HeIght Gradient Histogram (HIGH) for 3D Scene Labeling

Gangqiang Zhao, Junsong Yuan, Kang Dang
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
gqzhao@ntu.edu.sg, jsyuan@ntu.edu.sg, kangdang@gmail.com

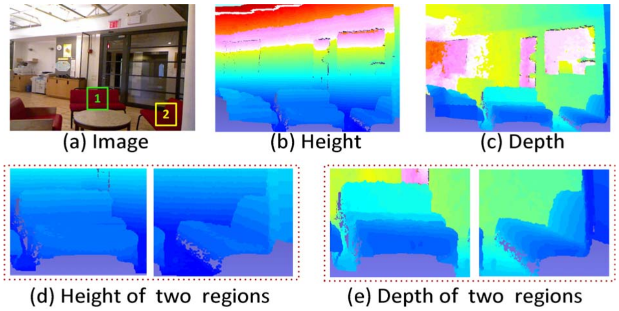
Abstract

RGB-D (color + 3D pointcloud) based scene labeling has received much attention due to the affordable RGB-D sensors such as Microsoft Kinect. To fully utilize the RGB-D data, it is critical to develop robust features that can reliably describe the 3D shape information of the pointcloud data. Previous work has proposed to extract SIFT-like features from the depth dimension data directly while ignored the important height dimension data of the 3D pointcloud. In this paper, we propose to describe 3D scene using height gradient information and propose a new compact pointcloud feature called HeIght Gradient Histogram (HIGH). Using TextonBoost as the pixel classifier, the experiments on two benchmarked 3D scene labeling datasets show that HIGH feature can well handle the intra-category variations of object class, and significantly improve class-average accuracy compared with the state-of-the-art results. We will publish the code of HIGH feature for the community.

1. Introduction

Scene labeling, aiming to concurrently recognize and segment multiple object classes in a scene, is one of the fundamental problems in computer vision. Image based scene labeling has been studied extensively [39, 31, 37]. With the recent advances of 3D sensors (e.g., depth cameras, laser scanners), it is now convenient to obtain both RGB image and 3D pointcloud data to capture a scene. The work of Siberman and Fergus [33], Siberman *et al*. [34], Ren *et al*. [28], Valentin *et al*. [40], Zhao *et al*. [44] and Munoz *et al*. [25] have shown that using both image and 3D pointcloud data can achieve better labeling results than using image data alone.

Although it is a promising direction, one major challenge to use RGB-D data for scene labeling is to extract descriptive features from the pointcloud data, such that the 3D shape information can be utilized to improve scene understanding. To describe 3D object, the classical SpinImage [14] and 3D shape context [12] features are still popular choices. Both SpinImage and 3D shape context describe one 3D point using the coordinate information of its neighbouring points. The plain point coordinates may be appropriate for recognizing objects with similar shapes. However, for 3D scene labeling, there are large intra-category variations due to object shape diversity, viewpoint changes, partial occlusion, etc. High discriminative 3D features are required to handle these variations. To mitigate this challenge, Siberman and Fergus [33] and Ren *et al*. [28] treat depth image as grayscale image and extract SIFT-like [22] feature directly. However, since the depth data heavily depends on sensor's position and orientation, e.g., the depth of the same object can change significantly when the sensor changes position or view point, it affects the performance of depth based features. In addition, as a 2D representation, depth image cannot capture 3D shape information, which is an important cue for 3D object description. Thus special treatment needs to be taken.



(a) Image (b) Height (c) Depth
(d) Height of two regions (e) Depth of two regions

Figure 1. Height vs. Depth. (a) is one typical frame in NYU Depth dataset [33]. (b) and (c) are corresponding 3D pointclouds rendered according to the height and depth values of each point, respectively. The blue color represents the lower or nearer points while the red color represents higher or farther points. (d) and (e) highlight the height and depth data of two sofa regions in (a).

In this paper, we propose to employ the height dimension data of the 3D pointclouds for scene description due to its following advantages. First of all, in contrast with depth data, height data is less sensitive to the distance and view

978-1-4799-7000-1/14 $31.00 © 2014 IEEE
DOI 10.1109/3DV.2014.16

point of the 3D sensor, especially when the sensor has near horizontal motion. For example, two identical sofas in Figure 1(a) can be replaced with each other without changing their height data, as shown in Figure 1(d). As height data is usually benchmarked at the ground level, it remains the same even the 3D sensor changes the location on the ground plane. Second, height data can capture the spatial layout information of the scene, which provides important context cue for scene understanding. For example, in the indoor scene, sofas are usually placed on the floor while a television is possibly placed on a TV table. Moreover, height data contains rich information for 3D scene description. For example, different sofas have different sizes and shapes. But they have one common characteristic, i.e., a sudden height change between the sofa seat and the sofa back. It is easy to locate these regions in the height data (Figure 1). These regions distinguish "sofa" from other categorizes such as "table" and "TV".

To effectively describe 3D scene using height data, we propose a new feature by leveraging the height gradient orientations of neighboring points. Instead of directly treating height data as 2D grayscale image [33] [28], we consider 3D support region for each point, which captures important 3D shape information, as illustrated by Spin-Image [14] and 3D shape context [12]. We call our feature as HeIght Gradient Histogram (HIGH). To extract HIGH feature, we first obtain height dimension data from the pointcloud and estimate 3D height gradient for each point. After that, the support region of each point is partitioned into several concentric spherical shells, and a gradient orientation histogram is computed within each shell. Finally, we concatenate the histograms of all shells into a feature vector and normalize it to unit magnitude.

To use HIGH feature for 3D scene labeling, we train one multiclass classifier via TextonBoost [31]. As TextonBoost is originally proposed for 2D image labeling, we extend it to 3D pointcloud labeling by using 3D integral volume representation [15] to speed up weak feature calculation.

The contribution of this paper is three-fold. First, we propose a new perspective to look at 3D data. Compared with depth data, height dimension data describes another aspect of the 3D scene and captures important spatial layout information. Second, we propose a new HIGH feature by leveraging height gradient information. As a general feature extraction framework, our method can also use other kinds of gradient directly, e.g., depth and width gradient. Last but not least, we provide a thorough experimental validation of our feature and experimentally demonstrate that the proposed compact HIGH feature (32-dimension) outperforms several state-of-the-art features for 3D scene labeling on both the indoor NYU Depth dataset [33] and the outdoor Wachtberg laser scanner dataset [4].

2. Related works

To describe 3D data, dozens of 3D features are proposed in past two decades. Following Tombari's way [38], we categorize existed 3D features into two groups: signature feature [35][7][2] and histogram feature [14][12][29][43]. The former one calculates a single value for each neighboring point and the latter one maintains a histogram of neighboring point coordinates or their other properties.

Among signature features, Stein and Medioni propose structural indexing [35]. The signature value of each point encodes either angles between consecutive edges, or local surface orientations. Chua and Jarvis propose point signatures [8] to encode signed height of 3D curve. In Fingerprint [36], Sun and Abidi use both normal angle variations and contour radius variations. To make the signature less sensitive to noise, Manay *et al*. [32] present integral invariant signatures. Bending invariant signature of Elad *et al*. [11], wave kernel signature of Aubry *et al*. [1] and heat kernel signature of Bronstein and Kokkinos [7] are proposed to tackle non-rigid shapes. To handle scale variability, Bariya *et al*. [2] propose a descriptor that encodes components of normal values within the support region.

As far as histogram-based features are concerned, Johnson and Hebert propose SpinImage [14], which combines the advantages of structural indexing [35] and geometric hashing [19]. By extending the shape context [5] to three dimensions, Frome *et al*. propose 3D shape context [12]. Zhong suggests an improvement of 3D shape context [45] using a base octahedron to divide the support region. Instead of using plain 3D coordinates as SpinImage and 3D shape context, geometric properties of neighboring points can be used to enhance the discriminative power of the descriptor. Rusu *et al*. propose Fast Point Feature Histograms (FPFH) [29], which describes normal differences between pairs of points in the support region. Tombari *et al*. propose SHOT descriptor [38], which encodes normal values of neighbouring points. Mian *et al*. [24] accumulate 3D tensors of mesh triangle areas within a cubic support region. Kokkinos *et al*. [7] present intrinsic shape context descriptor by considering the geodesic geometry of neighbouring points.

Inspired by the success of SIFT [22] and SURF [3], several researchers try to employ the gradients to describe 3D data. Zaharescu *et al*. [43] propose Mesh-HOG to emulate SIFT-like image descriptors [43] for 3D mesh data. Darom and Keller [10] propose a local depth SIFT (LD-SIFT) descriptor for 3D mesh and Knopp *et al*. [17] extend SURF [3] for mesh description. Liu *et al*. [21] propose to extract rotation-invariant HOG descriptors using Fourier analysis. To describe RGB-D images, Siberman and Fergus [33] treat depth image as grayscale image and extract SIFT feature directly using depth gradients. Ren *et al*. [28] also propose a depth kernel descriptor (Depth-KDES) by using depth image representation. However, the kernel parameters should be adjusted for each dataset and the 3D spatial information of neighbour points is not used in depth images. Another theme of research, such as VIP feature of Wu *et al.* [42], leverages local shape information to help RGB feature extraction. As for the height dimension data, Golovinskiy *et al.* [13] use the average and standard deviation of height data to describe 3D objects while Kim *et al.* [16] use the distribution of height data to describe 3D objects. Different with these methods, we propose to use the 3D height gradient of points to build the local features in order to account for the shape variances of objects.

3. HIGH Feature for Pointcloud Data

There are several key steps to extract HIGH feature from 3D pointcloud. First, we extract height dimension data from pointcloud. Then we calculate height gradient for each point. Next, the support region of each point is partitioned into several sub-regions in order to incorporate the spatial contextual information of neighboring points. After that, a histogram of gradient is computed within each sub-region. Finally, the histograms of all sub-regions are concatenated into a feature vector which is normalized to unit magnitude. We will elaborate the details below.

3.1. Height Gradient Estimation

Before building HIGH feature, the height dimension data of each point is extracted from 3D pointcloud firstly. Denote the pointcloud as $\mathcal{P}$ and each point $\mathbf{p} \in \mathcal{P}$ has three dimensions, i.e., $\mathbf{p} = (p_x, p_y, p_z)$, where x, y and z correspond to height, width and depth, respectively. For each point $\mathbf{p}$ in the pointcloud, its height data is $f(\mathbf{p}) = p_x$. Then, the height gradient of each point is estimated by considering its neighbouring points.

Definition 1 (Height Gradient) *Denote height data of all points as a function $f : \mathbb{R}^3 \to \mathbb{R}$. Consequently, the height gradient is defined as:*

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k}, \tag{1}$$

where $\mathbf{i} = (1, 0, 0)$, $\mathbf{j} = (0, 1, 0)$, $\mathbf{k} = (0, 0, 1)$ are standard unit vectors.

For a given point $\mathbf{p}$, its height gradient $\nabla f(\mathbf{p})$ is a 3D vector that points to the direction of the greatest rate of increase of height value, and whose magnitude is that rate of increase. $\nabla f(\mathbf{p})$ is computed using linear gradient reconstruction method [9] by considering the neighbours whose distances to $\mathbf{p}$ are less than Γ.

3.2. Support Region Partition

The descriptor for each point $\mathbf{p}$ is computed using a support region, defined as a sphere S centered at point $\mathbf{p}$, as shown in Figure 2(a). To incorporate spatial contextual information of the support region, we do not use the whole region as a global feature. Instead, we partition the support region into K concentric spherical shells, as shown in Figure 2(b). The radius R of support region is equally split to K segments while each segment corresponds to one concentric spherical shell. Thanks to this partition strategy, the corresponding shell of each point $\mathbf{q}$ in the support region can be decided by only considering its distance to point $\mathbf{p}$:

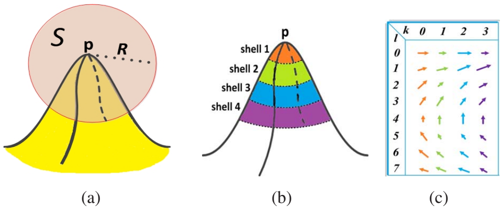


Figure 2. Spatial and gradient orientation bins for HIGH feature building. (a) shows support region (sphere S) for point $\mathbf{p}$. (b) shows the divided sub-regions for point $\mathbf{p}$. Each sub-region is one concentric spherical shell, as shown in one color. (c) is one example of 2D feature histogram with four spatial bins (k) and eight orientation bins (l). The length of arrow represents the accumulated gradient magnitude in each bin.

$$\mathrm{d}(\mathbf{q}) = \|\mathbf{q} - \mathbf{p}\|_2, \tag{2}$$

where $\|\cdot\|_2$ is the $L2$ norm. Each concentric spherical shell is also called as one spatial bin.

3.3. Gradient Orientation

Inspired by the Rotation Invariant Feature Transform (RIFT) feature of Lazebnik *et al.* [20], gradient orientation of each point is measured relative to the direction pointing outward from current point $\mathbf{p}$. For neighboring point $\mathbf{q}$, the gradient orientation $\theta(\mathbf{q})$ is computed as:

$$\theta(\mathbf{q}) = \arccos\left(\frac{\nabla f(\mathbf{q})}{\|\nabla f(\mathbf{q})\|_2} \cdot \frac{\mathbf{q} - \mathbf{p}}{\|\mathbf{q} - \mathbf{p}\|_2}\right), \tag{3}$$

where $\nabla f(\mathbf{q})$ is gradient vector of $\mathbf{q}$. As unsigned gradient is used, the range of gradient orientation is $[0, \pi]$. The gradient information is encoded by using L orientation bins which are evenly spaced over 0 to π.

3.4. HIGH Feature Building

With the discretization of support region and gradient orientation, HIGH feature descriptor is obtained by accumulating gradient magnitude of neighbouring points into bins of a two-dimensional histogram $\mathbf{V}(k, l)$, $0 \le k < K$ and $0 \le l < L$. The first dimension k corresponds to spatial bins while the second dimension l corresponds to gradient orientation bins. In order to reduce aliasing and boundary effects of binning, the gradient magnitude of each point is interpolated bilinearly between adjacent bins. To make the description complete, we present details of feature histogram binning below.

For one neighboring point $\mathbf{q}$, we denote its spatial and gradient orientation index (s, g) as:

$$s = \frac{d(\mathbf{q})}{R}K, \qquad g = \frac{\theta(\mathbf{q})}{\pi}L, \tag{4}$$

where s and g are two real numbers. The gradient of point $\mathbf{q}$ contributes to two neighboring spatial bins $\{s_1, s_2\}$, where $s_1 = \max(\lceil s-1 \rceil, 0)$ and $s_2 = \min(\lfloor s+1 \rfloor, K-1)$. Similarly, it also contributes to two orientation bins $\{g_1, g_2\}$, where $g_1 = \max(\lceil g-1 \rceil, 0)$ and $g_2 = \min(\lfloor g+1 \rfloor, L-1)$. With linear interpolation scheme, each corresponding bin of feature vector $\mathbf{V}(k,l)$ is updated as:

$$\mathbf{V}(k,l) = \mathbf{V}(k,l) + w\|\nabla f(\mathbf{q})\|_2, \tag{5}$$

where $k \in \{s_1, s_2\}$, $l \in \{g_1, g_2\}$, $\|\nabla f(\mathbf{q})\|_2$ is gradient magnitude of point $\mathbf{q}$ and the weight w is set to be $w = (1-|s-k|)(1-|g-l|)$ where $|\cdot|$ represents the absolute value. After updating histogram $\mathbf{V}$ using all neighboring points in support region, we normalize $\mathbf{V}$ to unit magnitude. In our experiments, we use four spatial bins and eight orientation bins and the resulting HIGH feature is very compact, i.e., 32 dimensions. Figure 2 (c) illustrates one example of feature histogram.

The idea of HIGH feature is inspired by the Rotation Invariant Feature Transform (RIFT) feature of Lazebnik *et al.* [20]. The original RIFT has achieved superior performance in image retrieval and image classification tasks, however, we cannot use it directly for 3D pointcloud data because it requires to estimate the gradient of intensity image. Instead, we propose to encode 3D height gradient of points, in order to account for shape variances of objects. Despite the compact size of the 32-dimensional HIGH feature vector, encouraging results are obtained for 3D scene labeling.

4. TextonBoost Classifier for Scene Labeling

To use HIGH feature for 3D scene labeling, we train one multiclass classifier via TextonBoost [31]. As TextonBoost is originally proposed for 2D image labeling, we extend it to 3D pointcloud labeling by using 3D integral volume representation [15] to speed up weak feature calculation.

To train 3D scene labeling classifier, we first extract HIGH feature for each point and quantize all features to $\mathcal{K}$ clusters using k-means clustering. Each point is assigned to the nearest cluster center, producing the texton map. We denote the texton map as T where point $\mathbf{p}$ has value $T_{\mathbf{p}} \in \{1, ..., \mathcal{K}\}$. Then we train the classifier by boosting weak learners based on a set of shape filter responses. Each shape filter is defined by a texton index t (i.e., $t \in \{1, \cdots, \mathcal{K}\}$) and region $\mathcal{R}$. For a given point $\mathbf{p}$, response $n^{\mathbf{p}}_{[t,\mathcal{R}]}$ is the number of points that have texton index t in region $\mathcal{R}$ placed relative to point $\mathbf{p}$. Each weak learner $h_m(n, c)$ for one class c is a decision stump based on a shape filter response $n^{\mathbf{p}}_{[t,\mathcal{R}]}$ [31]. The learned "strong" classifier is an additive model of the form $H(n,c) = \sum_{m=1}^{M} h_m(n,c)$, where M is the number of weak learner. The weight of each individual weak learners is learned as [31].

In the original TextonBoost for image scene labeling [31], each shape filter is defined by a rectangular region, which can be efficiently estimated with integral images. As we use 3D pointcloud, we define each shape filter by a rectangular cuboid. Filter responses of rectangular cuboids can be efficiently computed by building integral pointcloud representation, inspired by integral video representation [15]. To estimate integral pointcloud for texton index t, we first voxelize 3D pointcloud to $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ grid cells. Denote $T^t(x', y', z')$ as the number of points that have texton index t in one grid cell (x', y', z'). The value of integral pointcloud $\hat{T}^t$ at location (x, y, z) is the number of points whose cell indexes are less than or equal to (x, y, z):

$$\hat{T}^t(x,y,z) = \sum_{x' \le x} \sum_{y' \le y} \sum_{z' \le z} T^t(x', y', z') \tag{6}$$

The integral pointcloud can be computed rapidly in one pass over the pointcloud [15]. To compute the shape filter response of any rectangular cuboid aligned with the coordinate axes, only eight array references to the integral pointcloud are necessary.

5. Experiments

To evaluate our HIGH feature, we test it on two public datasets: NYU Depth [33] and Wachtberg dataset [4]. In addition, we compare HIGH feature with several state-of-the-art 3D features: SpinImage [14], 3D shape context [12], FPFH [29], SHOT [38] and KDES-Depth [28]. As our feature extraction method can use other kinds of gradient directly, we also evaluate the discriminative power of depth and width gradients. Denote our feature building with depth gradient as "Depth" while our feature building with width gradient as "Width". The detailed implementation and performance evaluation of our feature are also reported.

5.1. Dataset

NYU Depth dataset contains 2284 RGB-D images of indoor scenes collected using Microsoft Kinect. We use the same ground truth data as [28]. There are 12 object categories plus a generic background class (containing rare objects). Wachtberg dataset [4] is obtained outdoor using one laser range scanner. There are five pointclouds: each of them contains one 360 degree scanning of urban scenes. The pointcloud data is manually labeled with five classes: ground, vegetation, facades, vehicles, and poles. More than half million of 3D points are labeled in total.

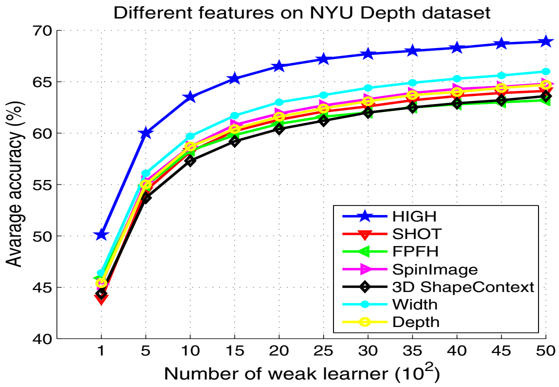


Figure 3. The performance of selected features on NYU Depth dataset [33]. See the text for details.

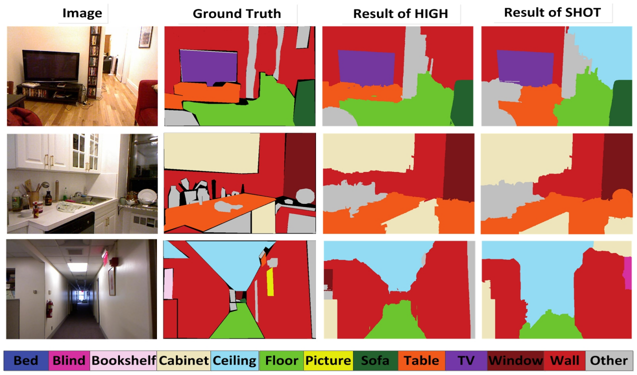


Figure 4. Labeling results on NYU Depth dataset [33] using HIGH feature or SHOT feature. The results are smoothed as [18].

5.2. Experimental Setting

To classify pointcloud data of NYU Depth dataset [33], we extract four scale HIGH descriptors for each point while radii R of support region are $4.0, 4.5, 5.0, 5.5$ centimeters, respectively. Γ is set to be 2.0 centimeter for gradient estimation. We use four spatial bins and eight orientation bins to obtain 32D HIGH feature. For SpinImage [14], FPFH [29], SHOT [38] and 3D shape context [12], we extract the same multi-scale feature descriptors. To compare with color features on this dataset, we extract Texton [23], local binary patterns (LBP) [27], multi-scale dense SIFT [22] and Color SIFT [41] as suggested by [18]. All features are quantised to 150 clusters using standard K-means clustering. As far as Wachtberg dataset [4], we extract HIGH feature with support radius $R = 100$ centimeter and $\Gamma = 50$ centimeter. The same radius is used for other 3D features.

We follow standard experiment setting for each dataset: for NYU Depth, we use 60% data for training and 40% for testing; for Wachtberg dataset, we use 80% data for training and 20% for testing. In NYU Depth, the width, height and length of each grid cell are 2 centimeter while they are 20 centimeter in Wachtberg dataset. The number of weak learners M in TextonBoost is set to 5000 to obtain the final results of two datasets. The results of both dataset are averaged over 5 random runs. Unless mentioned, Global accuracy (%) denotes the overall percentage of correctly classified image pixels and Average accuracy is the unweighted average of per-category classification accuracy [18].

5.3. Evaluation of Features for 3D Scene Labeling

NYU Depth Dataset. To analyze the discriminative power of each feature, we train the Textonboost classifier using only one type of feature each time. Each 3D point is classified independently without spatial smoothness. Figure 3 compares the performance of HIGH feature with other 3D features. The performance of each feature is evaluated with different numbers of weak learners. Each feature has a better performance when using more weak learners. High feature obtains the best average accuracy 68.9% when using 5000 weak learners. This outperforms SpinImage (63.8%), 3D shape context (63.6%), FPFH (63.2%) and SHOT (64.1%). FPFH feature outperforms SHOT feature when the number of weak learners is less than 1000. However, SHOT feature performs much better than FPFH when using more weak learners. The average accuracies of "Width" and "Depth" are 66.0% and 64.7%, respectively. HIGH feature outperforms "Depth" and "Width" features while "Width" feature also outperforms "Depth" feature in this challenging dataset.

Considering this dataset also contains image data, we further compares HIGH feature with four state-of-the-arts color features using the same TextonBoost framework. Table 1 summarizes the results of different features. For color features, dense SIFT [6][22] obtains the best accuracy of 71.1% and Texton [23] obtains comparable results. This comparison shows that our HIGH feature has significantly reduced the performance gap between image features and 3D features. In addition, we compare our feature with depth kernel descriptors (KDES-Depth) [28] and Depth-SIFT [33] in Table 1. We cite the performance of KDES-Depth in [28] and that of Depth-SIFT in [33]. HIGH feature again outperforms KDES-Depth and Depth-SIFT.

Figure 4 illustrates scene labeling results of example test frames using HIGH feature or SHOT feature. It shows that using HIGH feature we can obtain a better parsing result than using SHOT feature. We do well in many challenging situations with complex scene layouts. The use of height gradient can help prevent some failures in object class segmentation, such as the labeling of a wall as a ceiling.

Wachtberg Dataset. To further evaluate HIGH feature, we test it on Wachtberg dataset [4]. Table 2 compares the performance of HIGH feature with that of other 3D features. The HIGH feature outperforms SpinImage,

Table 1. Class-average accuracy (%) and global accuracy (%) of different features on NYU depth dataset. For all features except KDES-Depth [28] and Depth-SIFT [33], the performance is obtained using the same TextonBoost Classifier.

3D Feature	*Dimension*	*Average*	*Global*
HIGH (Our)	32	**68.9** ±0.6	**54.1** ±0.4
Width (Our)	32	66.0 ±0.7	49.7 ±0.6
Depth (Our)	32	64.7 ±0.9	48.7 ±0.3
SHOT	352	64.1 ±0.4	47.3 ±0.2
FPFH	33	63.2 ±0.3	48.0 ±0.1
SpinImage	153	63.8 ±0.9	47.4 ±0.5
3D shape context	64	63.6 ±0.4	46.7 ±0.7
KDES-Depth [28]	400	63.4 ±0.6	/
Depth-SIFT [33]	128	56.6 ±2.9	/

RGB Feature	*Dimension*	*Average*	*Global*
SIFT	128	**71.1** ±0.3	**56.5** ±0.1
Texton	17	71.0 ±0.4	55.5 ±0.2
ColorSIFT	372	70.2 ±0.9	55.7 ±0.5
LBP	121	68.2 ±0.6	52.6 ±0.4

3D shape context, FPFH and SHOT. The average accuracy is improved from 79.5% (SpinImage) to 83.7%. SpinImage also obtains a good result possibly because of the relatively small intra-category variations of this dataset. Different with NYU Depth dataset, "Depth" (78.7%) feature outperforms "Width" (68.6%) in this dataset.

Table 2 further compares our feature with Spectral Histogram (SH) and Distribution Histogram (DH) features [4]. As there is no public implementation of these two features, we cite their performance in [4] which are obtained by using Spectrally Hashed Logistic Regression (SHLR) classifier [4] or Max-Margin Markov Networks (FM^3N) classifier [26]. The proposed HIGH feature again greatly outperforms Spectral Histogram and Distribution Histogram features in this dataset. Figure 5 shows one snapshot of point-cloud parsing result using HIGH feature. It can be seen that most points of ground, vegetation and pole are correctly classified while vehicle is confused with facade and vegetation sometimes, partly due to the limited training samples.

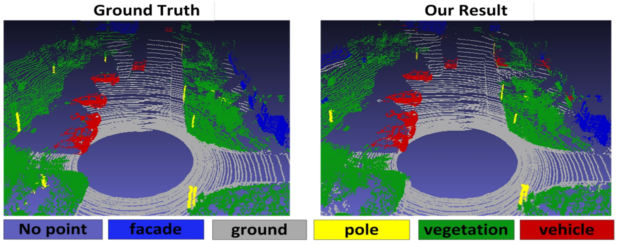


Figure 5. The labeling result of one point cloud in Wachtberg dataset [4] using HIGH feature. The results are smoothed as [18].

Table 2. Class-average accuracy (%) and global accuracy (%) of different features on Wachtberg dataset. For all features except SH and DH [4], the performance is obtained using the same TextonBoost Classifier. The F1 Value (%) is also provided for comparison with [4].

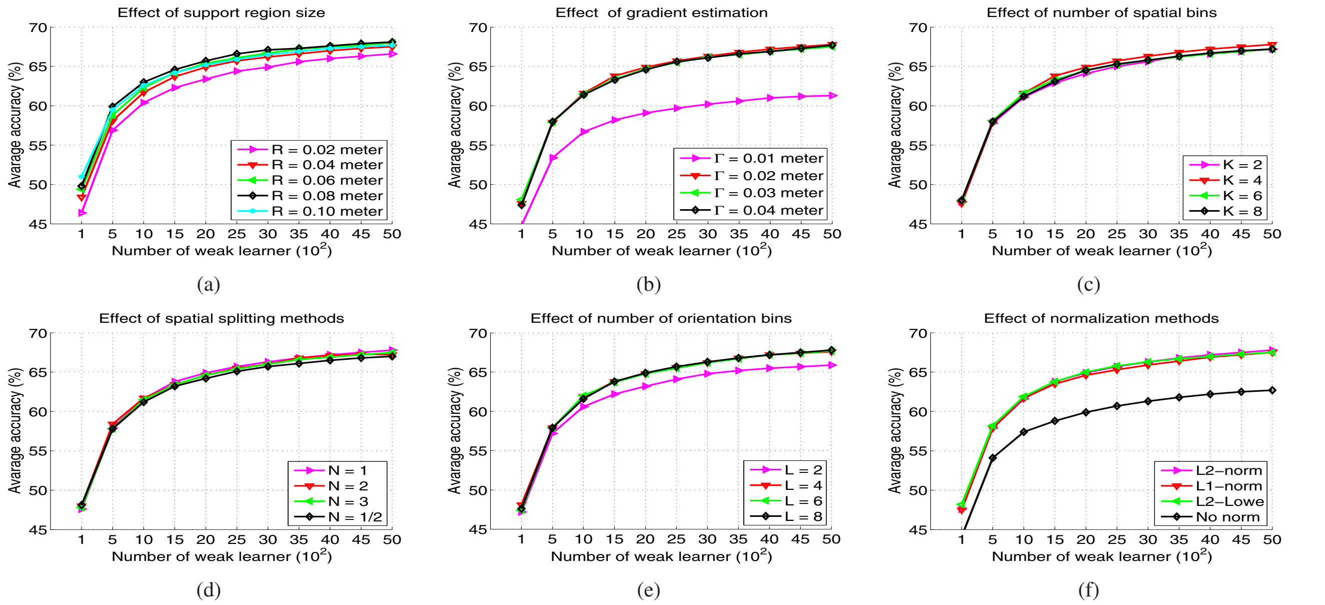
Feature	*Average*	*Global*	*F1 Value*
HIGH (Our)	**83.7** ±0.3	**91.9** ±0.5	**84.9** ±0.4
Width (Our)	67.8 ±0.2	81.2 ±0.6	68.6 ±0.3
Depth (Our)	76.7 ±0.6	85.5 ±0.7	78.7 ±0.7
SHOT	73.4 ±0.2	86.8 ±0.4	76.5 ±0.4
FPFH	70.3 ±0.6	81.4 ±0.2	70.2 ±0.3
SpinImage	79.5 ±0.5	90.9 ±0.4	82.5 ±0.3
3D shape context	78.5 ±0.6	89.6 ±0.4	81.2 ±0.6
SH + SHLR [4]	/	/	71.4
DH + SHLR [4]	/	/	71.5
SH + FM3N [4]	/	/	68.6
DH + FM3N [4]	/	/	70.8

5.4. Implementation and Performance Study

We now systematically study the implementation details of our HIGH feature. Throughout this section we refer results to our default HIGH feature which has the following properties: support region radius $R = 0.04$; gradient estimation region size $\Gamma = 0.02$; equally splitting radius R, spatial bins $K = 4$; gradient orientation bins $L = 8$; *L2-norm*. Each parameter is evaluated with other parameters fixed. Figure 6 summarizes the effects of various HIGH parameters on NYU Depth dataset. These will be examined in detail below. The main conclusions are that for good performance, one should use moderate size support region, relatively large region size for gradient estimation, relatively coarse spatial binning, fine orientation bins, and normalized feature vector.

Support Region Size. For HIGH feature, one major parameter is support region size which decides the number of neighboring points used to build the feature. Figure 6(a) shows the performance of HIGH feature with different sizes of support region. It can be seen that the classification performance increases significantly when radius R of support region changes from 0.02 meter to 0.04 meter. The performance increases slowly when radius R changes from 0.04 meter to 0.08 meter. However, the performance decreases when radius R is larger than 0.08 meter.

Gradient Computation. The height gradient of each point is computed using linear gradient reconstruction method [9] by considering neighbouring points whose distances to this point are less than Γ. The gradient value should catch the increase of height data and a relatively large region size is required. As Figure 6(b) shows, using region size $\Gamma = 0.02$ meter gives the best performance while $\Gamma = 0.03$ and $\Gamma = 0.04$ give similar results. On the other hand, reduce the region size to $\Gamma = 0.01$ decreases the performance significantly. The small region size pre-



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6. For details see the text. (a) Increasing the support region size can improve performance significantly until about $R = 0.04$ meter. (b) Reduce the region size for gradient estimation degrades performance significantly. (c) Increasing the number of spatial bins up to 4 bins can improve performance. (d) Equally splitting radius obtains the best result. (e) Increasing the number of orientation bins improves performance significantly up to about 4 bins. (f) The effect of different feature normalization schemes.

sumably makes the height gradient uninformative.

Spatial / Orientation Binning. HIGH feature descriptor is obtained by accumulating gradient magnitude of neighbouring points into bins of a two-dimensional histogram. Relative coarse spatial binning turns out to be essential for good performance. As Figure 6(c) shows, using 4 spatial bins gives the best performance but using 2 or 6 bins reduces the performance. Besides equally splitting radius R, we further test different spatial dividing methods by obtain the spatial index s in Eq.4 as $s = (\frac{d(\mathbf{q})}{R})^N K$. Figure 6 (d) shows the performance of different N while $N = 1$ slightly outperforms other selections. As Figure 6 (e) shows, increasing the number of orientation bins improves performance significantly up to about 4 bins, but makes little difference beyond this.

Descriptor Normalization. Depth gradient strengths vary over a wide range owing to local variations in object shape, partial occlusion, etc., so effective local contrast normalization turns out to be essential for good performance. We evaluated three different normalization schemes for HIGH feature descriptor. Let $\mathbf{V}$ be the unnormalized descriptor vector, $\|\mathbf{V}\|_k$ be its k-norm for $k = 1, 2$, and ϵ be a small constant. The schemes are: (a) *L2-norm*, $\mathbf{V} \rightarrow \mathbf{V}/\sqrt{\|\mathbf{V}\|_2^2 + \epsilon}$; (b)*L2-Lowe*, L2-norm followed by clipping (limiting the maximum values of $\mathbf{V}$ to 0.2) and renormalizing, as SIFT feature [22]; (c) *L1-norm* $\mathbf{V} \rightarrow \mathbf{V}/\|\mathbf{V}\|_1 + \epsilon$. Figure 6(f) shows that *L2-norm*, *L2-Lowe* and *L1-norm* all perform equally well. Omitting normalization significantly reduces the performance.

Table 3. Computational cost (seconds) of different methods to extract 3D features for 10,000 points. The time cost of each feature is the average result of 100 pointclouds. The dimension of each feature descriptor is also reported.

	HIGH	SpinImage	FPFH	SHOT	3D shapecontext
Time cost	0.45	0.41	0.16	1.10	4.11
Dimension	32	153	33	352	64

5.5. Computational Cost of HIGH Feature

In this experiment, we compare the computational cost of HIGH feature with several state-of-the-art 3D features: SpinImage [14], 3D shape context [12], FPFH [29], SHOT [38]. All these features are implemented using the same version of Point Cloud Library (PCL). Table 3 compares the performance of HIGH feature with other 3D features on a Xeon 2.67 GHz PC. It can be seen that HIGH feature has similar computational cost as SpinImage. Both HIGH and SpinImage are more efficient than 3D shape context and SHOT. The advantage of FPFH is that it is three times faster than HIGH and SpinImage.

5.6. Limitation of HIGH Feature

As the height gradient is used for descriptor building, the HIGH feature is sensor-view dependent. In addition, the height dimension data is not good for describing the small-scale objects, e.g. a small object placed on the table.

6. Conclusions

To capture the 3D shape information of the point-cloud data, we propose to extract height gradient histogram (HIGH) features by employing the height dimension data of 3D pointcloud. We have shown that the proposed HIGH feature can provide promising results for 3D scene labeling. We also systematically study the influence of various descriptor parameters. Our work clearly reveals the importance of the height dimension data for 3D scene description when comparing with depth dimension data. Our future work includes several aspects. Firstly, to handle generic 3D tasks, we will evaluate our feature extraction framework with other 3D point properties such as gradient of curvature [43] and gradient of intensity [20]. Secondly, instead of manually setting parameters of HIGH feature, we will learn their values for each dataset as [28]. Thirdly, we will evaluate the HIGH feature on other applications, e.g., 3D SLAM system [30].

Acknowledgment

This work is supported in part by a MoE Tier-1 project to Junsong Yuan. We thank Jingjing Meng to help proof read the paper.

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, pages 1626–1633, 2011.

[2] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino. 3d geometric scale variability in range images: Features and descriptors. *IJCV*, 99(2), Sept. 2012.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, June 2008.

[4] J. Behley, V. Steinhage, and A. Cremers. Performance of histogram descriptors for the classification of 3d laser range data in urban environments. In *ICRA*, 2012.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002.

[6] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[7] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, 2010.

[8] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *IJCV*, 25(1):63–85, Oct. 1997.

[9] C. D. Correa, R. Hero, and K.-L. Ma. A comparison of gradient estimation methods for volume rendering on unstructured meshes. *TVCG*, 17(3):305–319, May 2011.

[10] T. Darom and Y. Keller. Scale-invariant features for 3-d mesh models. *TIP*, 21(5):2758–2769, 2012.

[11] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *TPAMI*, 25(10):1285–1295, Oct. 2003.

[12] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[13] A. Golovinskiy, V. G. Kim, and T. A. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *ICCV*, pages 2154–2161, 2009.

[14] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *TPAMI*, 21(5), 1999.

[15] Y. Ke, R. Sukthankar, and M. Hebert. Volumetric features for video event detection. *IJCV*, 88(3):339–362, 2010.

[16] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas. Acquiring 3d indoor environments with variability and repetition. *ACM Trans. Graph.*, 31(6), Nov. 2012.

[17] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3d surf for robust three dimensional classification. In *ECCV*, pages 589–602, 2010.

[18] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[19] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *ICCV*, 1988.

[20] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *TPAMI*, 27, 2005.

[21] K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, T. Brox, and O. Ronneberger. Rotation-invariant hog descriptors using fourier analysis in polar and spherical coordinates. *Int. J. Comput. Vision*, 106(3):342–364, Feb. 2014.

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[23] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1), June 2001.

[24] A. S. Mian, M. Bennamoun, and R. A. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *IJCV*, 66(1):19–40, Jan. 2006.

[25] D. Munoz, J. A. Bagnell, and M. Hebert. Co-inference machines for multi-modal scene analysis. In *ECCV*, 2012.

[26] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. In *CVPR*, 2009.

[27] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29, 1996.

[28] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012.

[29] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[30] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: simultaneous localisation and mapping at the level of objects. In *CVPR*, pages 1352–1359, 2013.

[31] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), Jan. 2009.

[32] A. J. Y. Siddharth Manay, Byung-Woo Hong and S. Soatto. Integral invariant signatures. In *ECCV*, pages 87–99, 2004.

[33] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV Workshop*, 2011.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. ECCV, 2012.

[35] F. Stein and G. Medioni. Structural indexing: Efficient 3-d object recognition. *TPAMI*, 14(2):125–145, Feb. 1992.

[36] Y. Sun and M. Abidi. Surface matching by 3d points fingerprint. In *ICCV*, volume 2, 2001.

[37] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[38] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, 2010.

[39] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *ICCV*, 2003.

[40] J. P. C. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. S. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *CVPR*, pages 2067–2074, 2013.

[41] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *CVPR*, 2008.

[42] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). In *CVPR*, 2008.

[43] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *CVPR*, 2009.

[44] G. Zhao, X. Xiao, J. Yuan, and G. W. Ng. Fusion of 3d-lidar and camera data for scene parsing. *J. Vis. Comun. Image Represent.*, 25(1):165–183, Jan. 2014.

[45] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *ICCV workshop*, 2009.