

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277328662>

# 3D Object Recognition Based on Local and Global Features Using Point Cloud Library

Article · May 2015

---

CITATIONS

2

---

READS

772

3 authors, including:



[Khaled M. Alhamzi](#)

Mansoura University

2 PUBLICATIONS 3 CITATIONS

SEE PROFILE



[Mohammed Elmogy](#)

Mansoura University

72 PUBLICATIONS 81 CITATIONS

SEE PROFILE

## 3D Object Recognition Based on Local and Global Features Using Point Cloud Library

<sup>1</sup>Khaled Alhamzi, <sup>2</sup>Mohammed Elmogy, <sup>3</sup>Sherif Barakat

<sup>1</sup>Department of Information System, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt, kalhamzi@yahoo.com

<sup>2</sup>Department of Information Technology, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt, melmogy@mans.edu.eg

<sup>3</sup>Department of Information System, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt, sherifiib@yahoo.com

### Abstract

3D object recognition from point clouds is considered as a field of research that is growing fast. Based on the types of features used to represent an object, 3D object recognition approaches can be classified into two broad categories—local and global feature-based techniques. Local feature-based techniques are more robust to clutter and partial occlusions that are frequently present in a real-world scene. Whereas, global feature-based techniques are suitable for model retrieval and 3D shape classification especially with the weak geometric structure. Most systems for 3D object recognition use either local or global feature-based techniques. This is because of the difficulty of integrating a set of local features with a single global feature vector in an appropriate manner.

In this paper, a 3D object recognition system based on local and global features of the objects using Point Cloud Library (PCL) is proposed. The proposed system uses a hybrid technique based on Viewpoint Feature Histogram (VFH) method and Fast Point Feature Histogram (FPFH) method. VFH method is used as a global descriptor to recognize the object. Whereas, FPFH method is used as a local descriptor to estimate the position of the object in the real-world scene. The performance of the proposed system is evaluated by calculating the accuracy of the recognition process. The experimental results reveal that this system performs well on the tested objects as compared to some state of the art techniques.

**Keywords:** 3D Object Recognition, Point Cloud Library (PCL),

Fast Point Feature Histogram (FPFH) Descriptor, Viewpoint Feature Histogram (VFH) Descriptor

### 1. Introduction

Object recognition from 3D point clouds is a recent and promising research area in the computer vision field. The objective of the 3D object recognition is to identify objects correctly in the point cloud and determines their poses (i.e., location and orientation) [1].

The existing approaches for 3D object recognition can be classified into two broad categories [1][2][3]: global feature-based methods and local feature-based methods. Global feature-based methods are based on the whole visual appearance of the object. They have the ability to generalize an entire object with a single vector. Therefore, they can be directly used in standard classification techniques. An example for this category is Viewpoint Feature Histogram (VFH) method [4]. However, they ignore the shape details and need a prior segmentation of the object from the scene [5]. Consequently, these methods are not suitable for the recognition of a partially visible object from cluttered scenes [3]. In contrast, the local feature-based methods are based only on local geometric features extracted from the interest points. However, local features may require specialized classification techniques to deal with the feature vectors per object. They are better in dealing with clutter and occlusion than the global feature-based methods [3].

Most systems for 3D object recognition tend to use one of these two approaches. On the other hand, designing a system for 3D object recognition needs powerful processing tools and algorithms to process 3D data. The Point Cloud Library (PCL) is an open source and large-scale library for processing 2D/3D images. The PCL framework contains numerous state-of-the-art algorithms and tools [6].

This paper is organized as follows. Section 2 discusses the two types of the techniques for 3D object recognition. The first type is based on local features of the object and the other one based on global features of the object. Section 3 reviews some of the related work about using local and global features in 3D object recognition. In Section 4, the proposed architecture of our 3D object recognition system, which is based on the integration of local and global features, is introduced. After that, Section 5 presents the implementation and the experimental results. Finally, Section 6 contains the conclusion and our future work.

## 2. 3D Object Recognition Pipelines

In this section, we will evaluate the available 3D descriptors in the current version of PCL. There are some 3D descriptors in PCL library that will not be discussed in this paper because they are not directly applicable to point cloud data [7].

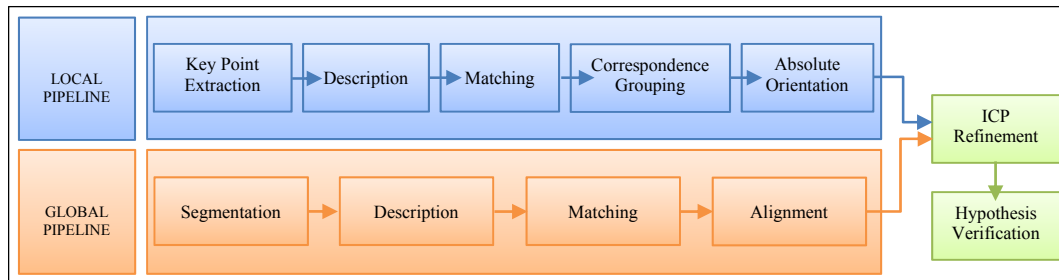
In PCL, 3D descriptors are divided into two types [8][9]: local and global. The process for computing and utilizing each type (recognition pipeline) is different. On the other hand, any descriptor has a method for computing unique values for a point. Some of them use the distances between the points. Others use the difference between the angles of the normals of the point and its neighbors. A given descriptor can be better with occlusions and partial views of objects, and the other one can be a scale invariant.

Table 1 contains some of the main features of the 3D descriptors presented in this paper. The second column shows the descriptor type. The third column shows the dimensionality of the descriptor's signature (the size of the histogram). The last column shows if the algorithm requires the computation of the surface normals at each point.

As shown in Figure 1, 3D descriptors can be used in composition with other components to build a full object recognition system. Practically, there are two entire pipelines for recognition system can be used: one depends on global descriptors that require the notion of objects, thus, need a pre-segmentation step. Whereas, the other one depends on the local descriptors that are computed locally at key points, thus, do not need a pre-segmentation step.

**Table 1.** The main features of the 3D descriptors that have been evaluated (Y=Yes; N=No).

Descriptors	Type	Size	Normal
SHOT	Local	352+9	Y
COLORSHOT	Local	1344+9	Y
PFH	Local	125	Y
PFHRGB	Local	250	Y
FPFH	Local	33	Y
ESF	Global	640	N
VFH	Global	308	Y
CVFH	Global	308	Y



**Figure 1.** The block diagrams of the local and global 3D descriptors pipelines [9].

### 2.1. Local Descriptors

3D local descriptors are computed for individual points that are given as input. More specifically, every point is associated with a descriptor that describes the local geometry of a point [9]. Generally,

3D local descriptors are used for specific applications like object recognition, surface registration, and local surface categorization.

### 2.1.1. Signature of Histograms of Orientations (SHOT)

The SHOT Local descriptor [10] depends on obtaining a local reference frame by the eigenvalue decomposition of an input point. It is rotation invariant and robust to clutter and noise. The descriptor for each key point is created by computing local histograms including geometry information of point locations in a spherical support structure. The final descriptor is created by concatenated all such local histograms together using 11 bins and 32 sectors of the spherical grid, which produces 352 values. There is also a color version that is called COLOR Signature of Histograms of Orientations (CSHOT) [11]. CSHOT descriptor adds color information to the SHOT descriptor resulting in a 1344 value descriptor (plus nine values to describe the local reference frame).

### 2.1.2. Point Feature Histogram (PFH)

Local descriptor like point feature histogram (PFH) [12] can be classified as geometry-based descriptor [9]. The purpose of this descriptor is to encode a local surface. It represents the orientation of normals and distances between point pairs. Point pairs are created a point with coordinates  $p_i$  and the points in its local neighborhood with coordinates  $p_j$ . Next, for each pair, a fixed reference frame is computed from their normals. Using this reference frame, the difference between the normals can be encoded with three angles. These angles, with the Euclidean distance between the points, are kept. These are binned into a 125 bin histogram. The final descriptor is the combination of the histograms of each angle.

### 2.1.3. Point Feature Histogram RGB (PFH-RGB)

PFH-RGB is another version of PFH that includes color information [7]. In addition to 125 values of the PFH, the PFH-RGB contains 125 extra values is a 5 binned histogram of the color information around the key point, giving the total size of 250 values for PFH-RGB. The use of color information in this descriptor increases the robustness of the technique, but more memory and time are required.

### 2.1.4. Fast Point Feature Histogram (FPFH)

The Fast Point Feature Histograms (FPFH) technique [13] is a fast and simplified version of the PFH descriptor. The PFH descriptor provides accurate results, but it is very computationally expensive to perform at real-time. Therefore, the FPFH descriptor was derived. The first step is to compute the histogram of the three angles among a point  $p$  and its  $k$ -nearest neighbors. This produces the Simplified Point Feature Histogram (SPFH). In the second step, for each point  $p$ , the  $k$ -neighbors are determined again, computing their SPFH. The final descriptor is created weighting each these SPFHs:  $FPFH(p) = SPFH(p) + 1/k \sum_{i=1}^k SPFH(i)/w_i$ . The three angles are represented into 11 bin histograms that are represented into a single 33 bin FPFH descriptor.

## 2.2. Global Descriptors

3D global descriptors are considered as the representation of the object geometry [9]. They are not computed for individual points, but for the whole cluster, that represents an object. For this reason, a pre-processing step (segmentation) is required, in order to retrieve possible candidates. They are only calculated for subsets of the point clouds that are likely to be objects. In general, global descriptors are used for object recognition, 3D shape classification, geometric analysis, model retrieval, and pose estimation.

### 2.2.1. Ensemble of Shape Functions (ESF)

The ESF [14] consists of ten 64-bin-sized histograms resulting in a single 640 value histogram for input point cloud. It is a combination of three shape functions [15] point distance ( $D_2$ ), angle ( $A_3$ ), and area ( $D_3$ ). ESF global descriptor is unique since it does not require the use of normals information to describe the cloud.

In this descriptor, a voxel grid algorithm is used as an approximation of the real surface. A voxel grid iterates over each the points in the cloud, for each iteration, three random points are chosen. For these points, the shape functions are computed [8]:

- $D_2$  computes the distances between two randomly selected points. After that, for each pair of points, it checks if the connecting lines between points reside: on the surface, off the surface, or mixed. Based on this, the distance value will be binned to one of the three possible histograms: on, off or mixed.
- $A_3$  computes the angle enclosed by the points. Next, the value is binned based on how the line opposite to the angle is (on, off or mixed).
- $D_3$  computes the area of the triangle formed by the 3 points. Similar to  $D_2$ , the result is also classified as on, off or mixed.

### 2.2.2. Viewpoint Feature Histogram (VFH)

The VFH [4] is a global descriptor that is based on the FPFH local descriptor. The VFH consists of two parts: a viewpoint direction component and an extended FPFH component. To compute the first part, the centroid of the object is found, which is the point that results from averaging the three coordinates of all points. Next, the vector between the viewpoint and the centroid is computed, and normalized. Lastly, for all points in the cluster, the angle between this vector and their normal is calculated, and the result is represented by a histogram. The second part is computed like the FPFH. The resulting four histograms (one for the viewpoint component and three for the extended FPFH component) are concatenated to build the final VFH descriptor. Thus, the total length of the VFH descriptor is 308.

### 2.2.3. Clustered Viewpoint Feature Histogram (CVFH)

CVFH global descriptor [16] is an extension to Viewpoint Feature Histogram. CVFH is suitable when the cluster of an object is missing many points. In this descriptor, rather than computing a single VFH histogram for the entire cluster, the object is split into stable regions. These regions are obtained by first removing the points with a high curve and after that using smooth region growing segmentation algorithm. The process of obtaining the CVFH, as in the following steps [7]: 1) determine the set  $S$  of stable regions; 2) for each  $s_i \in S$ , find the centroid ( $c$ ) and its normal ( $n_c$ ); 3) build a local reference frame ( $u_i, v_i, w_i$ ) like in the VHF but using  $c$  and  $n_c$  instead of the centroid and particular normal for the entire input cloud; 4) compute the histograms of the angles ( $\alpha, \phi, \theta, \beta$ ) as in VHF; 5) compute the Shape Distribution Component (SDC) by  $SDC = \frac{(c-p_i)^2}{\max\{(c-p_i)^2\}}$ ,  $i = 1, |S|$ . The CVFH is given by the concatenated histograms ( $\alpha, \phi, \theta, SDC, \beta$ ) which is a 308-bin histogram.

## 3. Related Work

There have been several previous work offered in 3D object recognition using local or global features. Therefore, in this section we give a brief review of some previously published work on object recognition from 3D cloud data. For example, Filipe and Alexandre [17] presented an explanation and evaluation of the current key point detectors available to be used in the PCL. They evaluated the invariance of the 3D key point detectors depending on to translations, scale, and rotations changes. They used the relative repeatability rate and the absolute as criteria for performance evaluation. They concluded from their experiments that SIFT3D and ISS3D achieved the best result, and the ISS3D showed the most invariant. Currently, they seeking to extension of some methods proposed for the key

point detectors to deal with large occlusions and rotations, and the evaluation of the best integration of key point detectors and descriptors.

Aldoma et al. [9] reviewed of many state-of-the-art 3D shape descriptors intent at object recognition. These descriptors existed in the PCL and available to the public. For object recognition system, they presented two complete pipelines built using parts obtainable in PCL: first pipeline depends on global descriptors while the other pipeline relies on local descriptors.

Alexandre [7] presented a comparative evaluation of some available descriptors in the PCL. The author used two approaches for the key point extraction. The first is Harris3D key point extractor. The other one consists of sub-sampling the input cloud and considering all the sub-sampled points as key points. Eventually, he concluded the following: 1) increasing the number of key points improves recognition results; 2) the descriptors that use color information must be used instead of that use only shape information; 3) the COLORSHOT Descriptor showed good performance for recognition and time.

Tang et al. [18] proposed an object recognition system for instance recognition and object localization in cluttered scenes using the RGB images and the depth information. Their system can handle illumination changes, occlusions, multiple objects, and multiple instances of the same object. They are studying the extensions to textureless objects, and planning to provide high-reliability 3D approaches verification.

Kasaei et al. [19] presented an approach to 3D instance-based object learning and recognition in a restaurant environment for an autonomous service robot. This method depends on the local shape descriptor known as a spin image. The system can deal with objects of unknown categories as well as false object candidates. This approach can also work well with noisy data obtained by Microsoft Kinect, as well as, the approach can detect and recognize free-form objects. For future work, they intend to study the possibility of improving performance using new 3D shape descriptors and integrate further tools for open-ended learning.

Bo et al. [20] presented the RGB-D dataset, and they proposed the Hierarchical Kernel Descriptors method, suggesting kernels based on depth information, including local shape.

Xie et al. [21] proposed object detection pipeline, which is based on dense feature models and RGB-D data to address an unorganized household or office environment. They also used color and shape features over a pose-verification stage to enhance performance. Their system demonstrated high performance and computational tractability. Currently, their system is not designed to handle non-textured and transparent objects. In addition, they are trying to collect a larger dataset of the hundreds of objects to evaluate further.

The primary objective of this paper is designing a system for object recognition from 3D point cloud based on local and global features. This system uses a hybrid technique based on VFH method and FPFH method. VFH method is used as a global descriptor to recognize the object. Whereas, FPFH method is used as a local descriptor to estimate the position of the object in the real-world scene.

#### 4. System Architecture

In this paper, we proposed a system for 3D object recognition and pose estimation. More specifically, our goal is to recognize non-rigid objects from 3D point cloud data and estimate their position in the real-world scene. Most of the previous 3D object recognition systems used either local or global features. Whereas, our proposed system uses the integration of local and global features of the object.

As shown in Figure 2, the proposed system consists of two phases: a training phase and a testing phase. During the training phase, the objects used to train the system are represented as 3D clouds, and the real objects are acquired by a depth sensor such as the Microsoft Kinect. For each object, 3D features are extracted and associated these characteristics to local descriptors. In addition, shape and geometry of the object are described by the global descriptor in a single vector. Lastly, these features and descriptors are stored in the database as a model of the object.

At the test phase, the proposed system is given below, for input point cloud:

- i. The input point cloud is pre-processed for key points extraction.
- ii. FPFH local descriptor and VFH global descriptor are computed from the pre-processed input point cloud.
- iii. In the matching step, VFH global descriptor is used to recognize the object.

- iv. To determine the position of the object in the scene, FPFH local descriptor uses the matching output to align the points in the source model to the points in the target model.

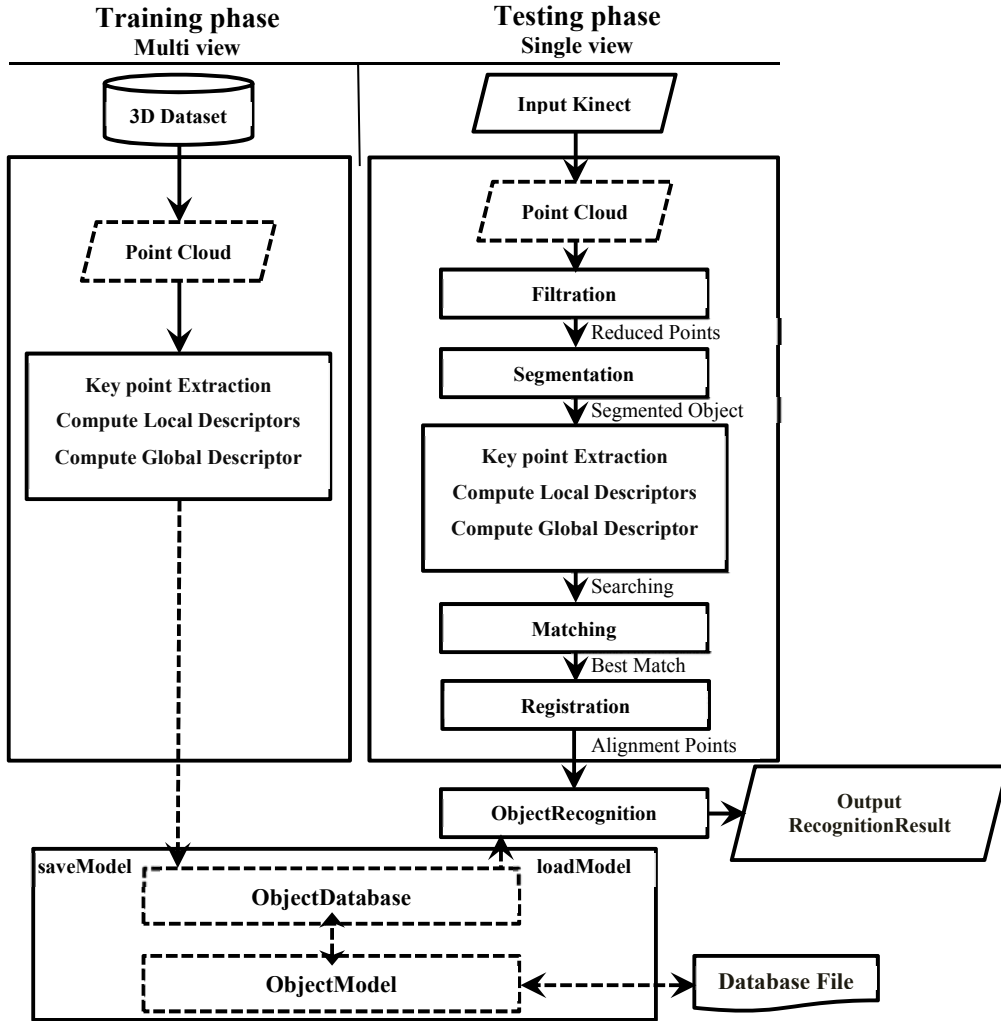


Figure 2. The architecture of the training and testing framework.

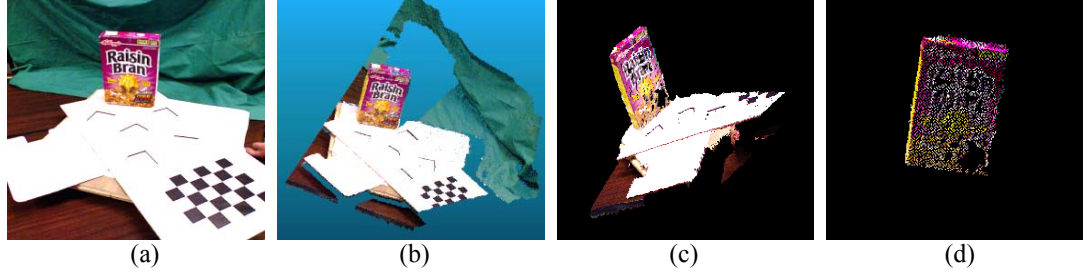
#### 4.1. Filtering

Modify the point clouds or point attributes as the yield from the Kinect sensor begin before processing algorithms. As an initial step, we applied a series of filters available in the PCL library on input point cloud in order to reduce undesirable data. These filters are as follows: 1) threshold depth filter is used to remove points with depth values that are too large or too small; 2) downsample filter is used to reduce the number of points; and 3) remove outliers filter is used to remove all points with too few local neighbors, as shown in Figure 3.

#### 4.2. Segmentation

As global features require the notion of the object, the scene needs to be segmented to extract objects. There are many techniques that perform this task. In this paper, we chose a simple but effective technique based on finding the dominant plane and subtracting it. After that, the Euclidean cluster extraction step is used to group a cloud into contiguous clusters. Consequently, we use SAMPLE

Consensus Segmentation (SAC Segmentation) method and an Extract Indices filter to finding the dominant plan and subtract it. In addition, Euclidean Cluster Extraction method used to cluster extraction in an Euclidean sense. This method based on threshold  $t$ , that refers to how close the two points needed to be belong to the same object.



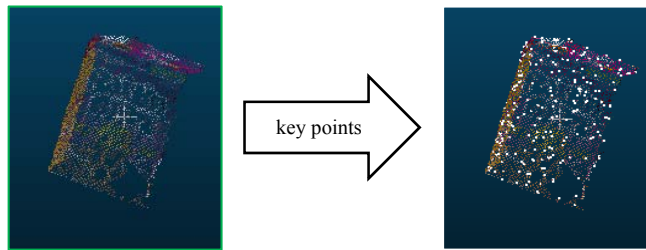
**Figure 3.** The point cloud processing for one of the used objects. (a) Example test image (b) Example test point cloud (c) Point cloud after applied filtration (d) Segmented object.

### 4.3. Key Point Extraction and Description

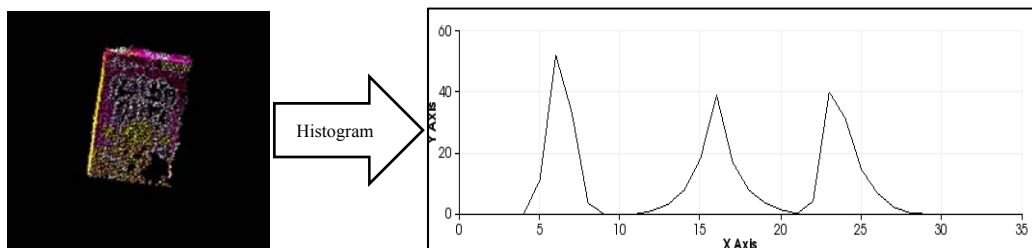
The output of the segmentation step represents the individual objects from the scene, as shown in Figure 3. After that, for each object we compute surface normal point, key points and local and global feature descriptors. These values enable us compute similar values for the same spot of a surface in two sets.

Surface normals could be calculated for the points of a cloud, which is considered as a feature. We used normal estimation to estimate a cloud's surface normals. The most important step in the 3D perception pipeline, based on local descriptors, is the extraction of 3D key points from cloud data. There are many 3D key point detection methods in PCL library. For instance, the Harris3D, SIFT3D, ISS3D, SUSAN detectors. In this paper, we chose SIFT3D detector because it produces high-precision results as approved in [17].

Once key points are extracted from cloud data by the SIFT3D detector, they are associated with the local description, as shown in Figure 4. FPFH local descriptor is used to compute the local feature descriptor of each key point, as illustrated in Figure 5. In addition, VFH global descriptor is used to describe the shape and geometry of the segmented object as single histogram. (See Figure 6).

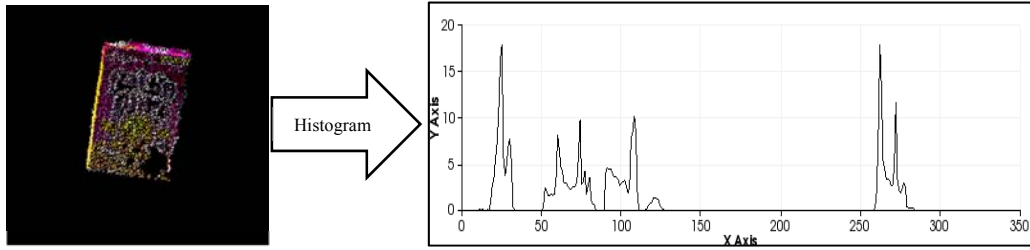


**Figure 4.** The result of the SIFT3D detector in key points extraction step.



**Figure 5.** An example of the resulting FPFH local descriptor for one of the used objects. Left part, the x-axis represents the number of histogram bins, and the y-axis represents the percentage of points falling in each bin.



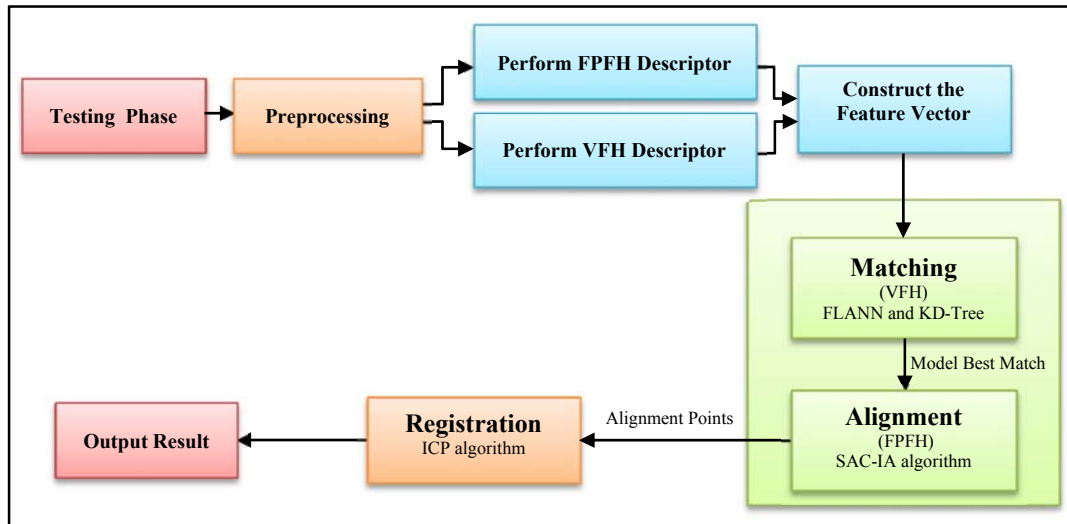


**Figure 6.** An example of the resulting VFH global descriptor for one of the used objects. Left part, the x-axis represents the number of histogram bins, and the y-axis represents the percentage of points falling in each bin.

#### 4.4. Matching and Registration

At this stage, we use the result of the integration of local and global descriptors to recognize the object and estimate its position. In this integration, all of the VFH global descriptor and FPFH local descriptor performs a particular task. VFH is used to classify the object based on the global features. Based on the result of classification, FPFH is used local features to estimate the position of the object. As shown in Figure 7, once descriptors are completed for the source cloud in the previous step, the single histogram of VFH is independently compared with all histograms in our database. More specifically, a single descriptor of VFH is compared using the Euclidean distance and the best Number Nearest Neighbor ( $N$  NNs) are retrieved. The  $N$  NNs denote the  $N$ -best similar views in the model database matching to the description obtained by of the target global feature and the metric used to compare the histograms. The histogram matching is done by “Fast Library for Approximate Nearest Neighbors” (FLANN) and KD-Tree.

Therefore, the partial view candidates obtained by matching can be aligned using a Sample Consensus-based Initial Alignment (SAC-IA) algorithm [13]. It is used for finding correspondences between two sets of local features. This alignment is further refined using a surface registration method such as Iterative Closest Point (ICP) [22] algorithm. As postprocessing step, ICP algorithm is used to find a precise alignment from the source cloud to the target cloud. Figures 10 and 11 show an example of the recognition process using our proposed system.



**Figure 7.** The integration of the local and global descriptors.

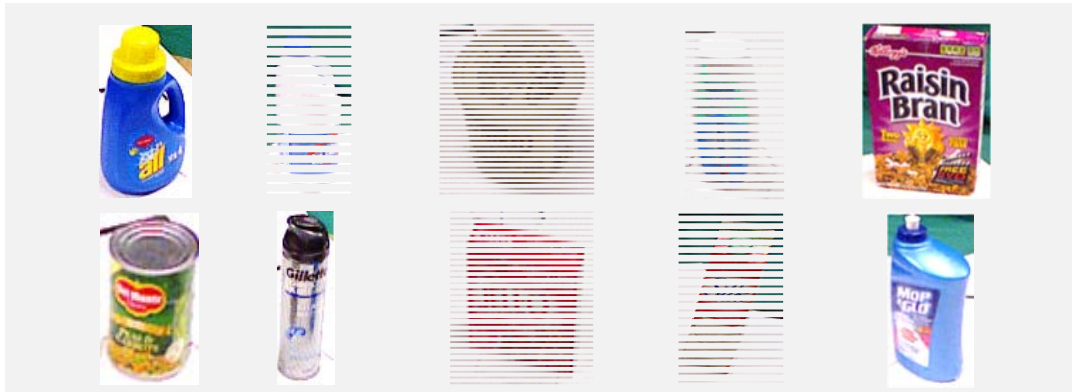
## 5. The Experimental Results

### 5.1. Dataset

We evaluated our system on two different standard data sets: the RGB-D dataset [20] and the Solutions in Perception Challenge dataset [23]. The RGB-D dataset contains 300 objects from 51 various categories, in 207621 segmented clouds. It does not require pre-processing to cloud data. Figure 8 shows a one-point cloud of one object from each of the ten categories that we tested. Whereas, The Solutions in Perception Challenge dataset provided by Willow Garage that collected as part of the solutions for perceptions challenge at ICRA 2011. It uses a set of 35 common household objects. It requires a pre-processing stage of the cloud data. Figure 9 shows a one-point cloud of one object from each of the ten categories that we tested from this dataset.



**Figure 8.** One view of each of the ten categories used in the experiments from RGB-D dataset. Top row, left to right: apple, banana, coffee\_mug, cap, keyboard. Bottom row, left to right: kleenex, orange, plate, shampoo, soda\_can.



**Figure 9.** One view of from each of the ten categories used in the experiments from Willow Garage dataset. Top row, left to right: ail, clorox, coffee, crest, foodbox. Bottom row, left to right: food\_cans, gillette, jello, juice, mop\_gio.

We use 100 clouds of RGB-D dataset as training data to build feature models and then evaluate our recognition system using the 100 clouds as test data. Whereas, the training set of Willow Garage dataset has total 97 clouds and then evaluate our recognition system using the 43 clouds as test data.

## 5.2. Setup

To perform a fair comparison of descriptors, each descriptor that uses normal is evaluated with the normal estimation radius search equal to 3 cm. The input cloud downsampling was obtained using a voxelgrid filter with leaf size equal to 1 cm. The SIFT3D key point detector was also used with minimum local search equal to 1.5 cm. Each local descriptors are evaluated with the size of the neighborhood equal to 0.1 cm.

## 5.3. Results

All tests were performed on a DELL-Inspiron-N5110 laptop with Intel Core i3, 2.10 GHz processor, 3GB RAM, and Windows 7 Ultimate 64-bit as an operating system. We have implemented the entire 3D object recognition system in Microsoft Visual Studio 2010 using PCL version 1.7.2.

Our experiments implemented in two steps, the first step, is used to perform experiments on three different types of global descriptors. The second step is used to perform experiments in the integration of VFH global descriptor as it showed the best performance with five various types of local descriptors.

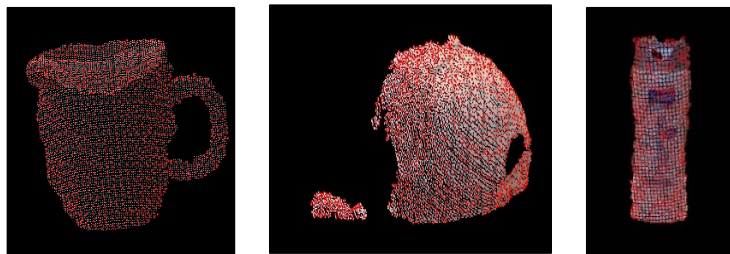
Table 2 contains the experiments' results of the recognize of an object with global descriptors. We can see from the average value of each column in Table 2, the VFH global descriptor has the best performance to recognize the object.

**Table 2.** The accuracy of the global descriptors for RGB-D dataset.

Objects	ESF	VFH	CVFH
Apple	0.40	0.90	0.80
Banana	0.70	1.00	0.60
Coffee_mug	0.50	1.00	0.70
Cap	0.60	1.00	0.60
Keyboard	1.00	1.00	1.00
Kleenex	0.20	0.90	0.50
Orange	0.55	1.00	0.50
Plate	1.00	1.00	1.00
Shampoo	0.90	1.00	0.70
Soda_can	0.40	0.90	1.00
Average	0.62	<b>0.97</b>	0.74

**Table 3.** The processing time (seconds) for RGB-D dataset.

Integratation descriptors	Time of training	Time of recognition
SHOT & VFH	29	8.31
CSHOT & VFH	34	26.65
PFH & VFH	360	5.76
PFHRGB & VFH	540	11.41
FPFH & VFH	133	<b>3.17</b>



**Figure 10.** A sample object recognition from our system from RGB-D dataset.

Table 3 presents training database time and testing object recognition time. In terms of time, the best descriptors that appear the fastest time to both recognize and estimate the position of the object. We can see in Table 3 the value of each row, The SHOT descriptor showed less time in the training data,

but when the test took a relatively long time compared to other descriptors. Similarly, CSHOT performance resembled the performance of the SHOT, but it took a longer time for adoption also on the color information. The PFH and PFHRGB descriptors are similar in performance, we note that the PFHRGB takes almost double the time it takes the PFH, this is because of the color information.

As shown in Tables 2 and 3, the integration of FPFH and VFH descriptors showed a good balance between recognition results and time complexity. Table 4 presents object recognition accuracy by FPFH and VFH descriptors with Willow Garage dataset.

**Table 4.** The performance accuracy of integration of the VFH and FPFH descriptors for Willow Garage dataset.

Objects	FPFH&VFH
Ail	0.50
Clorox	0.71
Coffee	1.00
Crest	0.40
FoodBox	1.00
FoodCans	1.00
Gillette	0.90
Jello	1.00
Juice	0.70
MopGio	0.80
Average	<b>89.64%</b>



**Figure 11.** A sample object recognition from our system from Willow Garage dataset.

## 6. Conclusion

3D object recognition from point clouds with local and global features is currently one of the most active research topic in the computer vision field. On that basis, we focused in this paper on state-of-the-art 3D descriptors that aimed at recognizing the object, explaining how they work, including perform experiments to illustrate and evaluate their relative performance.

Based on the experimental results, the VFH has the best performance to recognize the object among other global descriptors. In terms of time, FPFH has the best results. Therefore, we proposed a 3D object recognition system based on FPFH local descriptor and VFH global descriptor using PCL. The system can recognize non-rigid objects. The results of the proposed system showed that the system performed well with the objects that have been trained.

In the future, we intend to study the possibility of improving performance using new 3D descriptors and integrate further tools, such as depth information. We also seek to develop the system so that it can handle multiple objects and occlusions.

## 7. References

- [1] Guo Y., Bennamoun M., Sohel F., Lu M., and Wan J. "3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, Nov 2014.
- [2] Akagündüz E., and Ulusoy I. "3D object recognition from range images using transform invariant object representation," *Electron. Lett.*, vol. 46, no. 22, pp. 1499–1500, 2010.

- [3] Bayramoglu N., and Alatan A. "Shape index SIFT: Range image recognition using local features," in Proc. 20th Int. Conf. Pattern Recognition., pp. 352–355, 2010.
- [4] Rusu R. B., Bradski G., Thibaux R., and Hsu J. "Fast 3D recognition and pose using the viewpoint feature histogram," IEEE/RSJ Int. Conf. in 2010.
- [5] Aldoma A., Tombari F., Stefano L. Di., and Vincze M. "A global hypotheses verification method for 3D object recognition," *Compute. Vis.–ECCV 2012*. Springer Berlin Heidelberg, 2012.
- [6] Rusu R. B. and Cousins S. "3D is here: Point Cloud Library (PCL)," in IEEE Int. Conf. on Robotics and Automation (ICRA), Shanghai, China, May 2011.
- [7] Alexandre L. A. "3D descriptors for object and category recognition: a comparative evaluation," Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ Int. Conf. on IROS, Vilamoura, Portugal. Vol. 1. No. 2. 2012.
- [8] 3D object recognition (descriptors) [http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI\\_tutorial\\_4:\\_3D\\_object\\_recognition\\_%28descriptors%29](http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_4:_3D_object_recognition_%28descriptors%29) (Last accessed on 08/12/2014).
- [9] Aldoma A., Marton Z. C., Tombari F., Wohlkinger W., Potthast C., Zeisl B., and Vincze M. "Point Cloud Library," IEEE Robotics & Automation Magazine 1070.9932/12, 2012 .
- [10] Tombari F., Salti S., and Stefano L. "Unique signatures of histograms for local surface description," in Eur. Conf. on Compute. Vis. 2010.
- [11] Tombari, F., Salti, S., and Di Stefano, L. "A combined texture-shape descriptor for enhanced 3D feature matching," in IEEE Int. Conf. on Image Processing, September 2011.
- [12] Rusu R. B., Blodow N., Marton Z. C., and Beetz M. "Aligning point cloud views using persistent feature histograms," in Proc 21st IEEE/RSJ Int. Conf. IROS, Nice, France, Sept. 22–26, 2008.
- [13] Rusu R. B., Blodow N., and Beetz M. "Fast point feature histograms (FPFH) for 3-D registration," in Proc. IEEE Int. Conf. Robotics and Auto-mation (ICRA), Kobe, Japan. 2009.
- [14] Wohlkinger W. and Vincze M. "Ensemble of shape functions for 3D object classification," in Robotics and Biomimetics (ROBIO), 2011 IEEE Int. Conf. on, December 2011.
- [15] Osada R., Funkhouser T. A., Chazelle B., and Dobkin D. P. "Matching 3D models with shape distributions," in Int. Conf. on Shape Modeling and Applications, Genoa, Italy, 2001.
- [16] Aldoma A., Vincze M., Blodow N., Gossow D., Gedikli S., Rusu R. B., and Bradski G. "CAD-model recognition and 6DOF pose estimation using 3D cues," IEEE Int. Conf. on. IEEE, 2011.
- [17] Filipe S., and Alexandre L. A. "A Comparative Evaluation of 3D Keypoint Detectors in a RGB-D Object Dataset," 9th Int. Conf. on Compute. Vis. Theory and Applications, Lisbon, Portugal. 2014.
- [18] Tang J., Miller S., Singh A., and Abbeel P. "A textured object recognition pipeline for color and depth image data," Robotics and Automation (ICRA), 2012 IEEE Int. Conf. on. IEEE, 2012.
- [19] Kasaei S. H., Lopes L., Tomé A. M., Chauhan A., Mokhtari V. "A. An Instance-Based Approach to 3D Object Recognition in Open-Ended Robotic Domains," *Transversal emRobóticaInteligente*. 2011.
- [20] Lai K., Bo L., Ren X., and Fox D. "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset," IEEE Int. Conf. on Robotics and Automation. 2011.
- [21] Xie Z., Uag J., Singh A., and Abbeel P. "Dense Feature Models for Object Detection using RGB-D Data," IEEE Int. Conf. on Robotics and Automation, 2013.
- [22] Besl P. J., and McKay N. D. "A Method for Registration of 3-D Shapes," IEEE Trans. 1992.
- [23] The solution in perception challenge. <http://www.acin.tuwien.ac.at/?id=312>. (Last accessed on 28/2/2015).