Pattern Recognition Letters 000 (2016) 1–9

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

GOOD: A global orthographic object descriptor for 3D object recognition and manipulation[☆]

S. Hamidreza Kasaei[a,*], Ana Maria Tomé[a,b], Luís Seabra Lopes[a,b], Miguel Oliveira[c]

[a] IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro, Aveiro, 3810-193, Portugal
[b] Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Portugal
[c] Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência R. Dr. Roberto Frias, 465, Porto 4200, Portugal

ARTICLE INFO

Article history:
Available online xxx

Keywords:
3D object recognition
Object Perception
Orthographic projection

ABSTRACT

Object representation is one of the most challenging tasks in robotics because it must provide reliable information in real-time to enable the robot to physically interact with the objects in its environment. To ensure robustness, a global object descriptor must be computed based on a unique and repeatable object reference frame. Moreover, the descriptor should contain enough information enabling to recognize the same or similar objects seen from different perspectives. This paper presents a new object descriptor named Global Orthographic Object Descriptor (GOOD) designed to be robust, descriptive and efficient to compute and use. We propose a novel sign disambiguation method, for computing a unique reference frame from the eigenvectors obtained through Principal Component Analysis of the point cloud of the target object view captured by a 3D sensor. Three principal orthographic projections and their distribution matrices are computed by exploiting the object reference frame. The descriptor is finally obtained by concatenating the distribution matrices in a sequence determined by entropy and variance features of the projections. Experimental results show that the overall classification performance obtained with GOOD is comparable to the best performances obtained with the state-of-the-art descriptors. Concerning memory and computation time, GOOD clearly outperforms the other descriptors. Therefore, GOOD is especially suited for real-time applications. The estimated object's pose is precise enough for real-time object manipulation tasks.

1. Introduction

Following the advent of inexpensive depth sensing devices such as Microsoft Kinect or the ASUS Xtion, which record RGB and depth information, the use of 3D data is becoming increasingly popular. One of the primary goals in service robotics is to develop reliable capabilities in the area of perception that will allow robots to robustly recognize objects and interact with the environment by manipulating those objects. For this purpose, a robot must reliably recognize the object. Furthermore, in order to interact with human users, this process of object recognition cannot take more than a fraction of a second.

Although many object recognition methods for both 2D and 3D data have been proposed [2], recognizing 3D objects in the presence of noise and variable point cloud resolution is still a challenging task. However, 3D data contains more information about the spatial positioning of objects, which in turn eases the process of object segmentation. Moreover, depth data is more robust than RGB data to the effects of illumination and shadows [25]. Therefore, 3D data can be employed to describe the surface of the objects based on geometric properties[1].

A 3D object recognition system is composed of several software modules such as *Object Detection*, *Object Representation*, *Object Recognition* and *Perceptual Memory*. *Object Detection* is responsible for detecting all objects in a scene. Object representation is concerned with the calculation of a set of features for the detected object, which are send to the *Object Recognition*. Objects are recognized by comparing their description against the descriptions of known objects (stored in the *Perceptual Memory*). *Object Representation* plays a prominent role because the output of this module is used for learning as well as recognition. Existing 3D object representation approaches are based on either global or local descriptors. Global descriptors encode the entire 3D object, while local

[☆] This paper has been recommended for acceptance by Gabriella Sanniti di Baja.

[*] Corresponding author. Tel.: +351 234 370 500; fax: +351 234 370 545.
E-mail address: seyed.hamidreza@ua.pt (S.H. Kasaei).

[1] 2D data can also be used to distinguish objects that have same geometric properties with different texture (e.x. a Coke can from a Diet Coke can).

http://dx.doi.org/10.1016/j.patrec.2016.07.006
0167-8655/© 2016 Elsevier B.V. All rights reserved.

descriptors represent a small area of an object around a specific keypoint. Generally, global descriptors are increasingly used in the context of 3D object recognition, object manipulation, as well as geometric categorization [1]. These must be efficient in terms of computation time as well as memory, to facilitate real-time performance. For example, Ensemble of Shape Functions (ESF) [32], Global Fast Point Feature Histogram (GFPFH) [28], Viewpoint Feature Histogram (VFH) [27] and Global Radius-based Surface Descriptor (GRSD) [16], are global descriptors. Local descriptors tend to handle occlusion and clutter better when compared to global descriptor. However, comparing 3D object views based on their local features tends to be computationally more expensive [1]. Examples in this category include Spin Images (SI) [12], Signature of Histograms of Orientations (SHOT) [31], Fast Point Feature Histogram (FPFH) [26] and Hierarchical Kernel Descriptors [3]. Invariance to the pose of an object is a critical property of any 3D shape descriptor. A number of 3D shape descriptors achieve pose invariance using either a reference axis only (e.x. Spin-Images [12]) or a complete object reference frame (e.x. Intrinsic Shape Signatures [34]).

In this paper, a new global 3D shape descriptor named GOOD (i.e. Global Orthographic Object Descriptor) is presented. GOOD provides an appropriate trade-off between descriptiveness, computation time and memory usage. The descriptor is designed to be scale and pose invariant, informative and stable, with the objective of supporting accurate 3D object recognition. A novel sign disambiguation method is proposed to compute a unique and repeatable reference frame from the eigenvectors obtained through Principal Component Analysis of the point cloud of the object. Using this reference frame, three three principal projections, namely *XoZ*, *XoY* and *YoZ*, are created based on orthographical projection. The space of each projection is partitioned into bins and the number of points falling into each bin is counted. From this, three distribution matrices are obtained for the projected views. Two statistic features, namely *entropy* and *variance* are then calculated for each distribution matrix. The distribution matrices are consequently concatenated together using the entropy and variance features to form a single description for the given object view.

In this paper, we assume that an object has already been segmented from the point cloud of the scene, and we will focus on detailing the 3D object descriptor. This descriptor works directly on 3D point clouds and requires neither triangulation of the object's points nor surface meshing. For additional details on the object detection and object recognition methodologies, we refer the reader to our previous works on interactive open-ended learning for 3D object recognitions [13,19,20].

The contributions presented in this paper are the following: (*i*) design a new sign disambiguation method to compute a unique and unambiguous complete local reference frame, from the eigenvectors obtained through Principal Component Analysis of the segmented point cloud of the object and (*ii*) a novel global object descriptor computed using that local reference frame, that provides a good trade-off between descriptiveness, computation time and memory usage.

The remainder of this paper is organized as follows. In Section 2, we discuss related works. The methodology for computing the local reference frame is presented in Section 3. Section 4 describes the proposed global object descriptor. Evaluation of the proposed shape descriptor is presented in Section 5. Finally, in Section 6, conclusions are presented and future research is discussed.

2. Related work

Three-dimensional shape description has been under investigation for a long time in various research fields, such as pattern recognition, computer graphics and robotics. Although an exhaustive survey of shape descriptor is beyond the scope of this paper, we will review a few recent efforts.

As previously mentioned, some descriptors use LRF to compute a pose invariant description. Therefore, this property can be used to categorize 3D shape descriptors into two categories including (*i*) shape descriptors without LRF; (*ii*) shape descriptors with LRF. Most of the shape descriptors of the first category use certain statistic features or geometric properties of the points on the surface like depth value, curvature and surface normal to generate a description. For instance, Shape Distributions descriptor [21] represents an object as a shape distribution sampled from a shape function measuring global geometric properties of the object. Extended Gaussian Images (EGI) descriptor [11] is based on the distribution of surface normals on the Gaussian sphere. Since descriptiveness of the EGI depends on the shape of the object and it is not suitable for non-convex object. Chen and Bhanu [5] proposed a local surface patch (LSP) descriptor that encodes the shape of objects by accumulating points in particular bins along the two dimensions that are the shape index value and the cosine of the angle between the surface normals. Wohlkinger and Vincze [32] introduced a global shape descriptor called Ensemble of Shape Function (ESF) that does not require the use of normals to describe the object and the characteristic properties of an object is represented using an ensemble of ten 64-bin histograms of angle, point distance, and area shape functions. Point Feature Histogram (PFH) [29] can be used as local or global shape descriptor. The PFH represents the relative orientation of normals, as well as distances, between point pairs. For each point *p*, k-neighborhood points are selected based on a sphere centered at *p* with radius *r*. Afterwards, a surface normal for each point is estimated. Subsequently, four features are calculated for every pair of points using their surface normals, positions and angular variations. In a later work [26], in order to improve the robustness of PFH in case of point densities variations, the distance between point pairs is excluded from the histogram of PFH. The computation complexity of a PFH is $O(n^2)$, where n is the number points in the point cloud. Fast Point Feature Histogram (FPFH) [26] is an extension version of PFH. The FPFH estimates the sets of values only between every point and its k nearest neighbors. This is different from PFH, where all pairs of points in the support region are considered. Therefore, the computational complexity is reduced to $O(k.n)$. The FPFH is a scale and pose invariant descriptor which is not suitable for grasping. Viewpoint Feature Histogram (VFH) [27] is another extension of PFH descriptor. The VFH descriptor computes the same angular features as the PFH. Additionally, it computes another statistics between the central viewpoint direction and the normals estimated at each point. The VFH shape descriptor produces a single histogram that encodes the geometry of the whole object and its viewpoint. Because of the global nature of VFH, the computational complexity of VFH is $O(n)$. The descriptiveness of the above shape descriptors are limited because the 3D spatial information either is not taken into account or it is discarded during the description process. Unlike the above approaches, some researchers have recently adopted deep learning algorithms for 3D object representation, learning and recognition [15,30,33]. These works use a collection of 2D images rendered from different view points to learn a shape representation that aggregates information from input views and provides a compact shape descriptor. As it was pointed out in [33], training a deep artificial neural network for 3D object representation requires a large collection of 3D objects to provide accurate representations and typically involves long training times.

In contrast, the shape descriptors in the second category encode the spatial information of the objects' points using a Local Reference Frame (LRF). Some descriptors provide a description using only a Reference Axis. For example, Spin-Images [12] uses surface normal of a vertex as a reference axis and proposed a spin image representation by projecting the surface points to the tangent plane of the vertex. Then, each projected point is represented by a pair (α, β), where α is the distance to the surface normal, i.e., the radius, and β is the perpendicular distance from the point to the tangent plane. Consequently, a histogram is formed by counting the occurrences of different discretized distance pairs. Spin images descriptor has been successfully used in many applications, but one limitation of this descriptor is that it is not scale invariant. Dinh and Kropac [8] proposed multi-resolution pyramids of spin images in order to improve the discrimination of the original spin image and speed up the matching process. Some variants of the spin image shape descriptor also presented such as Tri-Spin-Image descriptor (TriSI) [10] and color spin image [23]. Similar to the SI, 3D Shape Context (3DSC) [9] uses the surface normal of an basis point as its LRF. The 3DSC descriptor is calculated by counting the weighted number of points falling into each bin of an sphere grid centered on the basis point and its north pole oriented with the surface normal. The sphere grid is constructed based on dividing the support area into bins by logarithmically spaced boundaries along the radial dimension and equally spaced boundaries in the azimuth and elevation dimensions.

Whenever only an axis is used as a reference frame, there is an uncertainty in the rotation around the axis that should be handled for generating a robust and repeatable description. In order to eliminate this issue, several descriptors (e.g. 3D Shape Context) proposed to compute multiple descriptions for different possible rotations of the object. Since this kind of solutions are caused increasing the computational cost in terms of both execution time as well as memory usage, they are not optimum and real solutions. Furthermore, the recognition process becomes not only significantly slow, but also more laborious.

Differently, Zhong [34] proposed a shape descriptor namely Intrinsic Shape Signatures (ISS) using defining a LRF based on the eigenvectors of the scatter matrix of the point cloud of the object and describing the point distribution in the spherical angular space. Similar to Zhong work, Mian et al. [17] introduced LRF computed with eigenvectors of the covariance matrix of the object's points. However, in both cases the eigenvectors define the principal directions of the data, their sign is not defined unambiguously. Accordingly, different descriptors can be generated for the object. As highlighted before, they are neither computationally efficient nor repeatable. [22] proposed a multi-view 3D object recognition approach. In this approach, each object is projected into 46 projection planes distributed on a sphere, whereas we just compute three principal orthographic projections. Their object representation is clearly not efficient for real time application like robotics.

In order to achieve true rotation invariant descriptor, Tombari et al. [31] proposed a 3D shape descriptor namely Signature of Histograms of OrienTations (SHOT). To generate the description for the object, they first applied a sign disambiguation technique to the eigenvectors of the scatter matrix of the object and constructed a unique and unambiguous LRF. The object's points are then aligned with the LRF. Consequently, similar to 3D Shape Context, a spherical coordinate based approach is used to generate a SHOT description for the given object. 3D object descriptors that use spherical coordinate system suffer from the singularity issue at the poles, because bins at the poles are significantly smaller than bins around the equator.

Our shape descriptor differ from all of the listed descriptors above as it is simultaneously unique, unambiguous, and robust to noise and varying low-level point cloud density. Besides, our approach can be used not only for object recognition but also for object manipulation.

3. Local reference frame

A *Local Reference Frame* (LRF), invariant to translations and rotations and robust to noise is important for object recognition as well as object manipulation. Since the repeatability of a LRF directly affects the descriptiveness of the object representation [17], the LRF should be as repeatable and robust as possible to improve the performance of object recognition. In this section, we propose a method to compute a LRF. For this purpose, the three principal axes of a given object are firstly determined based on Principal Component Analysis (PCA). Given a point cloud of an object that contains m points, $\mathbf{O} = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$, the geometric center of the object is defined as:

$$\mathbf{c} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{p}_i, \tag{1}$$

where $\mathbf{p}_i$ is a three dimensional point in the object's point cloud. The normalized covariance matrix, $\mathbf{C}$, of the object is constructed:

$$\mathbf{C} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{p}_i - \mathbf{c})(\mathbf{p}_i - \mathbf{c})^T. \tag{2}$$

Then, eigenvalue decomposition is performed on $\mathbf{C}$:

$$\mathbf{CV} = \mathbf{EV}, \tag{3}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ contains the three eigenvectors, $\mathbf{E} = diag(\lambda_1, \lambda_2, \lambda_3)$ is a diagonal matrix of the corresponding eigenvalues and $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Since the covariance matrix is symmetric positive, its eigenvalues are positive and the eigenvectors are orthogonal.

Eigenvectors define directions which are not unique, i.e. not repeatable across different PCA trials. This is known as the sign ambiguity problem, for which there is no mathematical solution [4]. Since there are two possible directions for each eigenvector, a total of eight reference frames can be created from the same set of eigenvectors. A mechanism is needed to transform this reference frame into a unique object reference frame, which will be always the same across multiple trials.

We start with a provisional reference frame, in which the first two axes, X and Y, are defined by the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. However, regarding the third axis, Z, instead of defining it based on $\mathbf{v}_3$, we define it based on the cross product $\mathbf{v}_1 \times \mathbf{v}_2$. This way, because the result of the cross product follows the right-hand rule, the number of alternatives is reduced to four. It is now enough to disambiguate the directions of the X and Y axes. So either the directions of X and Y are both changed or both remain unchanged.

To complete the disambiguation, the object's point cloud, $\mathbf{O}$, is transformed to be placed in the provisional reference frame. Then, the number of points that have positive x, S_x^+, and the number of points that have negative x, S_x^-, are counted as follows:

$$S_x^+ = \{i : x_{p_i} > t\}, \quad S_x^- = \{i : x_{p_i} < -t\}, \tag{4}$$

where t is a threshold (e.g. $t = 0.015m$) that is used to deal with the special case when a point is close to the YoZ plane, and therefore can change from negative to positive X in different trials. Afterwards, the variable S_x is defined as:

$$S_x = \begin{cases} +1, & |S_x^+| \geq |S_x^-| \\ -1, & otherwise \end{cases}, \tag{5}$$

where |.| denotes the number of points of the argument. A similar indication, S_y, is computed for the Y axis. Finally, the sign of the axes is determined as:

$$s = S_x \,.\, S_y, \tag{6}$$

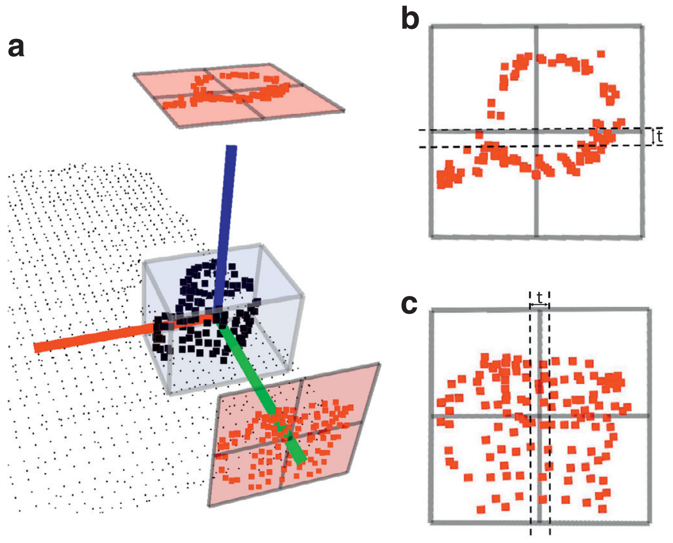


Fig. 1. Visualization of sign disambiguation procedure: *(a)* orthographic projection of the object on the *XoZ* and *XoY* planes; *(b) XoY* plane is used to determine the sign of *Y* axis; *(c) XoZ* plane is used to determine the sign of *X* axis. The red, green and blue lines represent the unambiguous *X*, *Y*, *Z* axes respectively.

where s can be either -1 or $+1$. In case of $s = -1$, the directions of X and Y must be changed, otherwise not. Therefore, the final LRF (X, Y, Z) will be defined by $(s\mathbf{v}_1, s\mathbf{v}_2, \mathbf{v}_1 \times \mathbf{v}_2)$. An illustrative example of the sign disambiguation procedure is provided in Fig. 1.

4. Object descriptor

This section describes the computation of the proposed object descriptor, GOOD, in the obtained LRF centered in the geometric center of the object. The descriptor consists of a concatenation of the orthographic projections of the object on the three orthogonal planes, *XoY*, *YoZ* and *XoZ*. Each projection is described by a distribution matrix. To ensure correct comparison between different object shapes, the number of bins in the distribution matrices must be the same and the bins should be of equal size. Therefore, each distribution matrix must be computed from a square area in the projection plane centered on the object's center, and this square area must have the same dimensions for the three projections. The side length of these square areas, l, is determined by the largest edge length of a tight-fitting axis-aligned bounding box (AABB) of the object. The dimensions of the AABB are obtained by computing the minimum and maximum coordinate values along each axis. With this setup, the number of bins, n, is the only parameter that must be specified to compute GOOD. For each projection, the $l \times l$ projection area is divided into $n \times n$ square bins. Finally, a distribution matrix $\mathbf{M}_{n\times n}$ is obtained by counting the number of points falling into each bin.

For each projected point $\rho = (\alpha, \beta) \in R^2$, where α is the perpendicular distance to the horizontal axis and β is the perpendicular distance to the vertical axis, a row, $r(\rho) \in \{0, \dots, n-1\}$, and a column, $c(\rho) \in \{0, \dots, n-1\}$, are associated as follows:

$$r(\rho) = \left\lfloor \frac{\alpha + \frac{l}{2}}{\frac{l+\epsilon}{n}} \right\rfloor = \left\lfloor n\frac{\alpha + \frac{l}{2}}{l+\epsilon} \right\rfloor, \tag{7}$$

$$c(\rho) = \left\lfloor \frac{\beta + \frac{l}{2}}{\frac{l+\epsilon}{n}} \right\rfloor = \left\lfloor n\frac{\beta + \frac{l}{2}}{l+\epsilon} \right\rfloor, \tag{8}$$

where ϵ is a very small value used to deal with the special cases when a point is projected onto the upper bound of the projection area, and $\lfloor x \rfloor$ returns the largest integer not greater than x. Note that the projected view is shifted to right and top by $\frac{l}{2}$ (i.e. $\alpha + \frac{l}{2}$ and $\beta + \frac{l}{2}$). Furthermore, to achieve invariance to point cloud density, $\mathbf{M}$ is normalized such that the sum of all bins is equal to one (see Fig. 2). The matrix $\mathbf{M}$ is called distribution matrix, because it represents the 2D spatial distribution of the object's points. According to standard practice, this matrix is converted to a vector $\mathbf{m}_{1\times n^2} = [M(1,1), M(1,2), \dots, M(n,n)]$. The three projection vectors will be concatenated producing a vector of dimension $3 \times n^2$ which is the final object descriptor, GOOD. Statistical features are used to decide the order in which the projection vectors will be concatenated.

For the first projection in the descriptor, the one with largest area is preferred. The number of points is not a good indicator of area because all points of the object are represented in the

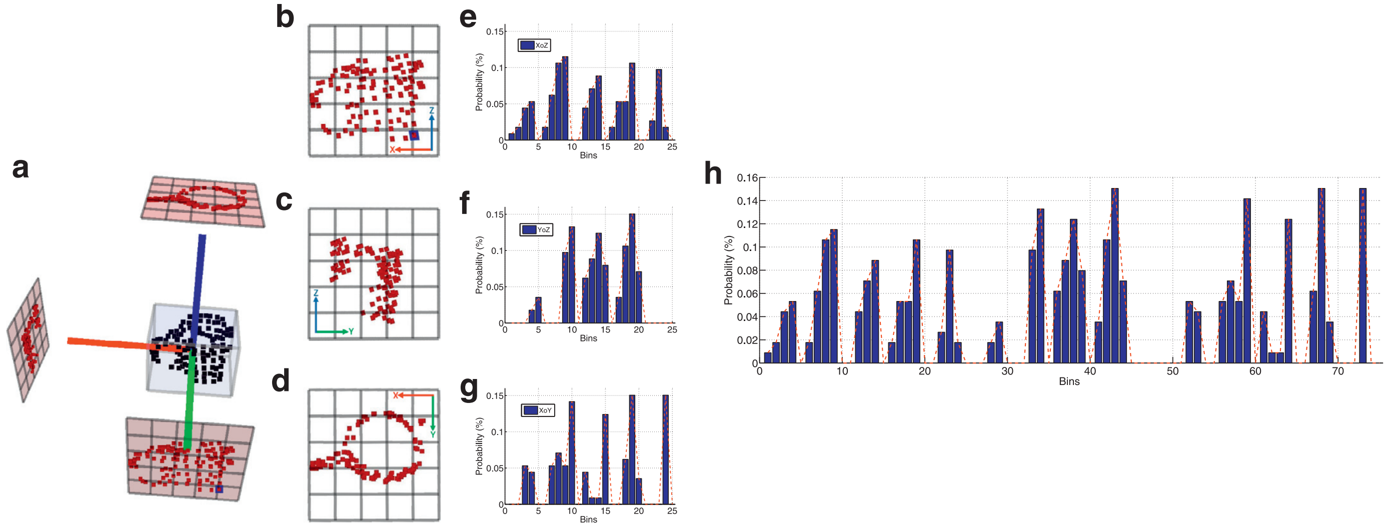


Fig. 2. An illustrative example of the producing a GOOD shape description for a mug object (i.e. $d = 5$): *(a)* The mug object and its bounding box, reference frame and three projected views; the object's points are then projected onto three planes; therefore, *XoZ (b)*, *YoZ (c)* and *XoY* projections *(d)* are created. Each plane is partitioned into bins and the number of point falling into each bin is counted. Accordingly, three distribution matrices are obtained for the projections; afterwards, each distribution matrix is converted to a distribution vector, (i.e. *(e)*, *(f)* and *(g)*) and two statistic features including *entropy* and *variance* are then calculated for each distribution vector; *(h)* the distribution vectors are consequently concatenated together using the statistics features, to form a single description for the given object. The ordering of the three distribution vectors is first by decreasing values of entropy. Afterwards the second and third vectors are sorted again by increasing values of variance.

Please cite this article as: S.H. Kasaei et al., GOOD: A global orthographic object descriptor for 3D object recognition and manipulation, Pattern Recognition Letters (2016), http://dx.doi.org/10.1016/j.patrec.2016.07.006

three projections. The number of occupied bins (the ones with a mass greater than 0) could be used as a measure of area. However, this measure tends to be brittle when the boundary of the object is close to boundaries between bins. Therefore, in this work the entropy of the projection is used. Entropy, a measure from Information Theory [6], nicely takes into account both the number of occupied bins and their density. In this work, the entropy of a projection is computed as follows:

$$H(\mathbf{m}) = -\sum_{i=1}^{n} \mathbf{m}_i \log_2 \mathbf{m}_i, \tag{9}$$

where $\mathbf{m}_i$ is the mass in bin i. The logarithm is taken in base 2 and $0\log_2 0 = 0$. The projection with highest entropy is the one that will appear in the first n^2 positions of the descriptor.

The next step is to select, from the remaining two projections, which one should appear in the second part of the descriptor (positions n^2 to $2n^2 - 1$). It is common that these two projections have similar areas, and therefore similar entropies, leading to instability of the decision if it is made based on entropy. Therefore, instead of entropy, we use variance to make this decision. Since the projection matrices are probability mass functions (pmf), the variance is defined as follows:

$$\sigma^2(\mathbf{m}) = \sum_{i=1}^{n} (i - \mu_{\mathbf{m}})^2 \mathbf{m}_i, \tag{10}$$

where $\mu_{\mathbf{m}}$ is the expected value (i.e. a weighted average of the possible values of i, corresponding to the geometric center of the projection), which is computed as follows:

$$\mu_{\mathbf{m}} = \sum_{i=1}^{n^2} i\mathbf{m}_i, \tag{11}$$

unlike the simple mean, which gives each projection equal weight, the mean of a projection weights each bin, i, according to its probability distribution, $\mathbf{m}_i$. The variance measure, $\sigma^2(\mathbf{m})$, is used to measure the spread or variability of the spatial distribution of the object's points in the projection vector. A small variance indicates that the projected points tend to be very close to each other and to the mean of the vector, i.e. the shape of distribution is small and compact. A high variance indicates that the data points in the projection vector are very spread out from the mean.

An illustrative example of the proposed shape descriptor is depicted in Fig. 2. In this example, after determining the local reference frame, a mug is projected onto the three orthogonal planes. Based on the entropy criterion, the *XoZ* projection (Fig. 2b) is selected to appear in the first part of the descriptor. Based on the variance criterion, the *YoZ* projection (Fig. 2c) is selected to appear in the second part of the descriptor. The remaining projection, *XoY* (Fig. 2d), appears in the last part of the descriptor.

In order to grasp an object, it is necessary to know true dimensions of different parts of the object. Such information is not adequately represented in most shape descriptors (e.g. Viewpoint Feature Histogram [27]). Because GOOD is composed of three orthogonal projections, it is especially rich in terms of information suited for manipulation tasks.

In Fig. 3, again we consider the projections of a mug. Here, we adopt a multi-view orthographic projection layout in which there is a central or front view, a top view and a side view. The central view is the one selected based on the entropy criterion and appearing in the first part of the descriptor. The top view contains the projection in the orthogonal plane formed by the horizontal axis of the central projection and the third axis. The side view contains the projection in the orthogonal plane formed by the vertical axis of the central projection and the third axis. The figure shows that projections can be further processed for object manipulation purposes. In the top view, the gray symbols *C*, *W*, *D* and *T* represent how the projection can be further processed and some features for manipulation task are extracted, namely inner radius (*C*), thickness (*T*), handle length (*W*) and handle thickness (*D*).

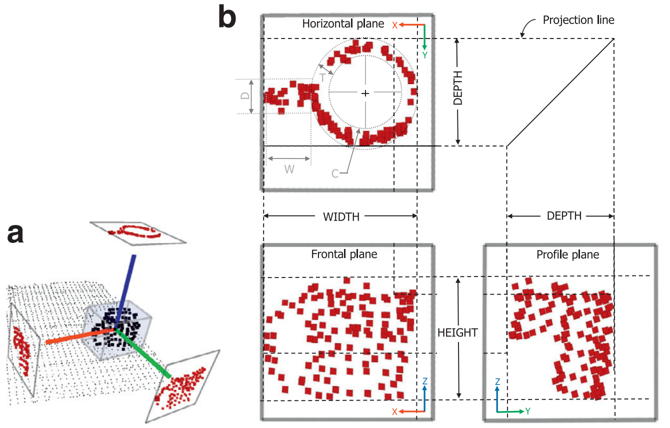


Fig. 3. Example of how the projections used to build GOOD can also be used for extracting features relevant for object manipulation (see text): (a) Local reference frame and projections; (b) Projections in multi-view layout.

5. Experimental results

Several experiments were carried out to evaluate the performance of the proposed object descriptor concerning ***descriptiveness***, ***scalability***, ***robustness*** and ***efficiency*** characteristics. The proposed descriptor has a parameter namely *number of bins* (i.e. *d*) that must be well selected to provide a good balance between recognition accuracy, memory usage and computation time. For this purpose, 10 experiments were performed for different values of the descriptor's parameter. The performance of the shape descriptor and its scalability were examined on the Washington RGB-D Object Dataset [14]. Afterwards, various tests were executed to measure the robustness of the proposed shape descriptor concerning different levels of noise and varying mesh resolutions on the Restaurant Object Dataset [13]. Next, two efficiency evaluations relating to computational efficiency and memory usage were performed. Furthermore, a real demonstration was performed to show all the characteristics of the proposed descriptor.

The largest publicly available dataset, namely Washington RGB-D Object Dataset [14], consisting of 250,000 views of 300 common household objects. The objects are categorized into 51 categories arranged using WordNet [18] hypernym-hyponym relationships (similar to ImageNet [7]). The Restaurant Object Dataset contains 241 views of one instance of each category (*Bottle*, *Bowl*, *Flask*, *Fork*, *Knife*, *Mug*, *Plate*, *Spoon*, *Teapot*, and *Vase*) [13].

In all experiments, an instance-based learning approach is used, i.e. object categories are represented by sets of known instances. The instance-based approach is a baseline method for evaluating representations. However, more advanced approaches like SVM and object Bayesian approaches can be easily adapted. Similarly, a simple baseline recognition mechanism in the form of a Euclidean nearest neighbor classifier is used. Moreover, the proposed descriptor was compared with four state-of-the-art object descriptors that are available in the Point-Cloud Library[2] (PCL version 1.7 and 1.8) including VFH [27], ESF [32], GFPFH [28] and GRSD [16].

The selected descriptors were evaluated based on a 10-fold cross validation algorithm in terms of Accuracy [24]. In each iteration, a single fold is used for testing, and the remaining data

[2] http://pointclouds.org/.

Please cite this article as: S.H. Kasaei et al., GOOD: A global orthographic object descriptor for 3D object recognition and manipulation, Pattern Recognition Letters (2016), http://dx.doi.org/10.1016/j.patrec.2016.07.006

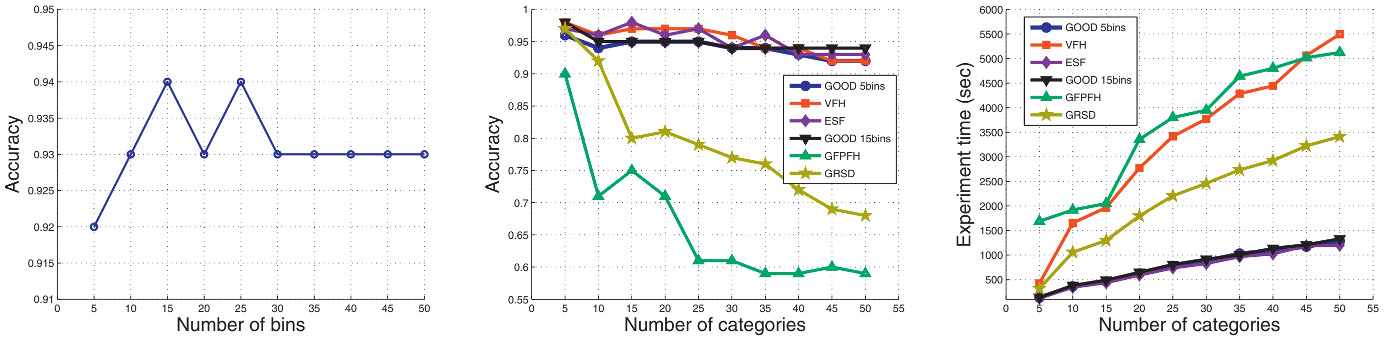


Fig. 4. Object recognition performance in descriptiveness and scalability experiments; (*left*) effect of number of bins on performance; (*center*) scalability of the selected descriptors with respect to varying numbers of categories in the dataset as a function of accuracy vs. Number of categories; (*right*) scalability experiment time vs. Number of categories.

Table 1
Summary of descriptiveness experiments.

Number of bins	Descriptor size	Memory (Kb)	Accuracy
5	75	0.3	0.92
10	300	1.2	0.93
15	675	2.7	0.94
20	1200	4.8	0.93
25	1875	7.5	0.94
30	2700	10.8	0.93
35	3675	14.7	0.93
40	4800	19.2	0.93
45	6075	24.3	0.93
50	7500	30.0	0.93

Table 2
Summary of scalability experiments.

Number of categories	Accuracy					
	GOOD(5bins)	GOOD(15bins)	VFH	ESF	GFPFH	GRSD
5	0.96	**0.98**	**0.98**	0.97	0.90	0.97
10	0.94	0.95	**0.96**	**0.96**	0.71	0.92
15	0.95	0.95	0.97	**0.98**	0.75	0.80
20	0.95	0.95	0.97	**0.96**	0.71	0.81
25	0.95	0.95	**0.97**	**0.97**	0.61	0.79
30	0.94	0.94	**0.96**	0.94	0.61	0.77
35	0.94	0.94	0.94	**0.96**	0.59	0.76
40	0.93	**0.94**	**0.94**	0.93	0.59	0.72
45	0.92	**0.94**	**0.94**	0.93	0.60	0.69
50	0.92	**0.94**	**0.94**	0.93	0.59	0.68

are used as training data. The cross-validation process is then repeated 10 times, which each of the 10 folds used exactly once as the test data. For all selected shape descriptors, the default parameters in the respective PCL implementations were used. All tests were performed with an i7, 2.40GHz processor and 16GB RAM.

5.1. Descriptiveness

As mentioned above, GOOD has a parameter called *number of bins* that has effect on descriptiveness, efficiency and robustness. Therefore, it must be well selected to provide a good balance between recognition performance, memory usage and computation time. The descriptiveness of the proposed descriptor with respect to varying *number of bins* was evaluated using Washington dataset. Results are presented in Fig. 4 (*left*) and Table 1.

In these experiments, the configurations that obtained the best precision and recall figures were number 3 and 5. Although, a large number of bins provides more details about the point distribution, it increases computation time, memory usage and sensitivity to noise. Therefore, since the difference to other configurations is not very large, we prefer to use configuration number 1 (i.e. $b = 5$) which displays a good balance between recognition performance, memory usage, and processing speed. The accuracy of the proposed system with this configuration was 92%. It shows that the overall performance of the recognition system is promising and the proposed descriptor is capable of providing distinctive global feature for the given object. The following results are computed using this the default value, unless otherwise noted.

5.2. Scalability

A set of experiments was carried out to evaluate the performance of the proposed descriptor on the Washington dataset, concerning its scalability with respect to varying numbers of categories.

Results are depicted in Fig. 4 (*center*) and (*right*). One important observation is that the accuracy decreases in all approaches as more categories are used (Fig. 4 (*center*)). This is expected since the number of categories known by the system makes the classification task more difficult and the difference in performance between descriptors becomes smaller. Moreover, it can be concluded from Table 2 that when the number of object categories increases (i.e. more than 35 categories), VFH and GOOD descriptors achieve the best accuracy and stable performance regarding varying numbers of categories Table 3. It is clear from Fig. 4 (*right*) that the computation time of our approach is significantly smaller than VFH, GRSD and GFPFH. However, GOOD, VFH and ESF descriptors obtain an acceptable scalability regarding varying numbers of categories, the scalability of GRSD and GFPFH are very low and their performance drops aggressively when the number of categories increases. Although ESF descriptor achieves better performance than our approach with 5 bins (i.e. GOOD 5bins), the length of ESF descriptor (i.e. compactness) is around 8.5 times more than our descriptor (see Table 4). It is notable that whenever the size of dataset is larger than 35 object categories, the difference between EFS performance and our approach with 5 bins, is equal or less than 1% and in the similar situation our approach with 15 bins (i.e. GOOD 15bins) works better than ESF descriptor.

5.3. Robustness

The robustness of the proposed object descriptor with respect to different levels of Gaussian noise and varying mesh resolutions was evaluated and compared with other global object descriptors. These experiments were run on the mentioned Restaurant Object Dataset.

Table 3
Summary of robustness to Gaussian noise experiments.

Gaussian noise(mm)	Accuracy				
	GOOD 5bins	VFH	ESF	GFPFH	GRSD
1	0.94	**0.95**	0.90	0.40	0.66
2	**0.93**	**0.93**	0.74	0.17	0.61
3	0.92	**0.93**	0.49	0.10	0.51
4	0.91	**0.94**	0.32	0.09	0.35
5	0.89	**0.91**	0.24	0.09	0.26
6	**0.85**	**0.85**	0.23	0.09	0.19
7	**0.83**	0.78	0.23	0.09	0.12
8	**0.78**	0.57	0.22	0.09	0.08
9	**0.69**	0.42	0.23	0.09	0.10
10	**0.67**	0.36	0.22	0.09	0.09

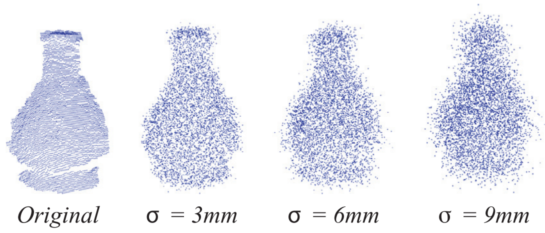


Fig. 5. An illustration of a *Vase* object with different levels of Gaussian noise.

5.3.1. Gaussian noise

Ten levels of Gaussian noise with standard deviations from 1 to 10 mm were added to the test data. For a given test object, Gaussian noise is independently added to the *X*, *Y* and *Z*-axes. As an example, a *Vase* object with three levels of standard deviation of Gaussian noise ($\sigma = 3\,\text{mm}, \sigma = 6\,\text{mm}, \sigma = 9\,\text{mm}$) is depicted in Fig. 5. The results are presented in Fig. 6 (*left*) and Table 3. An important observation can be made from Figs. 6 and 4. Although GOOD, ESF and VFH descriptors achieved a really good performance on noise free data, GOOD outperformed ESF, GFPFH and GRSD descriptors by a large margin under all levels of Gaussian noise. While the performance of VFH descriptor was similar to our approach under a low-level noise (i.e. $\sigma \leq 6\,\text{mm}$), our shape descriptor outperformed all descriptors under high levels of noise.

It can be concluded from this observation that GOOD descriptor is robust to noise due to use an stable, unique and unambiguous object reference frame. In contrast, since the VFH and GFPFH descriptors are rely on surface normals to calculate their shape descriptions, they are highly sensitive to the noise. GRSD employs radial relationships to describe the geometry of points at each voxel cell and ESF uses distances and angels between randomly sampled points to generate a shape description; therefore, GRSD and ESF are also sensitive to the noise and its performance decrease rapidly when the standard deviation of the Gaussian noise increases. In addition, GOOD descriptor uses three distribution matrices that are constructed based on orthographical projection, therefore less affected by noise (i.e. in each orthographic projection one dimension is discarded).

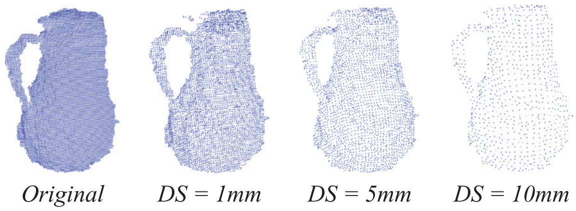


Fig. 7. An illustration of a *Flask* object with different levels of downsampling.

5.3.2. Varying point cloud density

Two sets of experiments were carried out to examine the robustness of the proposed descriptor with respect to varying point cloud density. In the first set of experiments, the original density of training objects was kept and the density of testing objects was reduced (downsampling) using a voxelized grid approach[3] In the second set of experiments, the original density was kept in testing objects and reduced in training objects. This is initiated with a root volume element (voxel) and the eight children voxels in which each internal node has exactly eight children nodes. These are recursively subdivided until all voxels contain at most one point or the minimum voxel size is reached (i.e. The cloud is divided in multiple voxels with the desired resolution). Afterwards all the points that fall into the same voxel will be downsampled with their centroid. In this evaluation, each object (either test or train) is downsampled using five different voxel sizes including {1, 5, 10, 15, 20} millimetre. An illustration example of a *Flask* object with four level of downsampling is depicted in Fig. 7. The robustness results regarding varying point cloud density in test and train data are presented in Fig. 6.

From experiments of reducing density of test data (i.e. Fig. 6(left)), it was found that our approach is more robust than the other descriptors concerning low-level downsampling (i.e. $DS \leq 3.5\,\text{mm}$) and works slightly better than the other in high-level downsampling resolution (i.e. $DS \geq 18\,\text{mm}$). In contrast, the performance of VFH, ESF and GRSD were better than GOOD descriptor

[3] http://pointclouds.org/documentation/tutorials/voxel_grid.php.

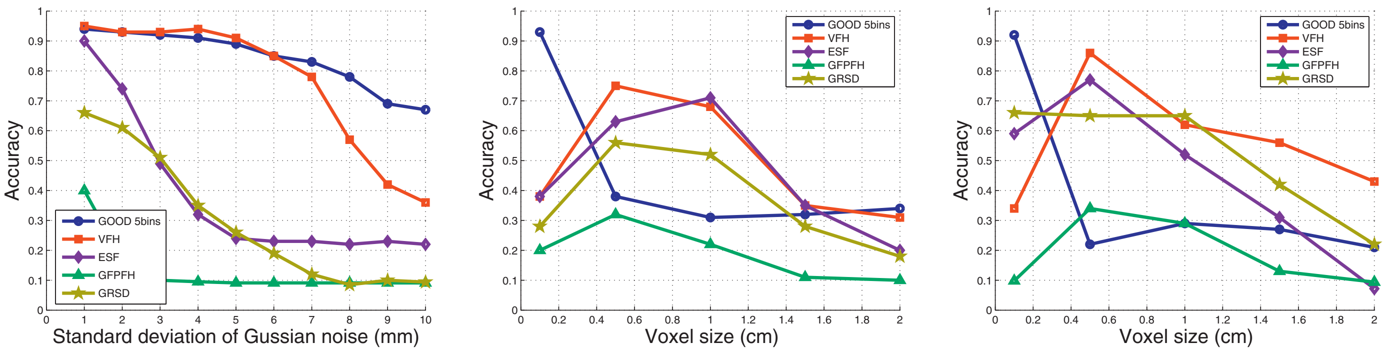
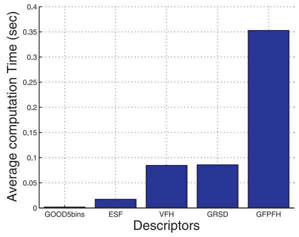


Fig. 6. The robustness of the selected descriptors to different level of Gaussian noise and varying point cloud density: (*left*) different levels of Gaussian noise applied to the test; (*center*) different levels of downsampling applied to the test data; (*right*) different levels of downsampling applied to the train data.

Table 4
Length of selected 3D shape descriptors.

No.	Descriptor	Feature length (float)	Adjustable length	Implementation
1	GFPFH	16	No	PCL 1.7
2	GRSD	21	No	PCL 1.8
3	GOOD	75	Yes	–
4	VFH	308	No	PCL 1.7
5	ESF	640	No	PCL 1.7



Descriptor name	Computation Time (sec)
GOOD	0.00168
ESF	0.01823
VFH	0.07440
GRSD	0.08555
GFPFH	0.42680

Fig. 8. Average computation time of the selected descriptors on 20 randomly selected objects from the RGB-D dataset.

in mid-level downsampling resolution (i.e. 3.5 mm < DS < 18). The performance of GFPFH was very low under all level levels of point cloud resolution. Besides, it can be concluded from Fig 6 (right) that when the level of up-sampling increases, VFH, ESF and GRSD descriptors achieve better performance than GOOD and GFPFH descriptors.

5.4. Efficiency

In this subsection, evaluations regarding computational efficiency and memory footprint (i.e. the amount of main memory that a program uses or references while running) are presented and discussed.

5.4.1. Memory footprint

The length or size of a descriptor, has direct influence on memory usage and computation time in object recognition process (see Fig. 4). The length of all descriptors used in this evaluation is given in Table 4.

Although GFPFH and GRSD are the tow most compact descriptors in this evaluation (see Table 4), their computation time and are not good as depicted in Figs. 4 and 8. Our approach is the third compact descriptor that provides good balance between computation time and descriptiveness with 75 floats. However, VFH and ESF descriptors achieve a good description power, their feature length is around 4.1 and 8.5 times larger than our approach and 20 and 40 times larger than GFPFH descriptor respectively. ESF is the lowest compact descriptor compared to all the other descriptors.

5.4.2. Computation time

Several experiments were performed to measure computational time of all descriptors used in this evaluation. Since the number of object's points directly affects the computational time, we calculate the average time required to generate a description for 20 randomly selected objects from the RGB-D dataset. Fig. 8 compares the average computation time of the selected object descriptors in which several observations can be made; first, GOOD descriptor is the most computation time efficient descriptor. In contrast, GFPFH descriptor is the computationally most expensive descriptor. ESF, VFH and GRSD descriptors achieve a medium performance in terms of computation time. Overall, GOOD descriptor achieves the best performance, which is around 10 times better performance than ESF and 44, 50 and 254 times better performance than VFH, GRSD and GFPFH descriptors. VFH, GRSD and GFPFH descriptors are extremely time consuming descriptors. The underlying reason is that GOOD descriptor works directly on 3D point clouds and requires neither triangulation of the object's points nor surface meshing. According to the evaluations our approach is competent for robotic applications with strict limits on the memory footprint and computation time requirements.

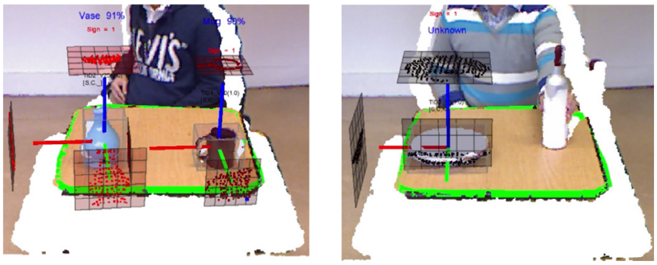


Fig. 9. Two snapshots showing the object perception system performing object recognition and pose estimation using the GOOD descriptor; (left) the instructor puts a *Mug* and a *Vase* on the table. The gray bonding boxes and red, green and blue lines signal the pose of the object and the GOOD descriptions are visualized and computed; this frame shows that the system is able to compute the GOOD description and estimate pose of objects in the scene. Moreover, it demonstrates that the *Vase* and the *Mug* are properly recognized; (right) A *Plate* enters the scene. Its shape description and pose are computed and visualized. Because there is no prior knowledge about plates, it is classified as *Unknown* [13].

5.5. System demonstration

To show all the described functionalities and properties of the proposed GOOD descriptor, a real demonstration was performed. For this purpose, GOOD has been integrated in the object perception system presented in [13,20] and [19] (see Fig. 9). In this demonstration a table is in front of a robot and two users interact with the system. During the demonstration, users presented objects to the system and provided the respective category labels. Therefore, throughout this session, the system must be able to detect, conceptualize and recognize unknown (i.e. new) objects. It should be noted that a constraint has been set on the Z axis that the initial direction of Z axis of objects' LRF should be similar to direction of Z axis of the table. It is assumed that there is no learned categories in the memory at the beginning of the demonstration. It was observed that the proposed object descriptor is capable to provide distinctive global feature for recognizing different type of objects. It also estimates pose of objects and build orthographic projections for object manipulation purposes. A video of this demonstration is available in: https://youtu.be/iEq9TAaY9u8.

6. Conclusion

This paper presented a global object descriptor named GOOD (i.e. Global Orthographic Object Descriptor) that provides a good trade-off between descriptiveness, computation time and memory usage, allowing concurrent object recognition and pose estimation. For an object, GOOD is computed on a unique and repeatable local reference frame. It is calculated with the discretization of the three orthographic projections and their concatenation to form a single description for the given object. A set of experiments were carried out to assess the performance of GOOD and compare it with other state-of-art descriptors with respect to several characteristics including descriptiveness, scalability, robustness (Gaussian noise and varying low-level point cloud density) and efficiency (memory footprint and computation time).

Experimental results show that the overall classification performance obtained with GOOD is comparable to the best performances obtained with the state-of-the-art descriptors. GOOD outperformed the selected state-of-the-art descriptors (i.e. VFH, ESF,

Please cite this article as: S.H. Kasaei et al., GOOD: A global orthographic object descriptor for 3D object recognition and manipulation, Pattern Recognition Letters (2016), http://dx.doi.org/10.1016/j.patrec.2016.07.006

GRSD and GFPFH descriptors), achieving appropriate descriptiveness and significant robustness to Gaussian noise. GOOD was robust to varying low-level point cloud density too. The accuracy of VFH, ESF and GRSD was better than GOOD in the case of varying medium and high point cloud density. In addition, GOOD obtained the best computation time performance. Besides, GOOD demonstrates the capability of estimating objects' poses and building orthographic projections for object manipulation purposes.

We are currently working on integrating and using the GOOD descriptor for manipulation purposes and we would like to put the source code of the GOOD descriptor available to the research community in the ROS[4] repository and Point Cloud Library[5] in the near future.

Acknowledgments

This work was funded by National Funds through FCT project PEst-OE/EEI/UI0127/2014 and FCT scholarship SFRH/BD/94183/2013.

References

[1] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R.B. Rusu, S. Gedikli, M. Vincze, Point cloud library, IEEE Robot. Autom. Mag. 1070 (9932/12) (2012).
[2] A. Andreopoulos, J.K. Tsotsos, 50 years of object recognition: Directions forward, Comput. Vis. Image Underst. 117 (8) (2013) 827–891.
[3] L. Bo, K. Lai, X. Ren, D. Fox, Object recognition with hierarchical kernel descriptors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, IEEE, 2011, pp. 1729–1736.
[4] R. Bro, E. Acar, T.G. Kolda, Resolving the sign ambiguity in the singular value decomposition, J. Chemometr. 22 (2) (2008) 135–140.
[5] H. Chen, B. Bhanu, 3d free-form object recognition in range images using local surface patches, Pattern Recogn. Lett. 28 (10) (2007) 1252–1262.
[6] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley & Sons, 2012.
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, 2009, pp. 248–255.
[8] H.Q. Dinh, S. Kropac, Multi-resolution spin-images, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, 1, IEEE, 2006, pp. 863–870.
[9] A. Frome, D. Huber, R. Kolluri, T. Bulow, J. Malik, Recognizing objects in range data using regional point descriptors, in: Computer Vision-ECCV 2004, Springer, 2004, pp. 224–237.
[10] Y. Guo, F.A. Sohel, M. Bennamoun, M. Lu, J. Wan, Trisi: A distinctive local surface descriptor for 3d modeling and object recognition., in: GRAPP-IVAPP, 2013, pp. 86–93.
[11] B.K. Horn, Extended gaussian images, Proc. IEEE 72 (12) (1984) 1671–1686.
[12] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 433–449.
[13] S. Kasaei, M. Oliveira, G. Lim, L. Seabra Lopes, A.M. Tome, Interactive open-ended learning for 3d object recognition: an approach and experiments, J. Intell. Robot. Syst. 80 (3–4) (2015) 537–553.
[14] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view rgb-d object dataset, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 1817–1824.
[15] Y. Li, S. Pirk, H. Su, C.R. Qi, L.J. Guibas, Fpnn: field probing neural networks for 3d data, arXiv preprint arXiv:1605.06240 (2016).
[16] Z.-C. Marton, D. Pangercic, R.B. Rusu, A. Holzbach, M. Beetz, Hierarchical object geometric categorization and appearance classification for mobile manipulation, in: 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2010, IEEE, 2010, pp. 365–370.
[17] A. Mian, M. Bennamoun, R. Owens, On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes, Int. J. Comput. Vis. 89 (2–3) (2010) 348–361.
[18] G.A. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.
[19] M. Oliveira, G.H. Lim, L. Seabra Lopes, S. Hamidreza Kasaei, A.M. Tome, A. Chauhan, A perceptual memory system for grounding semantic representations in intelligent service robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), 2014, IEEE, 2014, pp. 2216–2223.
[20] M. Oliveira, L.S. Lopes, G.H. Lim, S.H. Kasaei, A.M. Tomé, A. Chauhan, 3d object perception and perceptual learning in the race project, Robot. Auton. Syst. (2015).
[21] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Trans. Graph. 21 (4) (2002) 807–832.
[22] G. Pang, U. Neumann, Fast and robust multi-view 3d object recognition in point clouds, in: International Conference on 3D Vision (3DV), 2015, IEEE, 2015, pp. 171–179.
[23] G. Pasqualotto, P. Zanuttigh, G.M. Cortelazzo, Combining color and shape descriptors for 3d model retrieval, Signal Process.: Image Commun. 28 (6) (2013) 608–623.
[24] D.M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011).
[25] D. Regazzoni, G. de Vecchi, C. Rizzi, Rgb cams vs rgb-d sensors: low cost motion capture technologies performances and limitations, J. Manuf. Syst. 33 (4) (2014) 719–728.
[26] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (fpfh) for 3d registration, in: IEEE International Conference on Robotics and Automation, 2009. ICRA'09., IEEE, 2009, pp. 3212–3217.
[27] R.B. Rusu, G. Bradski, R. Thibaux, J. Hsu, Fast 3d recognition and pose using the viewpoint feature histogram, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, IEEE, 2010, pp. 2155–2162.
[28] R.B. Rusu, A. Holzbach, M. Beetz, G. Bradski, Detecting and segmenting objects for mobile manipulation, in: IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), 2009, IEEE, 2009, pp. 47–54.
[29] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3d point cloud based object maps for household environments, Robot. Auton. Syst. 56 (11) (2008) 927–941.
[30] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 945–953.
[31] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 356–369.
[32] W. Wohlkinger, M. Vincze, Ensemble of shape functions for 3d object classification, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2011, IEEE, 2011, pp. 2987–2992.
[33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: a deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.
[34] Y. Zhong, Intrinsic shape signatures: A shape descriptor for 3d object recognition, in: IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), 2009, IEEE, 2009, pp. 689–696.

[4] http://www.ros.org/.
[5] http://pointclouds.org/.

Please cite this article as: S.H. Kasaei et al., GOOD: A global orthographic object descriptor for 3D object recognition and manipulation, Pattern Recognition Letters (2016), http://dx.doi.org/10.1016/j.patrec.2016.07.006