sensors

MDPI

Article

New Compact 3-Dimensional Shape Descriptor for a Depth Camera in Indoor Environments

Hyukdoo Choi [1,2] **and Euntai Kim** [1,*]

[1] School of Electrical & Electronic Engineering, Yonsei University, Seoul 03722, Korea; goodgodgd@yonsei.ac.kr

[2] LG Electronics, Seoul 08592, Korea

* Correspondence: etkim@yonsei.ac.kr; Tel.: +82-2-2123-7729

Academic Editor: Vittorio M. N. Passaro

Received: 31 December 2016; Accepted: 11 April 2017; Published: 16 April 2017

Abstract: This study questions why existing local shape descriptors have high dimensionalities (up to hundreds) despite simplicity of local shapes. We derived an answer from a historical context and provided an alternative solution by proposing a new compact descriptor. Although existing descriptors can express complicated shapes and depth sensors have been improved, complex shapes are rarely observed in an ordinary environment and a depth sensor only captures a single side of a surface with noise. Therefore, we designed a new descriptor based on principal curvatures, which is compact but practically useful. For verification, the CoRBS dataset, the RGB-D Scenes dataset and the RGB-D Object dataset were used to compare the proposed descriptor with existing descriptors in terms of shape, instance, and category recognition rate. The proposed descriptor showed a comparable performance with existing descriptors despite its low dimensionality of 4.

Keywords: shape descriptor; correspondence; point cloud; principal curvature

1. Introduction

RGB-D sensors with affordable prices and decent performance have been available since 2010, and a new era in 3D computer vision and robotics has begun. There has been tremendous progress in research dealing with 3D data such as human pose and gesture recognition [1,2], point cloud registration [3,4], simultaneous localization and mapping (SLAM) [5], and object recognition [6]. In these studies, a vector that encodes distinctive property of local region, called a descriptor, plays an important role where descriptors are usually used to find correspondences [3,4,7,8] between two images or to encode a higher level descriptor for objects or scenes.

For 2D images, a number of descriptors have been proposed. After the monumental work of SIFT [9] and SURF [10], many researchers have competed for the best descriptor in terms of distinctiveness, processing time, and robustness to changes in transformation, noise, and illumination. BRISK, BRIEF, and FREAK [11–13] are currently popular because of their lower computational burden and better matching performance.

For 3D point cloud, shape descriptors adopted the legacy of 2D descriptors to encode the local shape of a point cloud. Popular options are Spin Image [14], fast point feature histogram (FPFH) [15], signature of histograms of orientations (SHOT) [16], and Tri-Spin-Image (TriSI) [17]. We tested these descriptors, but found they were not as discriminative as the 2D descriptors despite their lengthy descriptor sizes. Another important observation was that the local shapes from RGB-D sensors had limited complexity compared to 2D image patches. We concluded that existing shape descriptors have many redundant dimensions, and this motivated us to design a new descriptor that was as short as possible but as effective as existing descriptors.

The purpose of this study is three-fold: (1) to determine why existing shape descriptors become redundantly lengthy; (2) to analyze the cause of and to quantify the redundancy of high-dimensional descriptors; and (3) to propose a new efficient descriptor and prove the effectiveness of the new descriptor through experimentation. The proposed descriptor is derived from principal curvatures and concatenated with the gradients of curvatures. We found that this 4D descriptor was as effective as those mentioned above.

The paper is organized as follows: Section 2 reviews the existing shape descriptors in a historical context to determine the origin of the redundancy and proposes a new approach for local shape descriptors. A descriptor reflecting the new approach is presented in Section 3 and is compared with existing descriptors in terms of discriminative power in Section 4, followed by conclusions.

2. Derivation of a New Shape Descriptor

To understand the popularity of lengthy shape descriptors, we reviewed the historical progress from 2D image descriptors to 3D shape descriptors. Then we statistically evaluated the redundancy of the shape descriptors, and presented here a new approach for local shape descriptors.

2.1. Legacy of 2D Image Descriptors

Most of the image descriptors mentioned in Section 1 typically have dimensionality of 128 or 256. A dimensionality of 2^n is preferred for processing and memory efficiency. Previous studies have proven that correspondence matching accuracy tends to increase with dimensionality, but saturates around 128 or 256 [10,12]. From these results, we think a 'myth' was created that greater dimensionality resulted in better performance. However, as we proved in Section 2.3, this is not true for shape descriptors.

2.2. Brief Review of Shape Descriptors

Although many local shape descriptors have been proposed, we review here only a selection of notable works comparable to our descriptor. A thorough review of local shape descriptors was given in [18].

Spin Image [14] is one of the most famous shape descriptors. It encodes neighboring points as a radial distance from a normal line through a keypoint and a height from the tangent plane in a cylindrical local frame. Spin Image was built by constructing a 2D histogram of the distances and heights of the neighboring points. Several methods to improve the discriminative power of Spin Image have been proposed [17,19,20]. Pasqualotto et al. [20] proposed to combine color and shape Spin Images to compare two 3D models where similarities of two types of Spin Images are aggregated by fuzzy logic. TriSI [17] is one of the latest variations. To estimate TriSI, three spin images are computed along the three axes of the local reference frame (LRF) and concatenated into a single vector, and then the dimensionality of the vector is reduced by the principal component analysis (PCA) approach.

The point feature histogram (PFH) [21] is another important descriptor and the basis of the more popular descriptor, FPFH [15]. For every pair of points in the vicinity of a keypoint, a unique LRF is estimated, and then the three angular signatures of the pair are computed. The three histograms are made from the three signatures independently and concatenated to output the PFH descriptor vector. Since the PFH is computationally expensive, FPFH was proposed to reduce complexity and is now one of the most popular descriptors. The differential FPFH (dFPFH) descriptor [22] is the latest variation that captures surface irregularities by concatenating a difference vector between the FPFHs of inner and outer spheres.

The SHOT descriptor [16] partitions the local spherical space by the azimuth, elevation, and radial distance in the keypoint's LRF. A histogram of angles between the neighbor's normal vectors and the z-axis of the LRF is constructed for each space bin, and then the histograms are quadralinearly interpolated and concatenated to complete the SHOT descriptor. It outperformed Spin Image and FPFH in keypoint matching, but it is vulnerable to variations of resolution. SHOT is readily expandable to RGB-D data by adopting a texture-based histogram called color SHOT (CSHOT) [23].

Although the popular shape descriptors have colored versions, we do not discuss them. Color information is easily adopted to shape descriptors simply by concatenating shape and texture descriptors.

Other notable individual works include 3D shape context [24], intrinsic shape signature [25], rotational projection statistics [26], normal aligned radial feature [27], and binary robust appearance and normal descriptor (BRAND) [28]. Among them, BRAND has a similar motivation to ours and pursues a robust, fast, and memory efficient descriptor applicable to implementation in mobile phones and embedded systems. The descriptor consists of 256 binary relationships between 256 pairs of points in the vicinity of a keypoint. The dimensionality is high, but it takes only 32 bytes. Despite its memory efficiency, it outperformed CSHOT and Spin Image in keypoint matching.

Recently, data-driven approaches such as the convolutional neural network (CNN) have become popular to automatically extract discriminative features. The CNN approaches have successfully detected and classified objects [29–31] from images where the output of CNN usually worked as a global descriptors of objects. Some works [32,33] tried to use CNN for local patch retrieval. Outputs of middle layers of a network trained for object classification were used as patch descriptors. However, to our best knowledge, CNN features as a local descriptor have been tried only for RGB images but not for local shapes. Therefore, CNN features are still in different domain from local shape descriptors and more suitable for global descriptors. Though there are a number of global descriptors [34–36] to query the entire object, they are out of the scope of this study. Local descriptors have a unique role, for instance, point-to-point matching in point cloud registration.

2.3. Redundancy of Shape Descriptors

To analyze the amount of redundancy of shape descriptors, we estimated the number of effective principal components (nEPC). The nEPC counts the number of eigenvalues of a descriptor covariance matrix which are larger than $\lambda_1 * 0.01$, where λ_1 is the largest eigenvalue. The nEPC roughly estimates how many dimensions were actually used. Three RGB-D image sequences in the CoRBS dataset were used to estimate the nEPC: Cabinet, Desk, and Human [37]. The description of the dataset is given in Section 4.1. In each sequence, the five types of descriptors, FPFH, SHOT, Spin Image, TriSI and BRAND were extracted from 10,000 randomly sampled points, and the eigenvalues were computed from the covariance matrix of descriptors for each descriptor type in each sequence. Table 1 shows the results, where rEPC stands for the ratio of effective principal components.

Table 1. Number of effective principal components.

		FPFH	**SHOT**	**Spin Image**	**TriSI**	**BRAND**
Dimensionality		33	352	153	459	256
Cabinet	nEPC	7	60	30	54	205
	rEPC (%)	21.2	17.0	19.6	11.8	44.7
Desktop	nEPC	7	59	34	60	205
	rEPC (%)	21.2	16.8	22.2	13.1	44.7
Human	nEPC	8	56	35	64	205
	rEPC (%)	24.2	15.9	22.9	13.9	44.7

Two observations were made. For the first four descriptors from FPFH to TriSI, rEPCs were generally low from 5.2% to 27.3% which indicates that the descriptors occupy excessive memory with no effect. Their nEPCs grew with increment of dimensionalities, but the rEPCs decreased. Consequently the information in each dimension shrank as descriptor length increased. Second, BRAND had exceptionally high rEPC despite its high dimensionality because a BRAND descriptor is comprised of 256 independent binary tests while the others have highly correlated dimensions. However it still wasted more than half of total dimensions. Therefore, higher dimensionality provided limited benefit considering its excessive memory burden.

2.4. New Approach for Shape Descriptors

Here we propose a new approach for local shape descriptors specialized in recognizing shapes captured by affordable depth sensors in an ordinary environment. First, shape descriptors do not have to express highly complicated shapes. Object surfaces are not so complex in ordinary environments such as houses, offices and stores. Most of them are almost flat or simply curved. In addition, although depth resolution of depth cameras has been improved, there is still noise more than a few millimeters depending on situation [38] and a depth camera only captures a single side of a scene. Complex objects usually have small and peaky parts but they look relatively simple in a depth image. That is the reason that many previous works [14,16,17,22,26] used finely reconstructed 3D models from publically available datasets.

Second, since keypoint matching is not useful for shapes, short descriptors are preferred. In RGB images, where pixel values are highly dynamic, key points are readily detected and tracked. On the contrary, depths usually change smoothly over a continuous surface, so detecting and tracking key points is relatively difficult. Local shapes are rarely unique and peaky shapes easily look different from a different view pose. Accordingly, dense matching is more effective than key point matching thus descriptors have to be computed at almost all points. To compare crowded descriptors with limited processing power and memory, like in mobile phones or robots, short descriptors are more advantageous.

Third, local descriptors should be easy to understand, implement and reproduce. Since computing local descriptors is just beginning of an application, it should not burden users. The popular shape descriptors [14–16] are easy to understand and require no pre-training or pre-processing more than a smoothing filter and computing normal vectors. On the other hand, for instance, TriSI is not popularly used despite its better performance than Spin Image or SHOT, because it requires a pre-training stage to compress descriptor size by the PCA approach. Pre-training makes the performance of an algorithm to be dependent on training data, which is not always reliable and requires more effort. Following the new approach, we present in Section 3 our new descriptor design based on principal curvatures.

3. Principal Curvatures with Gradients

We propose a novel shape descriptor, principal curvatures with gradients (PCWG), which is four dimensional but as effective as existing descriptors. The first two elements are the principal curvatures of the surface, and the second two are the gradients of the principal curvatures along two principal directions. The idea of utilizing principal curvatures for correspondence matching is not completely novel, as proposed in [39] where curvatures are used for outlier rejection in ICP. However, in our method principal curvatures work as a shape descriptor for the first time along with curvature gradients.

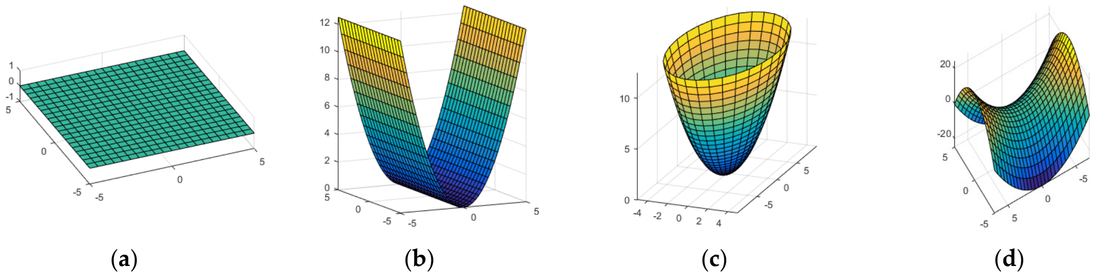
Furthermore, we present a novel method to accurately estimate principal curvatures by formulating the estimation problem as quadratic programming (QP). Cheng et al. proposed a principal curvature estimation method based on normal fitting [40]. The curvature values were optimized to fit the normal vectors on the surface rather than the point coordinates, but it required too many intermediate variables. Recently, Spek et al. presented a fast method to estimate curvatures by solving a non-linear optimization problem where the curvature values and normal estimation were iteratively refined [41]. On the contrary, our formulation is derived in an intuitive manner and solved by a closed-form equation.

3.1. Curvature Estimation

Provided that an arbitrary point cloud within a certain radius from a specific point is lying on a continuous surface, the intrinsic shape of the point cloud can be modeled by a quadratic surface. The basic form of a quadratic surface is given as follows:

$$z = \frac{1}{2}\left(\mathcal{C}_{\alpha}x^2 + \mathcal{C}_{\beta}y^2\right) \tag{1}$$

where $\mathcal{C}_{\alpha}$ and $\mathcal{C}_{\beta}$ are the primary and secondary curvatures of the surface, respectively, and $|\mathcal{C}_{\alpha}| \geq |\mathcal{C}_{\beta}|$. By varying the only two curvatures and applying a rotation, it can express various shapes as depicted in Figure 1. This expression is simple, descriptive, and comprehensive.



(**a**) (**b**) (**c**) (**d**)

Figure 1. Quadratic surfaces from: (**a**) Plane ($\mathcal{C}_{\alpha} = \mathcal{C}_{\beta} = \mathbf{0}$); (**b**) Parabola ($\mathcal{C}_{\alpha} > \mathcal{C}_{\beta} = \mathbf{0}$); (**c**) Elliptic paraboloid ($\mathcal{C}_{\alpha} > \mathcal{C}_{\beta} > \mathbf{0}$); and (**d**) Hyperbolic paraboloid ($\mathcal{C}_{\alpha} < \mathbf{0}$, $\mathcal{C}_{\beta} > \mathbf{0}$).

Let us derive the curvatures from a given point cloud. When there is a center point $\mathbf{p}_k$ and a normal vector $\mathbf{n}_k$ at $\mathbf{p}_k$, the point cloud around the center point within a radius r is denoted by $\mathbb{P} = \{\mathbf{p}_j\}_{j=1:N}$ where $\mathbf{p}_j$ is a neighboring point around $\mathbf{p}_k$. In this subsection, let us assume that the center point is translated to the origin for simplicity, that is, $\mathbf{p}_k \to 0$ and $\mathbf{p}_j - \mathbf{p}_k \to \mathbf{p}_j$. To generalize (1), it is rewritten in the matrix form and an arbitrary rotation is applied:

$$\mathbf{p}^T\mathbf{R}\mathbf{A}_0\mathbf{R}^T\mathbf{p} - \mathbf{b}_0^T\mathbf{R}^T\mathbf{p} = 0 \tag{2}$$

$$\mathbf{p}^T\mathbf{A}\mathbf{p} - \mathbf{b}^T\mathbf{p} = 0 \tag{3}$$

where $\mathbf{A}_0 = diag\left(\begin{array}{ccc} \mathcal{C}_{\alpha} & \mathcal{C}_{\beta} & 0 \end{array}\right)$, $\mathbf{b}_0 = \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right]^T$, and $\mathbf{R} \subseteq \mathbf{SO}(3)$ is a rotation matrix. Since $\mathbf{A} = \mathbf{R}\mathbf{A}_0\mathbf{R}^T$, it can be seen as eigen decomposition of $\mathbf{A}$ where the eigen values are $\mathcal{C}_{\alpha}$, $\mathcal{C}_{\beta}$, and 0 and the corresponding eigen vectors are the columns of $\mathbf{R} = \left[\begin{array}{ccc} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{array}\right] \subseteq \mathbf{SO}(\mathbf{3})$. Thus, $\mathbf{b} = \mathbf{R}\mathbf{b}_0$ becomes $\mathbf{b} = \mathbf{v}_3$, which corresponds to the zero eigenvalue. The general equation is constrained by $\mathbf{A}\mathbf{b} = \mathbf{A}\mathbf{v}_3 = \mathbf{0}_{3\times1}$, $\mathbf{b}^T\mathbf{b} = \mathbf{v}_3^T\mathbf{v}_3 = 1$, and $\mathbf{A}^T = \mathbf{A}$. The last constraint is for the orthonormal decomposition ($\mathbf{R}^T\mathbf{R} = \mathbf{I}$). Our first goal is to estimate the optimal $\mathbf{A}$ and $\mathbf{b}$ such that:

$$\begin{aligned} &\mathbf{A}, \mathbf{b} = \underset{\mathbf{A},\mathbf{b}}{\mathrm{arg}min} \sum_j \left\{\mathbf{p}_j^T\mathbf{A}\mathbf{p}_j - \mathbf{b}^T\mathbf{p}_j\right\} \\ &subject\ to\ \mathbf{A}\mathbf{b} = \mathbf{0}_{3\times1},\ \mathbf{b}^T\mathbf{b} = 1,\ \mathbf{A}^T = \mathbf{A}. \end{aligned} \tag{4}$$

The curvatures and rotation matrix are computed by eigen decomposition of $\mathbf{A}$. This problem belongs to quadratically constrained quadratic programming (QCQP), but it is not convex because of the non-convex constraints. General QCQP can be relaxed to be semidefinite programming, but it needs a formulation with huge matrices and hence is burdened by a heavy computational load. Since this method opposes our intentions, we performed a trick to ease the problem.

From the geometrical intuition, we replaced $\mathbf{b}$ with the normal vector $\mathbf{n}_k$ since the vector $\mathbf{n}_k$ corresponds to the z-axis for a quadratic surface. Consequently, Eq. (4) is simplified as:

$$\begin{aligned} &\mathbf{A} = \underset{\mathbf{A}}{\arg min} \sum_j \left\{ \mathbf{p}_j^T \mathbf{A} \mathbf{p}_j - \mathbf{n}_k^T \mathbf{p}_j \right\} \\ &\textit{subject to}\, \mathbf{A}\mathbf{n}_k = \mathbf{0}_{3\times 1},\, \mathbf{A}^T = \mathbf{A} \end{aligned} \tag{5}$$

Now we have a simple QP problem that is convex and has a closed-form solution. To solve the problem in the QP form, $\mathbf{A}$ is vectorized as follows:

$$\begin{aligned} &\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^T \\ &\textit{where}\, \mathbf{A} = \begin{bmatrix} a_1 & a_4 & a_6 \\ a_4 & a_2 & a_5 \\ a_6 & a_5 & a_3 \end{bmatrix} \end{aligned} \tag{6}$$

The objective function of Equation (5) is reformulated with the vector $\mathbf{a}$ as follows:

$$\begin{aligned} &\sum_j \left\{ \mathbf{p}_j^T \mathbf{A} \mathbf{p}_j - \mathbf{n}_k^T \mathbf{p}_j \right\} \\ &= \sum_j \{ \mathbf{f}(\mathbf{p}_j)\mathbf{a} - \mathbf{n}_k^T \mathbf{p}_j \} \\ &= (\mathbf{F}(\mathbf{p}_{1:N})\mathbf{a} - \mathbf{P}\mathbf{n}_k)^T (\mathbf{F}(\mathbf{p}_{1:N})\mathbf{a} - \mathbf{P}\mathbf{n}_k) \end{aligned} \tag{7}$$

where:

$$\mathbf{f}(\mathbf{p}) \triangleq \begin{bmatrix} x^2 & y^2 & z^2 & 2xy & 2yz & 2zx \end{bmatrix} \mathbf{F}(\mathbf{p}_{1:N}) \triangleq \begin{bmatrix} \mathbf{f}(\mathbf{p}_1) \\ \mathbf{f}(\mathbf{p}_2) \\ \vdots \\ \mathbf{f}(\mathbf{p}_N) \end{bmatrix} \textit{and}\, \mathbf{P} \triangleq \begin{bmatrix} \mathbf{p}_1^T & \mathbf{p}_2^T & \cdots & \mathbf{p}_N^T \end{bmatrix}^T.$$

Similarly, the constraint equation is rewritten as $\mathbf{A}\mathbf{n}_k = \mathbf{G}(\mathbf{n}_k)\mathbf{a} = \mathbf{0}_{3\times 1}$ where:

$$\mathbf{G}(\mathbf{n}_k) \triangleq \begin{bmatrix} \mathbf{n}_{k,x} & 0 & 0 & \mathbf{n}_{k,y} & 0 & \mathbf{n}_{k,z} \\ 0 & \mathbf{n}_{k,y} & 0 & \mathbf{n}_{k,x} & \mathbf{n}_{k,z} & 0 \\ 0 & 0 & \mathbf{n}_{k,z} & 0 & \mathbf{n}_{k,y} & \mathbf{n}_{k,x} \end{bmatrix}.$$

The final optimization formula is rearranged as the following equation:

$$\begin{aligned} &\mathbf{a} = \underset{\mathbf{a}}{\arg min} (\mathbf{F}(\mathbf{p}_{1:N})\mathbf{a} - \mathbf{P}\mathbf{n}_k)^T (\mathbf{F}(\mathbf{p}_{1:N})\mathbf{a} - \mathbf{P}\mathbf{n}_k) \\ &\textit{subject to}\, \mathbf{G}(\mathbf{n}_k)\mathbf{a} = \mathbf{0}_{3\times 1} \end{aligned} \tag{8}$$

The QP problem in this form is solved by the closed-form formula:

$$\begin{aligned} &\begin{bmatrix} \mathbf{F}(\mathbf{p}_{1:N})^T \mathbf{F}(\mathbf{p}_{1:N}) & \mathbf{G}(\mathbf{n}_k)^T \\ \mathbf{G}(\mathbf{n}_k) & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \lambda \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}(\mathbf{p}_{1:N})^T \mathbf{P}\mathbf{n}_k \\ \mathbf{0}_{3\times 1} \end{bmatrix}. \end{aligned} \tag{9}$$

The matrix $\mathbf{A}$ is reconstructed from the solution $\mathbf{a}$ as in Equation (6) and decomposed to obtain $\mathcal{C}_\alpha$, $\mathcal{C}_\beta$, and $\mathbf{R}$, which are the principal curvatures at $\mathbf{p}_k$, and the rotation matrix of which columns are principal axes, respectively.

3.2. Curvatures with Gradients

Since real-world surfaces are not always symmetric like quadratic surfaces, two curvature values are not sufficient to describe skewed shapes. To cover the remaining degree of freedom of local surfaces, we estimated the gradients of curvatures. The gradients were estimated by linear regression of the curvatures along the two principal axes, $\mathbf{v}_1$ and $\mathbf{v}_2$. The primary curvature around the keypoint can be modeled as:

$$\mathcal{C}_\alpha = \mathcal{D}_\alpha \mathbf{v}_1 \cdot (\mathbf{p}_1 - \mathbf{p}_k) + c + \varepsilon \tag{10}$$

where $\mathcal{D}_\alpha$, c, and ε are the gradients of the primary curvature, offset, and fitting error, respectively. The gradient that minimizes the sum of the fitting errors is computed by solving:

$$\begin{bmatrix} \mathbf{v}_1 \cdot (\mathbf{p}_1 - \mathbf{p}_k) & 1 \\ \mathbf{v}_1 \cdot (\mathbf{p}_2 - \mathbf{p}_k) & 1 \\ \vdots & \vdots \\ \mathbf{v}_1 \cdot (\mathbf{p}_N - \mathbf{p}_k) & 1 \end{bmatrix} \begin{bmatrix} \mathcal{D}_{\alpha\text{,k}} \\ c \end{bmatrix} = \begin{bmatrix} \mathcal{C}_{\alpha,1} \\ \mathcal{C}_{\alpha,2} \\ \vdots \\ \mathcal{C}_{\alpha,N} \end{bmatrix} \tag{11}$$

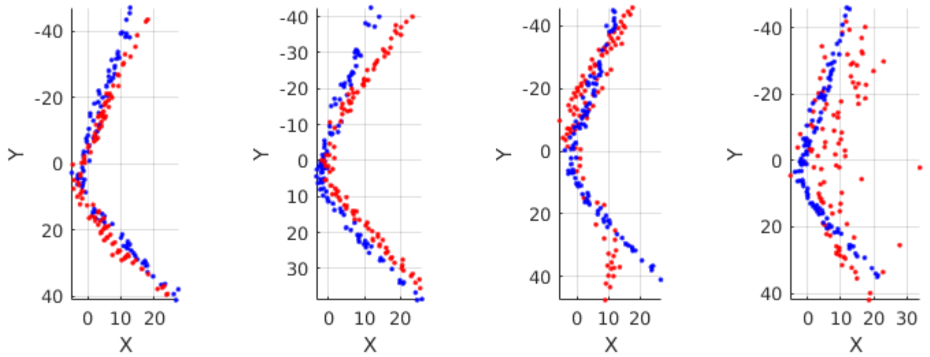
where $\mathcal{C}_{\alpha,j}$ is the primary curvature at the jth neighbor point. Similarly, the gradient of the secondary curvature, $\mathcal{D}_\beta$, is estimated by the linear regression over $\mathcal{C}_\beta$ along $\mathbf{v}_2$. While both curvature values are symmetric over positive or negative directions of principal axes, the signs of gradients differ with the directions of the principal axes. For consistency of gradient values, the direction of $\mathbf{v}_1$ was always selected to make $\mathcal{D}_\alpha$ positive. Then $\mathbf{v}_2$ was determined by $\mathbf{v}_2 = \mathbf{n}_k \times \mathbf{v}_1$. Therefore, our descriptor, PCWG, is presented as $\mathbf{d}_k = \begin{bmatrix} \mathcal{C}_\alpha & \mathcal{C}_\beta & \mathcal{D}_\alpha & \mathcal{D}_\beta \end{bmatrix}$. This descriptor expresses the sharpness of a shape by curvatures and the skewness by gradients. Generally, there are three types of shapes: (1) high gradient (the point is between flat and curvy regions); (2) low gradient and high curvature (the point is at the peak of the curve); and (3) low gradient and low curvature (the point is on the plane). We prove in Section 3.3 that these four parameters are sufficient to accurately describe local point clouds.

3.3. Gradient Weight

Although the gradient terms are auxiliary compared with the curvature terms of our PCWG descriptor, their values readily become larger than the curvatures and are even sensitive to noise, which makes the distance between descriptors to be distorted. As the two terms represent different properties, they do not have to be equally treated. Thus, the gradients should be weighted to suppress their effect on descriptor distances. Ideally, the distance between descriptors should be linearly related to the shape distance. The shape distance is defined as:

$$S_{ik} = \mathbf{D}(\mathbb{P}_i, \mathbb{P}_k) + \eta \mathbf{A}(\mathbb{N}_i, \mathbb{N}_k) \tag{12}$$

where $\mathbb{P}_i$ and $\mathbb{N}_i$ are the point cloud and normal vectors at the frame i, respectively, and $\mathbf{D}(\mathbb{P}_i, \mathbb{P}_k)$ is the mean point-to-plane distance between two point clouds, $\mathbf{A}(\mathbb{N}_i, \mathbb{N}_k)$ is the mean angular difference between the normal vector pairs of the point clouds, and η is the angular difference weight for a balance with the point-to-plane distance, which is set to 0.01. To convince the effectiveness of the distance metric, four different shapes (red) are compared with the reference shape (blue) in Figure 2 with the corresponding shape distances. From left to right, the compared shapes differs more with the reference shape with increasing shape distances. The effect of the angular difference weight on performances of descriptors is addressed in Section 4.5.



(**a**) (**b**) (**c**) (**d**)

Figure 2. Comparison of different shapes with shape distances. The blue dots represent the reference shape which is the same in the four figures while the red dots represent the different compared shapes. The corresponding shape distances are denoted by $\boldsymbol{S_{ik}}$ below the figures. (**a**) $S_{ik} = 1.88$; (**b**) $S_{ik} = 3.11$; (**c**) $S_{ik} = 5.98$; (**d**) $S_{ik} = 9.10$.

Based on the shape distance metric, the gradient weight, $\mathbf{w}$, needs to be adjusted to fit the linear regression $(\mathbf{d}_k - \mathbf{d}_i) \cdot \mathbf{w} = S_{ik}$ where $\mathbf{w} = \begin{bmatrix} \nu & \nu & \omega & \omega \end{bmatrix}$. The descriptor has different weights for the curvatures and gradients. Given a set of pairs of descriptors or local shapes, the weight can be optimized between the descriptor difference and the corresponding shape distance. The optimization problem is defined as:

$$\begin{array}{c} (\nu, \omega) = \underset{\nu,\omega}{\arg\min} \sum\limits_{i,k} \{(\mathbf{d}_k - \mathbf{d}_i) \cdot \mathbf{w} / S_{ik} - \mathbf{1}\} \\ \mathit{subject\ to}\, \nu, \omega > 0 \end{array} . \tag{13}$$

The relative weight for the gradient terms is calculated as $\acute{\omega} = \omega / \nu$. As the optimal weight value varies depending on the data from 0.2 to 0.5, we selected a fixed value of 0.3 for all evaluations in the following section. Therefore, the final form of PCWG is:

$$\mathbf{d}_k = \begin{bmatrix} \mathcal{C}_\alpha & \mathcal{C}_\beta & \acute{\omega}\mathcal{D}_\alpha & \acute{\omega}\mathcal{D}_\beta \end{bmatrix}. \tag{14}$$

In this section we have introduced how to compute the curvatures and their gradients by using the optimization techniques. The terms seem to be complicated but there are only two equations to solve, Equations (9) and (11), which are closed-form linear equations with no iterations. Besides, it is not computationally complex. Complexity of solving Equation (9) is linear with the number of neighbor points. Computing $\mathbf{F}(\mathbf{p}_{1:N})^T \mathbf{F}(\mathbf{p}_{1:N})$ is complex as $\boldsymbol{O}\left(\mathrm{ND}^2\right)$ where N is the number of neighbor points and D is the dimensionality of a point, which is a constant 3, and then subsequent solving linear equation (9 dimensional) and eigen decomposition of $\mathbf{A}$ (3 × 3 matrix) are finished in a constant time. The processing times with various parameters were measured and discussed in Section 4.2.

4. Evaluation Results

To prove the discriminative power of the proposed descriptor, it was compared with the five aforementioned descriptors using the public datasets. The performance of the descriptors was evaluated by multi-level recognition tests. The first level test was on shape recognition. It was a primitive performance for local shape descriptors to see how effectively descriptors distinguish between different local shapes. Thus, it is critical for point cloud registration or point-level association. The second and third level tests were on object instance and category recognitions, respectively, and they measured the statistical robustness of object-level description. Also, invariance of descriptors to noise and scale changes was addressed. We selected three values of 4, 5, and 6 cm for the support radius. A radius less than 4 cm results in unstable normal vectors. On the other hand, a radius larger than 6 cm is also undesirable because it becomes vulnerable to occlusions and cannot describe small shapes. In the following sections, we introduced the public datasets and existing descriptors, and compared them to our descriptor based on the evaluation results.

4.1. Datasets

We used three public RGB-D datasets, the CoRBS dataset [37] and the RGB-D Scenes dataset [42] for shape recognition and the RGB-D Object dataset [42] for object recognition. The properties of the datasets for shape recognition were summarized in Table 2. The CoRBS dataset was captured by Kinect v2 and we selected four video sequences which captured four different objects. In Table 2, the numbers after the scene name indicate the video ID to identify a specific video among the videos from the same scenes and the length in the fourth column means the length of a camera trajectory. The RGB-D Scenes dataset was captured by Kinect v1 in home and office environments. We selected five video sequences, which are the first videos from each scene.

Table 2. The properties of the datasets used for shape recognition.

Dataset (Sensor)	Scene Name	Dataset Index	Length (m)	# Frames
CoRBS (Kinect v2)	Electrical cabinet #2	1	23.0	1902
	Desk #2	2	11.5	2380
	Human #2	3	11.3	2547
	Racing car #2	4	34.1	3209
RGB-D Scenes (Kinect v1)	Desk #1	5	N/A	98
	Kitchen_small #1	6	N/A	180
	Meeting_small #1	7	N/A	180
	Table #1	8	N/A	125
	Table_small #1	9	N/A	199

The RGB-D Object dataset has a hierarchical structure of video sequences in four levels: category, instance, video, and frames. The dataset contains 300 objects which belongs to 51 categories, and there are multiple videos with hundreds of frames for each object taken at different viewpoints. The main advantage of the dataset is that various items usually observed in a home environment are included. The sample images from the datasets are shown in Figure 3.

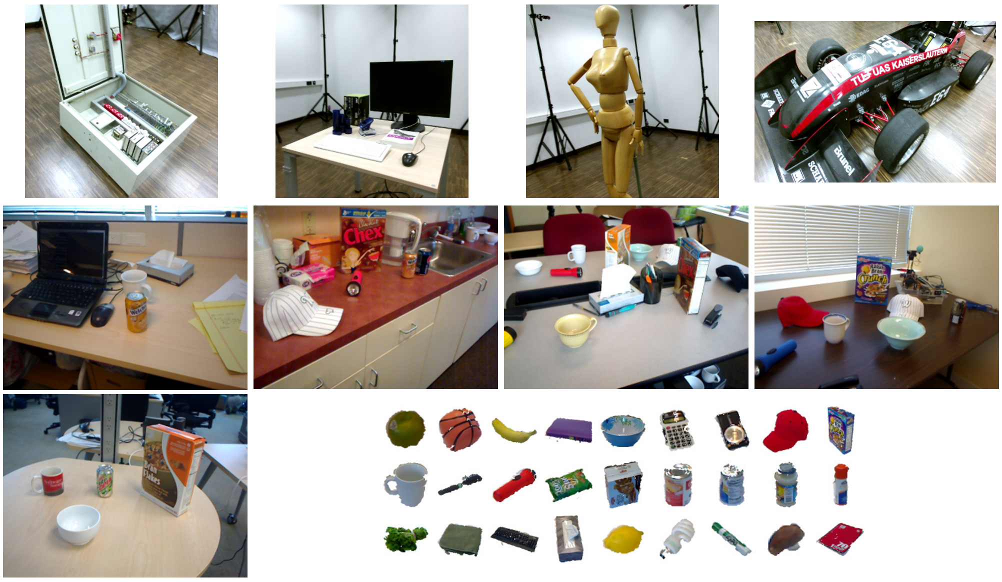


Figure 3. Sample images from the datasets: the top row shows the four objects from the CoRBS dataset; the next five scenes are from the RGB-D Scenes dataset; and the last figure shows exemplar images of RGB-D Object dataset.

4.2. Implementation Details

In the implementation, depth images are scaled down to a size of 320 × 240 in order to speed up the processing and suppress noise. As all the datasets provide 640 × 480 images, 2 × 2 depth pixels are averaged as a single depth value. Neighbor points around a keypoint were evenly sampled within the support radius to prevent curvatures from being biased to a denser region within the radius. The number of sampled points was less than or equal to 50. For efficiency of computation, neighbor search results were shared with both normal and descriptor estimation. The three components, neighbor searching, normal estimation, and descriptor estimation were implemented in OpenCL [43]. As the processing time might vary with a support radius and the maximum number of sampled points, we measured the processing time with different combinations of parameters with the GPU device of GTX 1080. The processing times for neighbor searching and normal estimation were less than 1 ms and they are ignorable for any combination of parameters. As normal vectors were commonly used by all the descriptors, we would analyze only the processing time of our descriptor.

As summarized in Table 3, descriptor estimation took several milliseconds. As expected, the processing time apparently increases with the maximum number of sampled points. The support radius was inversely related to the processing time. That is because the search area is more likely to be occluded with a larger radius and hence the effective number of sampled points decreases. The table shows that the proposed method is fast enough to run in real time regardless of the parameters, even with less powerful devices.

Table 3. Processing time (ms) of PCWG estimation vs. the support radius and the maximum number of sampled points.

Radius\#points	30	40	50
4	3.53	4.13	4.64
5	3.21	3.79	4.06
6	3.18	3.75	4.03

In Section 3.3, the gradient weight $\acute{\omega}$ was optimized by Equation (13) under the assumption that a distance between shape descriptors should be proportional to the corresponding shape distance. To see the validity of the assumption, shape recognition rates are evaluated while varying gradient weights as shown in Figure 4. The shape recognition rate (Precision-1) is defined in Section 4.4 in detail. The shape recognition rates are averaged over all the datasets in Table 2. The performance did not vary much with gradient weight but does with a support radius. As we expected in Section 3.3, the best result is obtained when the gradient weight is 0.3 and the radius is 5 cm. In addition, we can observe that the best radius is 5 cm and the smaller radius (4 cm) is less dependent on the gradient terms than the best radius.

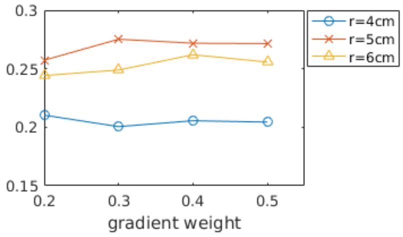


Figure 4. Shape recognition rates with different gradient weights and support radii.

4.3. Existing Descriptors

For comparison with the existing descriptors, we used the PCL implementation of FPFH [44], SHOT [16], and Spin Image [14]. Their default dimensionalities are 33, 352, and 153, respectively.

TriSI [17] was implemented by computing three spin images along the three principal axes, $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$, and concatenating them as a single vector. Guo et al. reduced the dimensionality of TriSI with the PCA approach, but we did not because the quality of the compression depended on the pre-training data which was difficult to standardize. We used the raw TriSI, and hence its dimensionality became 459. We implemented BRAND [28] by ourselves. The local binary pattern with a radius of 24 pixels was created by Gaussian distribution of $\mathrm{N}(0, 48^2/25)$ and texture information was not used but only geometric information was used for binary tests. The PCWG descriptor was also compared with the pure principal curvatures (PC) with a dimensionality of 2. Therefore, seven types of descriptors were compared in total.

4.4. Shape Recognition

As mentioned in Section 2.4, keypoint matching does not work effectively because local shapes are ambiguous. Instead, we developed a new method to evaluate local shape recognition performance. To evaluate how well the descriptors recognize a similar shape among various shapes, we needed to extract a set of *representative* shapes which are both frequently observed and as diverse as possible.

We extracted two sets of representative descriptors from each dataset: one was a reference set, and the other was a query set. The sets of pairs of representative descriptors and shapes were used to evaluate shape recognition rates. The samples of representative shapes were shown in Figure 2 where the blue shape is the reference shape and the red shapes are from the query set.

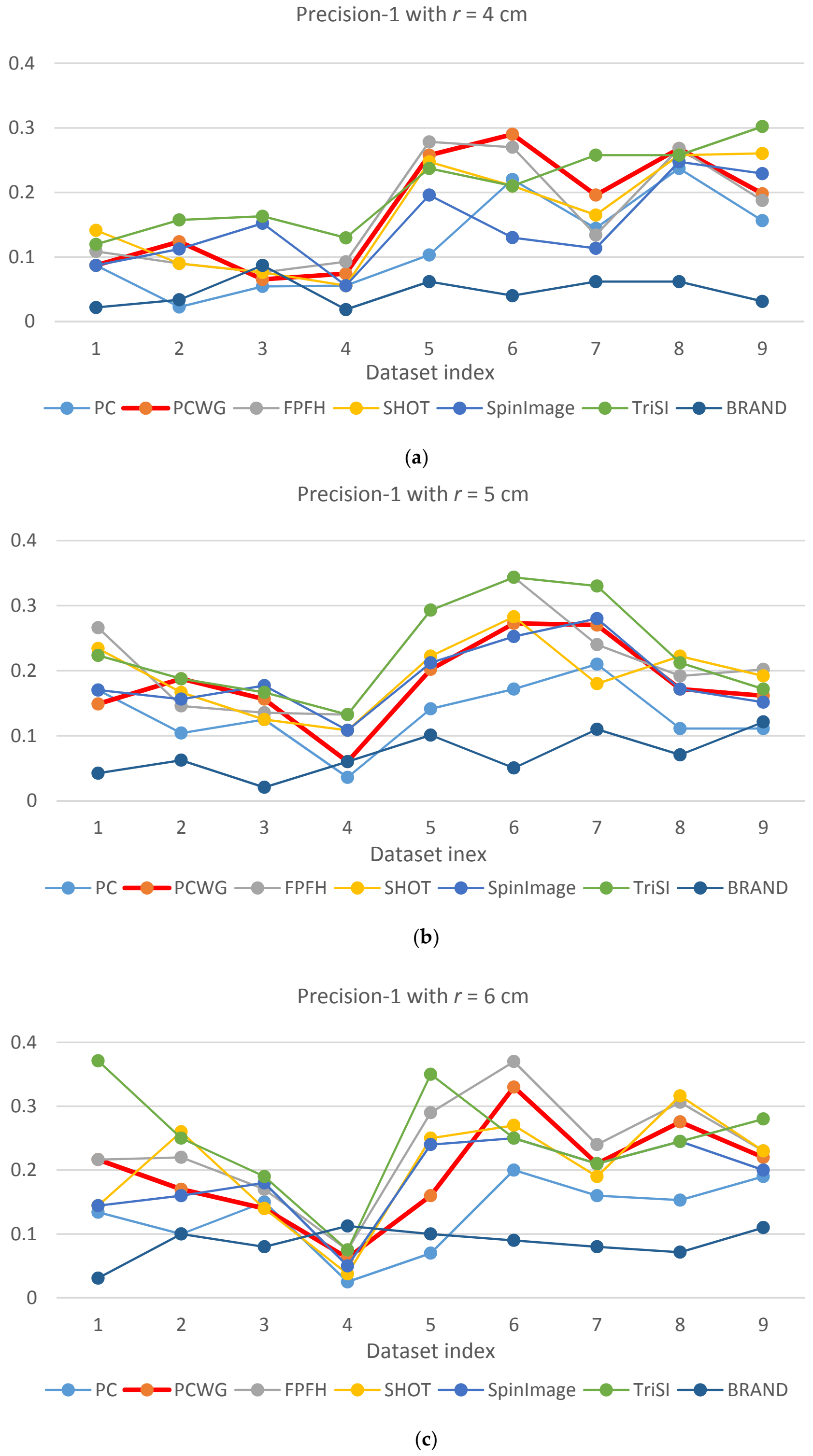
The representative shapes were found by clustering a huge pool of shape descriptors and selecting the centroids of the clusters. For fair competition among the descriptors, a concatenation of PCWG, FPFH, SHOT, and TriSI descriptors, named as a total descriptor, was used for the clustering. The principal curvatures and Spin Image were not used because they are included in the PCWG and TriSI descriptors, respectively. As naive clustering of the total descriptors is prone to be biased to dominant shapes (flat shapes), the iterative clustering with following steps was used:

1. Total descriptors are computed at the sampled points in frames of a video sequence.
2. Total descriptors are grouped by K-means clustering.
3. The dominant clusters are resampled to reduce the population.
4. Iterate from 2, until no dominant cluster exists.

In the first step, tens of points with more than 25 neighbor points were evenly sampled except for large planar regions in each frame. The number of clusters, K, was 100 in the second step. The dominant cluster was defined as a cluster with population more than $4T/K$, where T is the number of the total descriptors. In the third step, the dominant clusters were reduced to $2T/M$. To obtain the two sets of descriptors, the reference set was extracted first, and then the query set was selected after excluding the reference set from the pool of descriptors.

Given the two sets of representative descriptors, the correspondence to a queried descriptor was predicted by finding the closest descriptor in the reference set. Precision-1 refers to the ratio of the correspondences where the closest descriptor was from the closest shape in terms of the shape distance of Equation (12). Precision-5 means the ratio of the correspondences where the closest descriptor belongs to the five closest shapes. Correspondences with shape distances larger than the support radius were rejected in the evaluation. The two metrics were computed for the seven descriptors for the same sets of representative shapes.

Figures 5 and 6 show the evaluation results of Precision-1 and Precision-5, respectively, over the eight datasets with the three different support radii. In the figures, PCWG was denoted by the bold red lines. Overall, PCWG stayed in the middle of other descriptors. It means that PCWG's performance is comparable with the others despite its extremely low dimensionality. That is confirmed by Tables 4 and 5 where precision-1 and -5 were averaged over the nine datasets for each support radius and the last column shows the ranking of PCWG. In the tables, it is noteworthy that the relative performance of PCWG with a radius of 4 cm was ranked second and first in Precision-1 and Precision-5, respectively. It indicates that PCWG is good at roughly searching small shapes.

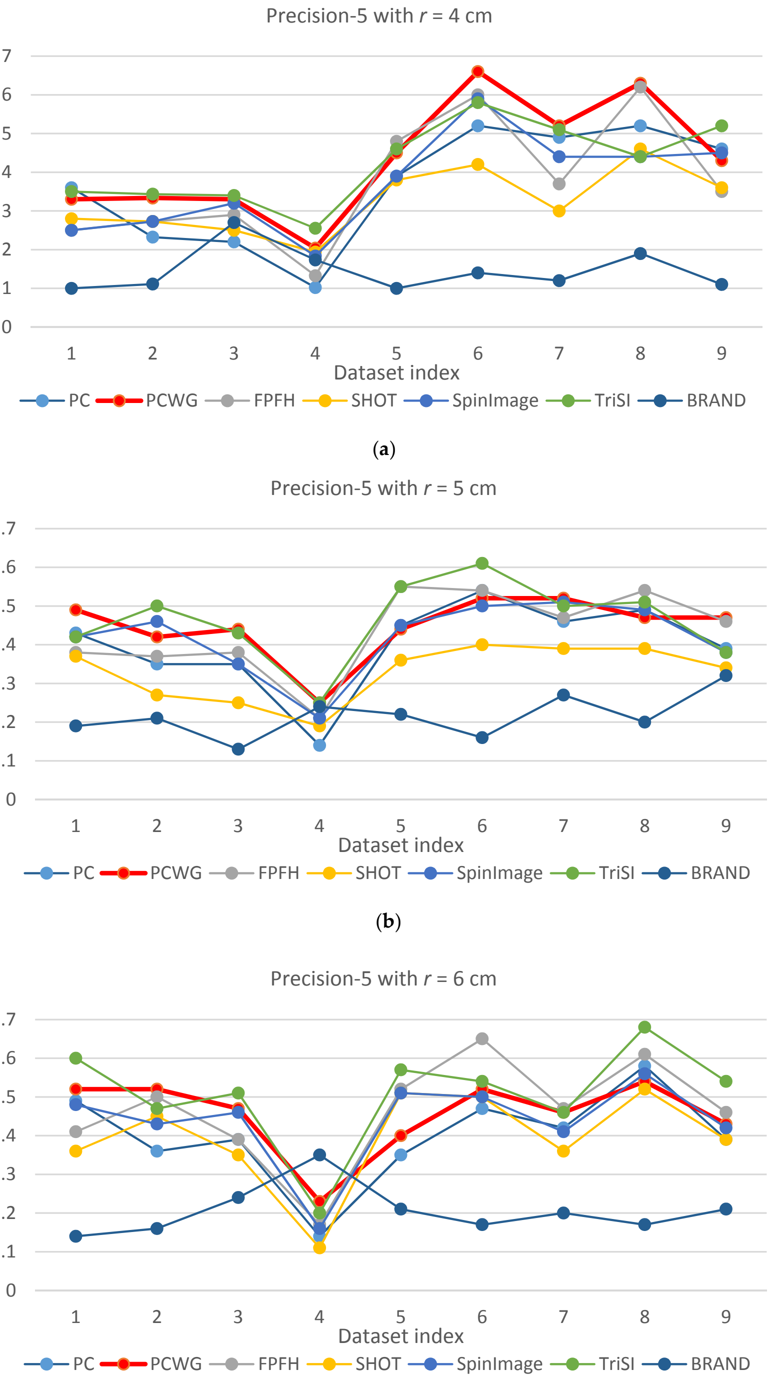


(**a**)

(**b**)

(**c**)

Figure 5. Precision-1 comparison of seven descriptors over nine datasets with different support radii: (**a**) r = 4 cm, (**b**) r = 5 cm, and (**c**) r = 6 cm.



(**a**)

(**b**)

(**c**)

Figure 6. Precision-5 comparison of seven descriptors over nine datasets with different support radii: (**a**) *r* = 4 cm, (**b**) *r* = 5 cm, and (**c**) *r* = 6 cm.

Table 4. Average precision-1 for three support radii.

Radius (cm)	PC	PCWG	FPFH	SHOT	SpinImage	TriSI	BRAND	Rank
4	0.120	0.173	0.167	0.167	0.147	0.204	0.046	2
5	0.131	0.181	0.217	0.193	0.187	0.229	0.071	5
6	0.131	0.198	0.235	0.204	0.187	0.247	0.086	4

Table 5. Average precision-5 for three support radii.

Radius (cm)	PC	PCWG	FPFH	SHOT	SpinImage	TriSI	BRAND	Rank
4	0.366	0.432	0.374	0.324	0.371	0.422	0.146	1
5	0.400	0.447	0.433	0.329	0.419	0.461	0.216	2
6	0.399	0.454	0.464	0.394	0.437	0.508	0.206	3

The best overall performance came from TriSI which is the most high-dimensional descriptor but the second best was FPFH of which dimensionality is just 33. On the other hand, the performance of SHOT and BRAND were disappointing since SHOT has the second largest dimensionality, 352, and BRAND was reported that it performed better than CSHOT or Spin Image in [28]. The reason seems to be that texture information was not adopted into BRAND in our implementation. For BRAND, there were so many equally distanced shapes because it used the hamming distance while the others use a floating-point L1 distance.

There is another notable point that precisions of most descriptors were generally low in the racing car dataset, especially when a support radius is small. The reason seems to be the scale of the object (racing car). In Figure 3, the racing car looks like it contains various shapes but the scale of the shapes are larger than the support radii. Since local shapes within the radii, 4 to 6 cm, were not distinctive enough in the dataset, the shape recognition rates generally fell down except for BRAND.

From the shape recognition results, we can conclude that the performance of descriptors does not depend on dimensionality, and our compact descriptor can work as effectively as high dimensional descriptors when querying shape from depth images.

4.5. Effect of Angular Difference Weight

The effect of the angular difference weight, η, in Equation (12) is addressed here. This parameter balances the point-to-plane distance and the angular difference. Since the shape distance was used to find ground truth correspondences for shape recognition, the parameter should be carefully selected but it is more desirable that the shape recognition rate is insensitive to the parameter. We evaluated Precision-1 with different values of the parameter. For simplicity, a single support radius, 5 cm, was used and the precisions over the nine datasets were averaged. The results were shown in Figure 7. While the angular difference weight varied from 0.0025 to 0.04, the precisions differed just 1 or 2 percentages. As the shape recognition rate is not sensitive to the angular difference weight, our shape distance metric with $\eta = 0.01$ can be considered to be generally reliable to find ground truth correspondences and not biased to any descriptor.

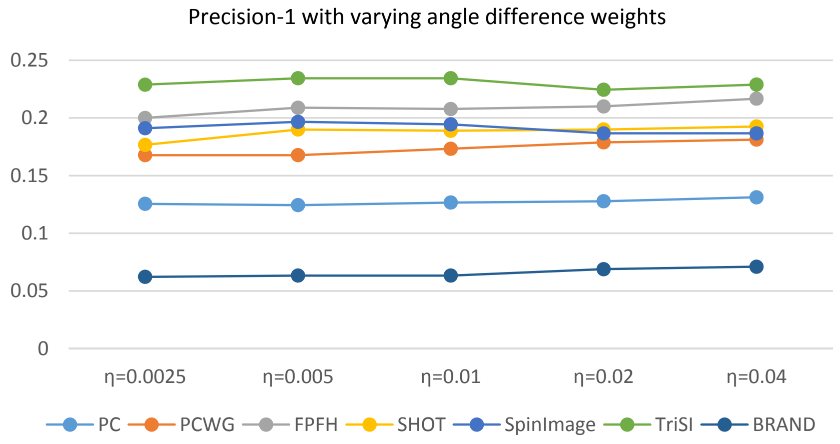


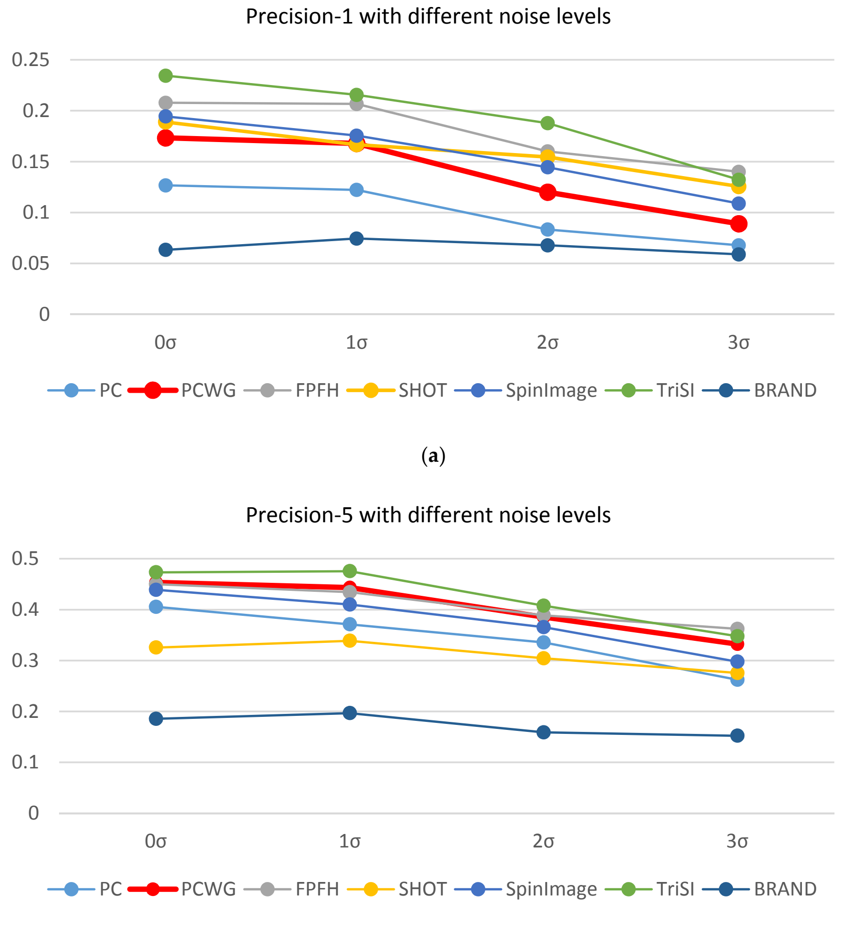
Figure 7. Comparison of Precision-1 of the seven descriptors where the precision was averaged over the nine datasets.

4.6. Robustness to Noise and Scale Variation

An ideal descriptor should be invariant to noise and scale changes. In order to evaluate the robustness of descriptors, shape recognition rates were recomputed with additive noise or scaled depth images. To simplify the test, we used a single support radius of 5 cm. Robustness to noise was evaluated by adding Gaussian noise to depth images. Wasenmüller et al. [45] showed the graph where the standard deviation of noise of Kinect v2 increased with a depth, which could be approximately modeled by the following linear equation:

$$\sigma(d) = 1.3 * d + 0.3 \tag{15}$$

where $\sigma(d)$ represents the standard deviation of noise in millimeters at a depth, d. To simulate amplifying noise, we added noise with the standard deviation of $\tau\sigma(d)$ to raw depth images. In our simulation, shape recognition rates were re-evaluated with different noise levels of $\tau = 0, 1, 2, 3$ and the results were shown in Figure 8 where precisions were averaged over the nine datasets. As expected, the both precisions tended to decrease with increasing noise levels regardless of descriptor types. The slopes of decrements did not differ much among the top-5 descriptors in the both precisions. Though curvatures and even its gradients of surfaces are known to be sensitive to noise, the graph of PCWG is blended with other descriptors in the figure. Thus the PCWG's robustness to noise is similar to the existing descriptors.



(**a**)

(**b**)

Figure 8. Precisions of seven descriptors with different noise levels. The horizontal axis represents the standard deviation of additive noise to raw depth images. 0 σ means that raw depth images were used. (**a**) Precision-1 and (**b**) Precision-5.

Another issue is scale invariance. Image scale of an object changes with a distance. Ideally, shape descriptors should not affected by distance when they describe shapes within the same physical radius at the same location. However, descriptors can be influenced by scale changes in reality because both image resolution and noise property vary with a distance. As a camera position changes, we cannot re-compute a descriptor at the exactly the same position with the previous position, where a descriptor was computed, but only at the closest position in the current point cloud. As this slight error as well as sensor random noise affects the normal vector at the point, descriptor may change.

To simulate robustness of descriptors to scale changes, we computed descriptors at different image scales. As commented in Section 4.2, we used scaled images of 320 × 240 resolution in our experiment where the raw images were at 640 × 480 resolution. The reference descriptor set in Section 4.4 was re-computed at both double-scale (640 × 480) and half-scale (160 × 120) images. These *scaled* descriptors were used as the query descriptor sets and the shape recognition rates were evaluated as summarized in Table 6. The precisions were averaged over the nine datasets. FPFH and SHOT showed the best results on average while PCWG was ranked low. Overall, descriptors tended to be more robust when a scale was reduced except for FPFH. Similar to the results in the previous section, BRAND showed the lowest performance. We counted binary 'one's in BRAND at different scales, and BRAND contained about 25 ones at mid and high resolution and 10 ones at low resolution on average. The number of ones was largely affected by image resolution, which explains the result. It seems that PCWG was less robust to scale changes because the curvatures are highly sensitive to variation in normal vectors. The other descriptors are also largely dependent on normal vectors, but the effect of a normal vector is weakened by quantization of property values and smoothing histograms in descriptors. On the other hand, curvatures are directly influenced by normal vectors. However, gradient terms in PCWG helped the robustness, compared to PC and PCWG showed almost comparable results in Precision-5.

Table 6. Shape recognition rates between different image scales.

Scale	Precision	PC	PCWG	FPFH	SHOT	SpinImage	TriSI	BRAND
1/2	Precision-1	0.538	0.622	0.668	0.917	0.946	0.860	0.025
1/2	Precision-5	0.888	0.871	0.835	0.950	0.985	0.939	0.090
2	Precision-1	0.246	0.318	0.775	0.569	0.341	0.426	0.020
2	Precision-5	0.602	0.645	0.947	0.721	0.664	0.711	0.094

4.7. Object Recognition

Object recognition is another important application of local shape descriptors. The six types of descriptors out of the seven were compared in object recognition except for BRAND, which showed meaninglessly low performance in the shape recognition. For object recognition, a typical bag-of-words (BoW) approach [46] was used. As our aim was not to achieve a higher recognition rate but to compare relative performances of descriptors, we used neither an SVM classifier nor the weighted distance [47] but used simple L1 distance for recognition. More advanced BoW techniques may result in higher recognition rates but if the performances of all the descriptors are improved, relative results will be the same. In addition, this simple classifier could not be optimized for any descriptor and is easy to implement and fast. To train code words, descriptors from the very first videos of all instances in the RGB-D object dataset were clustered for each type of descriptor. Objects were recognized in two levels, instance and category. In each level, recognition rates were evaluated with all the combinations of three codebook sizes (50, 100 and 200) and the support radii.

For object instance recognition, the video-level BoW descriptor was computed by averaging the BoW descriptors of the first five frames of the video. Two sets of video-level BoW descriptors were constructed. As multiple videos belong to each instance, one set is from the second videos of all instances and the other set is from the third videos. The performance of instance recognition was evaluated by cross validation between the two sets of video-level BoW descriptors.

For category recognition, the instance-level BoW descriptor was obtained by averaging the video-level descriptors belonging to the same instance. For each category, the instance-level descriptors were computed from five instances belonging to the category. Three of them were selected to model the reference category-level BoW descriptor by averaging the selected instance-level descriptors. The other two instance-level descriptors were matched with the closest category-level descriptor for category recognition. For cross validation, the initial selection of instances for the category-level descriptor was the three consecutive instances beginning from the first one, and the selection kept to be shifted to begin with the next one. Total ten tests were made for each category from two query instances for each of the five selections.

Figures 9 and 10 show the instance and category recognition results, respectively. Generally, PCWG ranked high in instance recognition but low in category recognition, and surprisingly, the simplest PC also showed the meaningful performance in instance recognition. On the contrary, SHOT showed the best performance in the category recognition but the worst in instance recognition. It seems that PCWG was better in matching specific shapes, while SHOT was better in the generalization of shapes. Spin Image showed generally low performance in category recognition. It is noteworthy that the both performances tend to increase with the radius in most of the descriptors. It is more apparent in the category recognition.

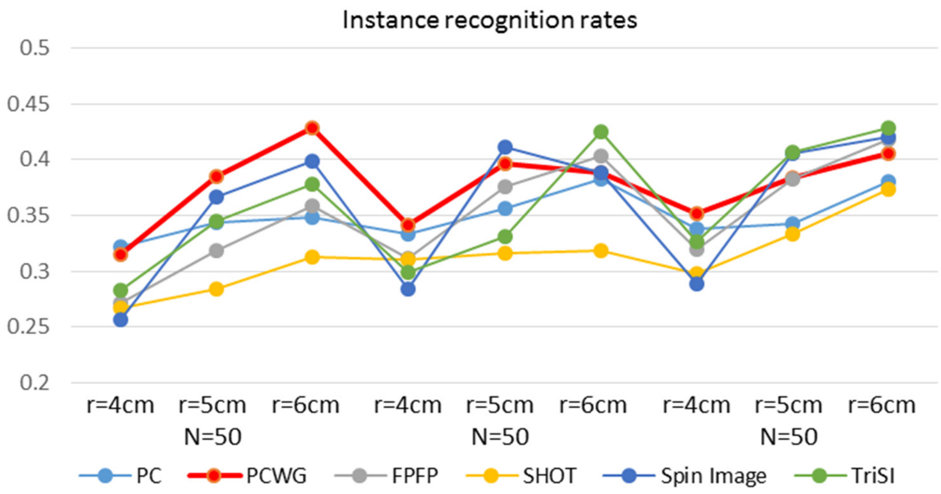


Figure 9. Instance recognition rates of six descriptors with different codebook sizes and support radii.

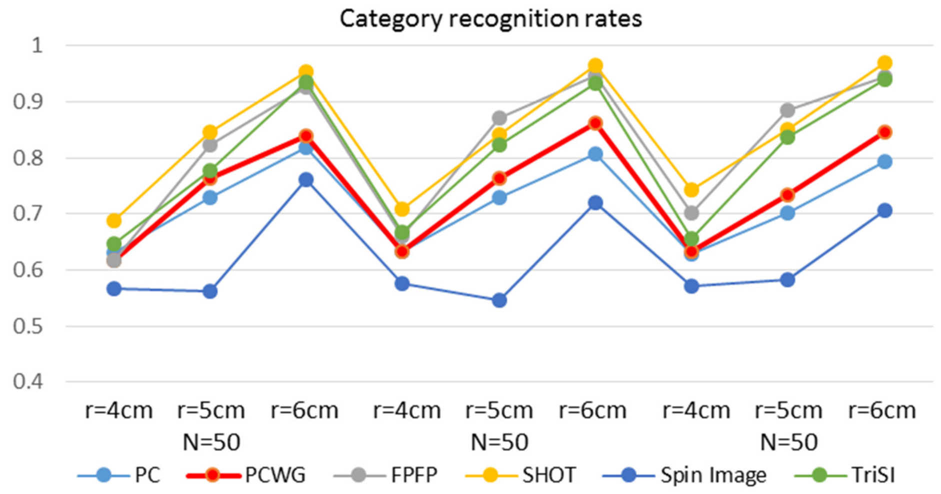


Figure 10. Category recognition rates of six descriptors with different codebook sizes and support radii.

Overall, the superiority of PCWG in instance recognition points out the large redundancy of the existing descriptors, while the inferiority in category recognition reveals an excessive sensitivity to small changes in shapes.

5. Conclusions

Since local shape descriptors have lower performances than local texture descriptors in general, their large dimensionalities motivated us to figure out the origin of high dimensionalities and an alternative compact descriptor with comparable performance. The answer to the question in the title is here: High dimensional descriptors have a potential to discriminate various shapes, especially when the shape is complicated and densely modeled. However, they usually waste memory dealing with depth images from Kinect-like popular sensors where the shapes are uncomplicated and the sensor resolutions are limited.

That is why we proposed a new descriptor, PCWG. The principal curvatures roughly describe a shape as a quadratic surface and their gradients add the details of the shape. Closed-form equations were derived to estimate the descriptor from the point cloud, and we implemented it based on GPU. We proved the inefficiency of the existing descriptors in Section 2 and showed the competitive performance of the PCWG in Section 4. Although the proposed descriptor is only four dimensional, it showed superior performance in shape and object instance recognition with a small support radius, a medium performance with larger support radii, and lower performance in object category recognition. The PCWG's high sensitivity to shapes is advantageous for low-level shape matching but disadvantageous for shape abstraction. This is because a small change in a part of a local shape affects the entire vector of the PCWG, while it affects only a part of the histogram-based descriptors.

In conclusion, our descriptor is useful for point association and object instance recognition in ordinary environments with limited resources. For the future works, we will develop a more advanced descriptor to express more complicated shapes with the least additional dimensions. For instance, texture information can be adopted to make up for the simplicity of PCWG. The extension of PCWG could lead to a new descriptor, which satisfies both the better performance in all recognition levels and the new approaches in Section 2.4.

Acknowledgments: This research was supported by the Industrial Strategic technology development, 10047635, Development of Hydraulic Robot Control Technology based on Accurate and Fast Force Control for Complex Tasks funded By the Ministry of Trade, Industry & Energy (MI, Korea).

Author Contributions: H. Choi, and E. Kim developed the algorithm, and carried out the experiment, and wrote the paper, all together.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buys, K.; Cagniart, C.; Baksheev, A.; De Laet, T.; De Schutter, J.; Pantofaru, C. An adaptable system for RGB-D based human body detection and pose estimation. *J. Vis. Commun. Image Represent.* **2014**, *25*, 39–52. [CrossRef]
2. Chen, X.; Koskela, M. Online RGB-D Gesture Recognition with Extreme Learning Machines. In Proceedings of the 15th ACM on International Conference on Multimodal Interaction, Sydney, Australia, 9–13 December 2013.
3. Morell-Gimenez, V.; Saval-Calvo, M.; Azorin-Lopez, J.; Garcia-Rodriguez, J.; Cazorla, M.; Orts-Escolano, S.; Fuster-Guillo, A. A Comparative Study of Registration Methods for RGB-D Video of Static Scenes. *Sensors* **2014**, *14*, 8547–8576. [CrossRef] [PubMed]
4. Saval-Calvo, M.; Orts-Escolano, S.; Azorin-Lopez, J.; Garcia-Rodriguez, J.; Fuster-Guillo, A.; Morell-Gimenez, V.; Cazorla, M. Non-rigid point set registration using color and data downsampling. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2015; pp. 1–8.

5. Endres, F.; Hess, J.; Engelhard, N.; Sturm, J.; Cremers, D.; Burgard, W. An evaluation of the RGB-D SLAM system. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St Paul, MN, USA, 14–18 May 2012; pp. 1691–1696.
6. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 345–360.
7. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663. [CrossRef]
8. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3-D Mapping With an RGB-D Camera. *IEEE Trans. Robot.* **2014**, *30*, 177–187. [CrossRef]
9. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded up Robust Features. InProceedings of the European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 404–417.
11. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
12. Calonder, M.; Lepetit, V.; Ozuysal, M.; Trzcinski, T.; Strecha, C.; Fua, P. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1281–1298. [CrossRef] [PubMed]
13. Alahi, A.; Ortiz, R.; Vandergheynst, P. FREAK: Fast Retina Keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 510–517.
14. Johnson, A.E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 433–449. [CrossRef]
15. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
16. Tombari, F.; Salti, S.; Di Stefano, L. Unique Signatures of Histograms for Local Surface Description. In Proceedings of the 11th European Conference on Computer Vision (ECCV), Hersonissos, Greece, 5–11 September 2010; pp. 356–369.
17. Guo, Y.; Sohel, F.; Bennamoun, M.; Wan, J.; Lu, M. A novel local surface feature for 3D object recognition under clutter and occlusion. *Inform. Sci.* **2015**, *293*, 196–213. [CrossRef]
18. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287.
19. Dinh, H.Q.; Kropac, S. Multi-Resolution Spin-Images. In Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; Volume 1, pp. 863–870.
20. Pasqualotto, G.; Zanuttigh, P.; Cortelazzo, G.M. Combining color and shape descriptors for 3D model retrieval. *Signal Process. Image Commun.* **2013**, *28*, 608–623. [CrossRef]
21. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning point cloud views using persistent feature histograms. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391.
22. Savelonas, M.A.; Pratikakis, I.; Sfikas, K. Fisher encoding of differential fast point feature histograms for partial 3D object retrieval. *Pattern Recognit.* **2016**, *55*, 114–124. [CrossRef]
23. Tombari, F.; Salti, S.; Stefano, L.D. A combined texture-shape descriptor for enhanced 3D feature matching. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 809–812.
24. Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [CrossRef]
25. Zhong, Y. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 29 September–2 October 2009; pp. 689–696.

26. Guo, Y.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. *Int. J. Comput. Vis.* **2013**, *105*, 63–86. [CrossRef]
27. Steder, B.; Rusu, R.B.; Konolige, K.; Burgard, W. Point feature extraction on 3D range scans taking into account object boundaries. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 2601–2608.
28. Nascimento, E.R.; Oliveira, G.L.; Campos, M.F.M.; Vieira, A.W.; Schwartz, W.R. BRAND: A robust appearance and depth descriptor for RGB-D images. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 1720–1726.
29. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *PP*, 1. [CrossRef] [PubMed]
31. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv* **2013**, arXiv:1310.1531.
32. Balntas, V.; Johns, E.; Tang, L.; Mikolajczyk, K. PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors. *arXiv* **2016**, arXiv:1601.05030.
33. Paulin, M.; Douze, M.; Harchaoui, Z.; Mairal, J.; Perronin, F.; Schmid, C. Local convolutional features with unsupervised training for image retrieval. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 91–99.
34. Rusu, R.B.; Holzbach, A.; Beetz, M.; Bradski, G. Detecting and segmenting objects for mobile manipulation. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 29 September–2 October 2009; pp. 47–54.
35. Aldoma, A.; Tombari, F.; Rusu, R.B.; Vincze, M. OUR-CVFH—Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation. In *Pattern Recognition. DAGM/OAGM 2012*; Pinz, A., Pock, T., Bischof, H., Leberl, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 113–122.
36. Marton, Z.C.; Pangercic, D.; Rusu, R.B.; Holzbach, A.; Beetz, M. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In Proceedings of the 2010 10th IEEE-RAS International Conference on Humanoid Robots, Cancun, Mexico, 15–17 November 2010; pp. 365–370.
37. Wasenmüller, O.; Meyer, M.; Stricker, D. CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–9 March 2016; pp. 1–7.
38. Lachat, E.; Macher, H.; Landes, T.; Grussenmeyer, P. Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling. *Remote Sens.* **2015**, *7*, 13070–13097. [CrossRef]
39. Xie, Z.; Xu, S.; Li, X. A high-accuracy method for fine registration of overlapping point clouds. *Image Vis. Comput.* **2010**, *28*, 563–570. [CrossRef]
40. Cheng, Z.; Zhang, X. Estimating differential quantities from point cloud based on a linear fitting of normal vectors. *Sci. China Ser. F Inform. Sci.* **2009**, *52*, 431–444. [CrossRef]
41. Spek, A.; Li, W.; Drummond, T. A Fast Method for Computing Principal Curvatures from Range Images. In Proceedings of the Australasian Conference on Robotics and Automation Robert Mahony, Canberra, Australia, 2–4 December 2015.
42. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 1817–1824.
43. Munshi, A. The OpenCL specification. In Proceedings of the 2009 IEEE Hot Chips 21 Symposium (HCS), Stanford, CA, USA, 23–25 August 2009.
44. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
45. Wasenmüller, O.; Stricker, D. Comparison of Kinect v1 and v2 Depth Images in Terms of Accuracy and Precision. In Proceedings of the Asian Conference on Computer Vision Workshop (ACCV workshop), Taipei, Taiwan, 20–24 November 2016.

46. Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In Proceedings of the Workshop on Statistical Learning in Computer Vision, ECCV, Prague, Czech Republic, 10–16 May 2004.
47. Nister, D.; Stewenius, H. Scalable recognition with a vocabulary tree. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2161–2168.