, since these methods do not consider any geometric constraints.

Joint depth and normal estimation Since the depth and surface normal are closely related in terms of 3D geometry, there has been growing interests in joint depth and normal estimation using neural networks to improve the performance. Several works [5, 47, 42, 20] jointly estimate depth and surface normal using multiple branches and propagate the latent features of each other. Nevertheless, since there are no explicit geometric constraints enforced on the depth estimation, the predicted geometry of these methods is still barely satisfactory.

Consequently, methods [44, 45, 30, 13, 28, 26, 16] are proposed to explicitly enforce geometric constraints on estimated depth maps. Hu et al. [13] and Kusupati et al. [16] utilize a Sobel-like operator to calculate surface normals from estimated depths, and then enforce them to be consistent with the ground truth. Nonetheless, the Sobel-like operator can be considered as a fixed filter kernel that indiscriminately acts on the whole image (see Fig. 3), leading to unacceptable inaccuracy and sensitivity to noises. To constrain surface normal more reliably, Qi et al. [28] and Long et al. [26] propose to utilize a differentiable least square module for surface normal estimation. These methods optimize the geometric consistency, of which solution is more accurate and robust to noises but limited to expensive computation. Yin et al. [46] introduce virtual normal, which is a global geometric constraint derived from the randomly sampled point triplets from estimated depth. However, this constraint struggles to capture local geometric features, given that the point triplets are randomly sampled from the whole image.

Edge preserving methods Out of the statistical relations between shape boundaries and image intensity edges, many works leverage this statistic prior to benefit many vision tasks. Anisotropic diffusion [27, 2, 40] is a well-known technique for image denoising without removing important details of image content, typically edges. Works [33, 41, 32, 1] propose variational approaches with anisotropic diffusion to model local edge structures for optical flow estimation. Some stereo/monocular depth estimation works rely on pretrained edge detection network [36] or Canny
edge detector [14, 29], to extract image edges to improve depth estimation. However, only a small fraction of the intensity edges keep consistent with true geometric shape boundaries. Our method could detect the true shape boundaries where 3D geometry changes instead of image intensity edges.

3. Method

Given a single color image $I$ as input, we use an encoder-decoder neural network to output its depth map $D_{\text{pred}}$. Our approach aims to not only estimate accurate depth but also recover high-quality 3D geometry. For this purpose, we correlate surface normal constraint with depth estimating. Overall, we enforce two types of supervision for training the network. First, like most of depth estimation works, we employ a pixel-wise depth supervision like $L_1$ loss over the predicted depth $D_{\text{pred}}$ and ground truth depth $D_{\text{gt}}$. Moreover, we compute the surface normal $N_{\text{pred}}$ from $D_{\text{pred}}$ using an adaptive strategy, and enforce consistency between $N_{\text{pred}}$ with the ground truth surface normal $N_{\text{gt}}$, named as Adaptive Surface Normal (ASN) constraint. The method is overviewed in Fig. 2.

**Local plane assumption.** To correlate surface normal constraint with depth estimation, we adopt the local plane assumption following [28, 26]. That is, a small set of neighborhoods of a point forms a local plane, of which normal vector approximates the surface normal. Hence, for a pixel on the depth map, its surface normal can be estimated by the local patch formed by its neighboring points. In theory, the local patch could have arbitrary shapes and sizes. In practice, however, square local patches are widely adopted with sizes $(2m + 1) \times (2m + 1), m = 1, 2, ..., n$, due to its simplicity and efficiency.

**Normal candidates sampling.** To compute the surface normal, unlike prior works utilize least square fitting [28, 26] or Sobel-like kernel approximation [13, 16], we propose a randomly sampling based strategy.

For a target point $P_i \in \mathbb{R}^3$, we first extract all the points $\mathbb{T}_i = \{ P_j \mid P_j \in \mathbb{R}^3, j = 0, \ldots, r^2 - 1 \}$ within a local patch of size $r \times r$. Then, we randomly sample $K$ point triplets in $\mathbb{T}_i$. All sampled point triplets for the target point $P_i$ are denoted as $\mathbb{T}_i = \{ (P_k^A, P_k^B, P_k^C) \mid P_k \in \mathbb{R}^3, k = 0, \ldots, K - 1 \}$. If the three points are not colinear, the normal vector of the sampled local plane can be directly computed by the cross-product:

$$\vec{n}_k = \frac{P_k^A \vec{P}_k^B \times P_k^A \vec{P}_k^C}{|P_k^A \vec{P}_k^B \times P_k^A \vec{P}_k^C|}.$$  \hspace{1cm} (1)

A normal vector will be flipped according to the viewing direction if it does not match the camera orientation. In this way, for each target point, we obtain $K$ normal candidates corresponding to $K$ sampled local planes. Next, we adaptively determine the confidence of each candidate to derive the final normal estimation result.

**Geometric context adaption.** We observe that the neighbors of a target point may not lie in the same tangent plane, especially at a region where the geometry changes, e.g., shape boundaries or sharp corners. Thus, we propose to
learn a guidance feature map that is context aware to reflect the geometric variation. Therefore, the network can determine the confidence of the neighboring geometry by measuring the learned context features.

Given the learned guidance feature map, we measure the $L_2$ distance of the features of a sampled point $P_i$ and the target point $P_j$, and then use a normalized Gaussian kernel function to encode their latent distance into $[0, 1]$:

$$
\mathcal{L}(P_i, P_j) = e^{-0.5\|f(P_i) - f(P_j)\|_2}
$$

$$
\overline{\mathcal{L}}(P_i, P_j) = \frac{\mathcal{L}(P_i, P_j)}{\sum_{n \in P_i} \mathcal{L}(P_i, P_n)},
$$

where $f(\cdot)$ is the guidance feature map, $\|\cdot\|_2$ is $L_2$ distance, and $P_i$ is the neighboring point set in the local patch of $P_i$ as aforementioned. E.q. 2 gives a confidence score, where the higher the confidence is, the more likely the two is to locate in the same tangent plane with target pixel. Accordingly, the confidence score of a local plane $(P^A_k, P^B_k, P^C_k)$ to the center point $P_i$ given by the geometric adaption is defined by:

$$
g_k = \prod_{t=A,B,C} \overline{\mathcal{L}}(P_i, P^t_k).
$$

This is the multiplication of three independent probabilistic scores of the three sampled points, which measures the reliability of a sampled local plane.

**Area adaption.** The area of a sampled local plane (triangle) is an important reference to determine the reliability of the candidate. A larger triangle captures more information and thus would be more robust to local noise, as shown in [46]. For a triangle $T_k$, we simply consider its projected area $s_k$ on the image as a measurement of confidence score. Note that the area is calculated on the 2D image, since the sampled triangles in the 3D space could be very large due to depth variation, leading to unreasonable overestimation.

Finally, the normal for a point $P_i$ is determined by a weighted combination of all the $K$ sampled candidates, where the weights represent the confidence given by our adaptive strategy:

$$
\tilde{n}_i = \frac{\sum_{k=0}^{K-1} s_k \cdot g_k \cdot \hat{n}_k}{\sum_{k=0}^{K-1} s_k \cdot g_k},
$$

where $K$ is the number of sampled triplets, $s_k$ is the projected area of three sampled point $(P^A_k, P^B_k, P^C_k)$ on the 2D image, and $n_k$ is its normal vector.

### 4. Implementation

**Network architecture** Our network adopts a multi-scale structure, which consists of one encoder and two decoders. We use HRNet-48 [37] as our backbone. Taking one image as input, one encoder produces coarse-to-fine estimated depths in four scales, and the other decoder is used to generate the guidance feature map that captures geometric context. The depth estimation decoder consists of four blocks in different scales, each of which is constituted by two ResNet [12] basic blocks with a feature reducing convolution layer. The appending convolution layers are used to produce one-channel output for depth estimation. The guidance feature encoder adopts a nearly identical structure with the depth encoder but without any feature reducing layers.

**Loss functions** Our training loss has two types of terms: depth loss term and surface normal loss term. For the depth term, we use the $L_1$ loss for our multi-scale estimation:

$$
l_d = \sum_{s=0}^{3} \lambda^{s-3} \|D^s_{\text{pred}} - D_{\text{gt}}\|_1,
$$

where $D^s_{\text{pred}}$ means the estimated depth map at the $s^{th}$ scale, $D_{\text{gt}}$ is the ground truth depth map, and $\lambda$ is a factor for balancing different scales. Here we set $\lambda = 0.8$.

To enforce geometric constraint on the estimated depth map, using our proposed adaptive strategy, we compute the surface normals only based on the finest estimated depth map. To regularize the consistency of the computed surface normals with ground truth, we adopt a cosine embedding loss:

$$
l_n = 1 - \cos(N_{\text{pred}}, N_{\text{gt}}),
$$

where $N_{\text{pred}}$ is the surface normal map calculated from the finest estimated depth map, and $N_{\text{gt}}$ is the ground truth surface normal. Therefore, the overall loss is defined as:

$$
l = l_d + \alpha l_n,
$$

where $\alpha$ is set to 5 in all experiments, which is a trade-off parameter to make the two types of terms roughly of the same scale.
Training details Our model is implemented by Pytorch with Adam optimizer (\(init_{lr} = 0.0001\), \(\beta_1 = 0.9\), \(\beta_2 = 0.999\), \(weight\_decay = 0.00001\)). The learning rate is polynomially decayed with polynomial power 0.9. The model is trained with only depth loss term in the first 20 epochs, and then with depth and surface normal loss terms in the last 20 epochs. The whole training is completed with 8 batches on four GeForce RTX 2080 Ti GPUs. We adopt 5 × 5 local patch and 40 sampling triplets in all experiments.

5. Experiments

5.1. Dataset

NYUD-V2 Our model is trained on NYUD-V2 dataset. NYUD-V2 is a widely used indoor dataset and contains 464 scenes, of which 249 scenes are for training and 215 for testing. We directly adopt the collected training data provided by Qi et al. [17], which has 30,816 frames sampled from the raw training scenes with precomputed ground truth surface normals. The precomputed surface normals are generated following the procedure of [7]. Note that DORN [9], Eigen et al. [5], Xu et al. [43], Laina et al. [18], and Hu et al. [13] use 407k, 120k, 90k, 95k and 51k images for training, which are all significantly larger than ours. For testing, we utilize the official test set that is the same as the competitive methods, which contains 654 images.

ScanNet We also evaluate our model on a recently proposed indoor dataset, ScanNet [4], which has more than 1600 scenes. Its official test split contains 100 scenes, and we uniformly select 2167 images from them for cross-dataset evaluation.

5.2. Evaluation metrics

To evaluate our method, we compare our method with state-of-the-arts in three aspects: accuracy of depth estimation, accuracy of recovered surface normal, and quality of recovered point cloud.

<table>
<thead>
<tr>
<th>Method</th>
<th>rel (%)</th>
<th>(\log_{10}) (%)</th>
<th>(rms) (%)</th>
<th>(\delta_1) (%)</th>
<th>(\delta_2) (%)</th>
<th>(\delta_3) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saxena et al. [15]</td>
<td>0.349</td>
<td>-</td>
<td>1.214</td>
<td>0.447</td>
<td>0.745</td>
<td>0.897</td>
</tr>
<tr>
<td>Karsch et al. [15]</td>
<td>0.349</td>
<td>0.131</td>
<td>1.21</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Liu et al. [25]</td>
<td>0.335</td>
<td>0.127</td>
<td>1.06</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ladicky et al. [17]</td>
<td>0.232</td>
<td>0.094</td>
<td>0.821</td>
<td>0.621</td>
<td>0.886</td>
<td>0.968</td>
</tr>
<tr>
<td>Li et al. [20]</td>
<td>0.187</td>
<td>0.078</td>
<td>0.74</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Roy et al. [34]</td>
<td>0.213</td>
<td>0.087</td>
<td>0.759</td>
<td>0.650</td>
<td>0.906</td>
<td>0.974</td>
</tr>
<tr>
<td>Liu et al. [22]</td>
<td>0.220</td>
<td>0.094</td>
<td>0.745</td>
<td>0.605</td>
<td>0.890</td>
<td>0.970</td>
</tr>
<tr>
<td>Wang et al. [38]</td>
<td>0.158</td>
<td>0.064</td>
<td>0.641</td>
<td>0.769</td>
<td>0.950</td>
<td>0.988</td>
</tr>
<tr>
<td>Eigen et al. [5]</td>
<td>0.149</td>
<td>-</td>
<td>0.620</td>
<td>0.896</td>
<td>0.958</td>
<td>0.987</td>
</tr>
<tr>
<td>Chakrabarti et al. [13]</td>
<td>0.143</td>
<td>0.063</td>
<td>0.635</td>
<td>0.788</td>
<td>0.958</td>
<td>0.991</td>
</tr>
<tr>
<td>Li et al. [21]</td>
<td>0.127</td>
<td>0.055</td>
<td>0.573</td>
<td>0.811</td>
<td>0.953</td>
<td>0.988</td>
</tr>
<tr>
<td>Laina et al. [18]</td>
<td>0.115</td>
<td>0.050</td>
<td>0.530</td>
<td>0.866</td>
<td>0.975</td>
<td>0.993</td>
</tr>
<tr>
<td>Hu et al. [13]</td>
<td>0.115</td>
<td>0.051</td>
<td>0.509</td>
<td>0.828</td>
<td>0.965</td>
<td>0.992</td>
</tr>
<tr>
<td>GeoNet [28]</td>
<td>0.128</td>
<td>0.057</td>
<td>0.569</td>
<td>0.834</td>
<td>0.960</td>
<td>0.990</td>
</tr>
<tr>
<td>VNL [46]</td>
<td>0.108</td>
<td>0.048</td>
<td>0.416</td>
<td>0.875</td>
<td>0.976</td>
<td>0.994</td>
</tr>
<tr>
<td>BTS [19]</td>
<td>0.113</td>
<td>0.049</td>
<td>0.407</td>
<td>0.871</td>
<td>0.977</td>
<td>0.995</td>
</tr>
<tr>
<td>Ours</td>
<td>0.101</td>
<td>0.044</td>
<td>0.377</td>
<td>0.890</td>
<td>0.982</td>
<td>0.996</td>
</tr>
</tbody>
</table>

5.3. Evaluations

Depth estimation accuracy We compare our method with other state-of-the-art methods on NYUD-V2 dataset. As shown in Table 1, our method significantly outperforms the other SOTA methods across all evaluation metrics. Moreover, to further evaluate the generalization of our method, we compare our method with some strong SOTAs on unseen ScanNet dataset. As shown in Table 2, our method still shows better performance than others.

Besides the quantitative comparison, we show some qualitative results for several SOTA methods that also use geometric constraints, including i) GeoNet [28] (least square normal); ii) VNL [46] (virtual normal constraint); iii) BTS [19] (predict local plane equations not directly predict depth). As shown in Fig. 4, the proposed method faithfully recovers the original geometry. For the regions with high curvatures, such as the sofas, our results obtain cleaner and smoother surfaces; our predicted depth map also yields high-quality shape boundaries, which leads to better accuracy compared to the ground truth depth map. Also, note even for the texture-less walls and floors, our estimated depth is still satisfactory.

Point cloud From the Table 3, in terms of the quality of point cloud, our method outperforms other methods by a
Figure 4. Qualitative comparisons with SOTAs on NYUD-V2. Compared with other methods, our estimated depth is more accurate and contain less noises. The recovered surface normal maps and point clouds demonstrate that our estimated depth faithfully preserve important geometric features. The black regions are the invalid regions lacking ground truth.

large margin. Surprisingly, although VNL [46] has better performance than GeoNet [28] in terms of depth evaluation errors, its mean Euclidean distance is worse than GeoNet, which demonstrates the necessity of evaluation specially designed for point clouds. As shown in Fig. 4 (the third row), our point cloud has fewer distortions and much more accurate than others. The point clouds generated from the depth maps of other methods suffer from severe distortions and struggle to preserve prominent geometric features, such as planes (e.g., walls) and surfaces with high curvatures (e.g., sofas). Besides, we also show a qualitative comparison between our point cloud and the ground truth from a different view in Fig. 4. The highly consistent result further demonstrates our method’s superior performance in terms of the quality of 3D geometry.

Surface Normal As shown in Table 4, our recovered surface normals have considerably better quality than that of the other methods. For reference, we also report the results generated by the methods that directly output normal maps in the network. Surprisingly, the accuracy of our recovered surface normals is even higher than this kind of methods that can explicitly predict normals. Also, we present qualitative comparisons in Fig. 4. It can be seen that our surface normal is smoother and more accurate than the others, which indicates that our strategy is effective for correlating normal constraints with depth estimation, resulting in not only accurate depth estimation, but also reliable surface normals and 3D geometry.

5.4. Discussions

In this section, we further conduct a series of evaluations with an HRNet-18 [37] backbone to give more insights into the proposed method.

Effectiveness of ASN To validate the effectiveness of our proposed adaptive surface normal constraint, we train models with different constraints: a) only $L_1$ depth constraint; b) depth and Sobel-like operator surface normal constraints (SOSN); c) depth and least square surface normal constraints (LSSN); d) depth and virtual normal constraints (VN); e) depth and our adaptive surface normal constraints (ASN).

As shown in Table 5, the model with our adaptive surface normal constraint outperforms (ASN) all the others. Although the models with Sobel-like operator (SOSN) and least square normal constraint (LSSN) shows better recovered surface normal, their depth estimation accuracy drops off compared with the model without geometric constraint. The model with virtual normal (VN) [46] constraint shows the worst quality of recovered surface normal among the four types of geometric constraints, given that virtual nor-

Table 4. Surface normal evaluation on NYUD-V2 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean ($)</th>
<th>Median ($)</th>
<th>11.25° (%)</th>
<th>22.5° (%)</th>
<th>30° (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Surface Normal from the Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3DP [17]</td>
<td>33.0</td>
<td>28.3</td>
<td>18.8</td>
<td>40.7</td>
<td>52.4</td>
</tr>
<tr>
<td>Ladicky et al. [17]</td>
<td>35.5</td>
<td>25.5</td>
<td>24.0</td>
<td>45.6</td>
<td>55.9</td>
</tr>
<tr>
<td>Fouhey et al. [3]</td>
<td>35.2</td>
<td>17.9</td>
<td>40.5</td>
<td>54.1</td>
<td>58.9</td>
</tr>
<tr>
<td>Wang et al. [39]</td>
<td>28.8</td>
<td>17.9</td>
<td>35.2</td>
<td>57.1</td>
<td>65.5</td>
</tr>
<tr>
<td>Eigen et al. [5]</td>
<td>23.7</td>
<td>15.5</td>
<td>39.2</td>
<td>62.0</td>
<td>71.1</td>
</tr>
<tr>
<td>Calculated Surface Normal from the Point cloud</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTS [19]</td>
<td>44.0</td>
<td>35.4</td>
<td>14.4</td>
<td>32.8</td>
<td>43.2</td>
</tr>
<tr>
<td>GeoNet [28]</td>
<td>36.8</td>
<td>32.1</td>
<td>15.0</td>
<td>34.5</td>
<td>46.7</td>
</tr>
<tr>
<td>DORN [9]</td>
<td>36.6</td>
<td>31.1</td>
<td>15.7</td>
<td>36.5</td>
<td>49.4</td>
</tr>
<tr>
<td>Hu et al. [13]</td>
<td>32.1</td>
<td>23.5</td>
<td>24.7</td>
<td>48.4</td>
<td>59.9</td>
</tr>
<tr>
<td>VNL [46]</td>
<td>24.6</td>
<td>17.9</td>
<td>34.1</td>
<td>60.7</td>
<td>71.7</td>
</tr>
<tr>
<td>Ours</td>
<td>20.0</td>
<td>13.4</td>
<td>43.5</td>
<td>69.1</td>
<td>78.6</td>
</tr>
</tbody>
</table>
Figure 5. Comparisons of models with different geometric constraints. Model with our ASN constraint achieves better surface normal estimation, even accurately capture detailed geometries, like the legs of chairs (see white boxes).

Table 5. Comparisons of models with different geometric constraints on NYUD-V2 dataset.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>rel($\downarrow$)</th>
<th>log10($\downarrow$)</th>
<th>$\delta_1$ ($\uparrow$)</th>
<th>Mean ($\downarrow$)</th>
<th>Median ($\downarrow$)</th>
<th>11.25° ($\uparrow$)</th>
<th>22.5° ($\uparrow$)</th>
<th>30° ($\uparrow$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>0.113</td>
<td>0.047</td>
<td>0.875</td>
<td>31.8</td>
<td>23.2</td>
<td>24.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1 + SOSN</td>
<td>0.118</td>
<td>0.049</td>
<td>0.867</td>
<td>22.8</td>
<td>16.1</td>
<td>36.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1 + LSSN</td>
<td>0.119</td>
<td>0.050</td>
<td>0.862</td>
<td>23.5</td>
<td>16.3</td>
<td>35.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1 + VN</td>
<td>0.111</td>
<td>0.047</td>
<td>0.876</td>
<td>31.7</td>
<td>21.4</td>
<td>28.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1 + ASN</td>
<td>0.111</td>
<td>0.047</td>
<td>0.876</td>
<td>22.2</td>
<td>15.8</td>
<td>36.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Ablation study of the proposed adaptive modules on NYUD-V2 dataset. We evaluate the accuracy of the recovered surface normals.

<table>
<thead>
<tr>
<th>Module</th>
<th>Mean ($\downarrow$)</th>
<th>Median ($\downarrow$)</th>
<th>11.25° ($\uparrow$)</th>
<th>22.5° ($\uparrow$)</th>
<th>30° ($\uparrow$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>only Area</td>
<td>22.6</td>
<td>16.0</td>
<td>36.4</td>
<td>63.6</td>
<td>74.4</td>
</tr>
<tr>
<td>only GC</td>
<td>22.3</td>
<td>15.8</td>
<td>36.9</td>
<td>64.1</td>
<td>74.8</td>
</tr>
<tr>
<td>Area+GC</td>
<td>22.2</td>
<td>15.8</td>
<td>36.9</td>
<td>64.2</td>
<td>74.9</td>
</tr>
</tbody>
</table>

Ablation study of adaptive modules To evaluate the effect of the proposed two adaptive modules, i.e., geometric context adaption and area adaption, we conduct an ablation study. We train models with different adaptive modules: only Geometric Context (GC) adaptation, only Area adaptation, and both. From Table 6, we can see the model with both adaptive modules achieves the best performance, which verifies the necessity of each adaptive module.

Visualization of guidance features The geometric adaptive module is the key to our adaptive surface normal constraint method. To better understand what the network learns, we visualize the learned features of the guidance map. We plot one from the multiple channels of the guidance feature map, which is shown in Fig. 6. It can be seen that the learned guidance map captures the shape context and geometric variations, giving informative boundaries.

For comparison, we use the Canny operator to detect the edges of the input image by image intensity variance. As we can see, our guidance feature map is not simply coincident with the Canny edges. For example, in Fig. 6, the Canny operator detects fragmented edges based on the texture of wall painting and sofas, while our guidance feature map indicates the true shape boundaries where the 3D geometry changes.
Figure 8. The accuracy of recovered surface normal versus the number of sampled triplets. The more triplets are sampled, the more accurate the recovered surface normal is.

Visualization of similarity kernels To validate whether our model can capture the true geometric boundaries of shapes, we select five points on the image and visualize their color coded similarity kernels in Fig. 7. The similarity kernels of Point A, B, and E indicate that our method could successfully distinguish different geometries, such as shape boundaries and corners. Furthermore, the similarity kernels of Point C and D show that our approach captures the 3D geometry variances of the shapes in the real world, instead of the color distinctions of the image. For example, Point D has large color variances in the image, but its similarity kernel has constant values indicating the unchanged geometry.

Number of sampled triplets. To quantitatively analyze the influence of the number of sampled triplets, we recover surface normals from our estimated depth maps using our adaptive surface normal computation method with $5 \times 5$ local patch. Based on Fig. 8, it is not surprising that more sampled triplets will contribute to more accurate surface normals. The accuracy increases dramatically from 10 $\sim$ 20 sampled triplets and gradually saturates with more triplets sampled. To balance efficiency and accuracy, the number of sampled triplets is recommended to be $40 \sim 60$.

Size of local patch. We evaluate the effect of the size of local patch to our method by training the network with different local patch sizes. As illustrated in Table 7, a larger local patch could improve the performance, especially for the surface normal, but the improvements are not significant. The reason behind is, our ASN constraint is an adaptive strategy that can automatically determine the reliability of the sampled points given different local patches; therefore, our method is robust to the choice of local patch size.

Area-based adaption We use the area of a sampled triangle as the combinational weight for adaption. To evaluate the effectiveness of the area-based adaption, we conduct an experiment with a comparison to the simple average strategy. We create a unit semi-sphere surface as noise-free data and then add Gaussian noises to simulate real noisy data (see Fig. 9 (a)). We compare the mean of angle errors of the normals estimated by these two methods with the increase of noises, and the results are given in Fig. 9 (b). We can see that our area-based adaption gives lower estimation error with the increase of noise level, demonstrating the robustness of the use of area for adaption.

Time complexity Here, we discuss the time complexity of different normal computation methods, including our sampling based method, Sobel-like operator [13, 16] and least square based method [28, 26]. Ours and the Sobel-like operator only involve matrix addition and vector dot/cross production operations; thus it is easy to show the time complexity is $O(n)$, while our time complexity will increase linearly with more samples. However, the least square module [28, 26] directly calculates the closed form solution of least square equations, which involves matrix multiplication, inversion and determinant, leading to the time complexity of $O(n^3)$. Therefore, our method effectively balances the accuracy and computational efficiency.

6. Conclusion

In this paper, we present a simple but effective Adaptive Surface Normal (ASN) constraint for monocular depth estimation. Compared with other surface normal constraints,
our constraint could adaptively determine the reliable local geometry for normal calculation, by jointly leveraging the latent image features and explicit geometry properties. With the geometric constraint, our model not only estimates accurate depth maps but also faithfully preserves important 3D geometric features so that high-quality 3D point clouds and surface normal maps can be recovered from the estimated depth maps.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. Wenping Wang acknowledges support from AIR@InnoHK – Center for Transformative Garment Production (TransGP). Christian Theobalt acknowledges support from ERC Consolidator Grant 4DReply (770784). Lingjie Liu acknowledges support from Lise Meitner Postdoctoral Fellowship.

References

[22] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep


