A Performance Evaluation of Surface Normals-based Descriptors for Recognition of Objects Using CAD-Models

C. M. Mateo[1], P. Gil[2] and F. Torres[2]

[1]*University Institute for Computing Research, University of Alicante, San Vicente del Raspeig, Spain*

[2]*Department of Physics, Systems Engineering and Signal Theory, University of Alicante, San Vicente del Raspeig, Spain*

Keywords: 3D Object Recognition, 3D Surface Descriptors, Surface Normal, Geometric Modelling.

Abstract: This paper describes a study and analysis of surface normal-base descriptors for 3D object recognition. Specifically, we evaluate the behaviour of descriptors in the recognition process using virtual models of objects created from CAD software. Later, we test them in real scenes using synthetic objects created with a 3D printer from the virtual models. In both cases, the same virtual models are used on the matching process to find similarity. The difference between both experiments is in the type of views used in the tests. Our analysis evaluates three subjects: the effectiveness of 3D descriptors depending on the viewpoint of camera, the geometry complexity of the model and the runtime used to do the recognition process and the success rate to recognize a view of object among the models saved in the database.

1 INTRODUCTION

The 3D object recognition process has had important advances in the last years. In recent works, many approaches use range sensors to obtain depth of objects present in a scene. The depth information has permitted to change the techniques and algorithms for extracting features from image. In addition, this one has been used to design and create new descriptors for identification objects from scene captured by range sensors (Rusu, 2009) and (Lai, 2013). LIDARSs, Time of Flight cameras (ToF) or RGBD sensors, such as Kinect or Asus Xtion PRO Live, provide depth and allow us to recover the 3D structure of scene from a single image. The choice of the kind of sensor is depending on the context and lighting conditions (indoors, outdoors) and type of specific application (guided/navigation of robots or vehicles, people detection, human-machine interaction, object recognition and reconstruction, etc.). Furthermore, the recognition methodology applied to retrieve the 3D object shape is different depends on whether the object is rigid or non-rigid. A variety of methods for detection of rigid and non-rigid objects were presented in (Wohlkinger et al., 2012) and (Lian et al., 2013), respectively.

In this work, rigid object recognition is done. But rigid object recognition can be based on visual features information such as bounding, skeleton, silhouette, colour, texture, moments, etc. or geometric features such as vectors normal, voxels, etc. obtained from depth information captured from a range sensor. Examples of descriptors for rigid objects based on geometric features, are: PFH (*Point Feature Histogram)* and FPFH (*Fast Point Feature Histogram*) (Rusu, 2009); VFH (*Viewpoint Feature Histogram*) (Rusu et al., 2010); CVFH (*Clustered Viewpoint Feature Histogram*) (Aldoma et al., 2011); and SHOT (*Signature of Histograms of Otientations*) (Tombari et al., 2010). All of them describe the geometry of an object using normal vectors to its surface which is represented by a point clouds. Other descriptors such as ESF (*Ensemble of shape Functions*) (Wohlkinger and Vincze, 2011a) and SVDS (*shape Distribution on Voxel Surfaces*) (Wohlkinger and Vincze, 2011b); GRSD (*Global Radius based Surface Descriptors*) (Marton et al., 2011) are based on voxels to represent the object surface. SGURF (*Semi-Global Unique Reference Frames*) and OUR-CVFH (*Oriented, Unique and Repeatable CVFH*) (Aldoma et al., 2012b) are also other noteworthy descriptors because they have the advantage to the ambiguity over the camera roll angle. SGURF is computed from a single viewpoint of the object surface and OUR-CVFH is based on a mix between SGURF and CVFH. CVFH is briefly discussed below.

In this paper, 3D rigid object recognition based on object category recognition is done. Also, we have

Mateo C., Gil P. and Torres F..
A Performance Evaluation of Surface Normals-based Descriptors for Recognition of Objects Using CAD-Models.
DOI: 10.5220/0005056504280435
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 428-435
ISBN: 978-989-758-040-6

introduced some novelty into the performance shown in (Wohlkinger et al., 2012) and (Alexandre, 2012). We have created views from a virtual camera which captures information of virtual models with different viewpoints. Afterwards, we have created the 3D rigid objects from CAD models using 3D printer to test if the behavioural changes of the descriptors are significant. Thereby, the errors in the recognition process can be better controlled. Thus, both descriptors, model and object, are computed from known perfect geometrical figures. Therefore, the recognition errors only depend on the geometry of the isolated object in the scene and the precision of descriptor for modelling and identifying these objects. It is important emphasize that evaluated descriptors cannot be used if the scene was not previously segmented and the objects are localized therein.

The rest of this paper is structured as follows. 3D descriptors based on geometric information are commented in Section 2. In Section 3, we present the similarity measures proposed for associating objects to models. Experimental results of the descriptors evaluation is shown in Section 4 and 5. Finally, section 6, contains the conclusions.

2 3D DESCRIPTORS

In this paper, we work with isolated rigid objects with uncluttered backgrounds in indoor scenes. Hence, our appearance model is based on a set of different feature descriptors. In particular, five descriptors are used in the experimentation. For each descriptor type, we use the same training framework. That is the same objects as dataset or test data. The training framework is detailed later (Section 4). The descriptors are always computed over a mesh consists of a point cloud. The descriptors only include geometric information based on the surface shape but they do not include colour or other type of visual features information. The idea is to evaluate 3D objects recognition methods based on 3D descriptors without using additional appearance information such as colour and texture from scene image, information position/orientation from geolocation and odometry techniques obtained. The absence of colour and texture provides generality for working with unknown objects and simplifies the runtime in the recognition task. Frequently, in the industrial environments are used objects and pieces without this kind of information. Those are made of metal or plastic material with homogeneous colour and they can only be differenced by means of geometry and surface features.

The five feature descriptors based on surface normal vectors: PFH, FPFH, SHOT, VFH and CVFH, were chosen because they retrieve enough geometrical information of shape. This information will give us the ability to make further analysis in industrial pieces. In the literature, descriptors are grouped as local and global recognition pipeline. The main difference among these groups is the size of signature and the number of signatures to describe the surface. In the first, descriptor is represented by a signature for each point of surface, but, in the second, it saves all viewpoint information using one signature for whole surface. A brief description:

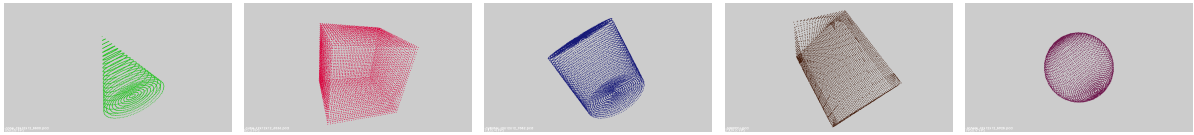
PFH, It is a set of signatures from several local neighbourhoods. For each point is computed a 3-tuple, $\langle \alpha, \phi, \theta \rangle$ of angles which represent the relation among normals in their neighbourhood, according to Darboux frame. Then in order to, compute each final signature, the method adds the relations among all points within neighbourhood in the surface. Therefore the complexity computational is $O\left(nk^2\right)$. The signature dimensionality is 125.

FPFH, This is based on the same idea that *PFH*, it uses a Darboux frame to make relations among pair of points within a neighbourhood with radio r for computing each local surface signature. This descriptor generates a linear complexity in the number of neighbours, $O(nk)$. This approximation changes the relations among a point and its neighbours located with a distance smaller than r, adding a specific weight according to the distance between point and every neighbour. The signature dimensionality is 33.

SHOT, In this descriptor a partitioned spherical grid is used as local reference frame. For each volume of the partitioned grid, a signature of the amount of $\cos\theta_i$ between the normal at each point of surface and the normal at the query feature point is computed. A normalization of descriptor is required to provide it robustness towards point density variations. The signature dimensionality is 352.

VFH, It is based on FPFH. Each signature consists of a histogram with two components; one has the angles $\langle \alpha, \phi, \theta \rangle$ which is calculated as the angular relation between a point's normal and the normal of the point cloud's centroid, and other represent the angles between the vector determined by the surface centroid and viewpoint. This descriptor has complexity of $O(n)$. The signature dimensionality is 308.

CVFH, This descriptor is an extension to *VFH*. The basic idea is to identify an object from splitting it



(a) Cone (b) Cube (c) Cylinder (d) Prism (e) Sphere

Figure 1: Primive shapes of the models.

in a set of smooth and continuous regions or clusters. The edges, ridges and other discontinuities in the surface are not considered because these parts are more affected by the noise. Thereby, for each of these clusters is computed its *VFH* descriptor. *CVFH* describes a surface as a histogram in which each histogram item represents the centroid to surface and the average of the normals among all points of surface. Again, the dimensionality is 308.

Other descriptors such as Radius-based (*RSD* and *GRSD*) or voxels-based (*SVDS* and *ESF*) are not studied here. This decision was taken because the results shown in (Aldoma et al., 2012a) and (Alexandre, 2012) that Normal-based descriptors are best with household object as proving the accumulated recognition rate, ROC curve for recognition and Recall-vs-(1-Precision).

3 SIMILARITY MEASURES

Similarity measures are used to associate the CAD-model and the object view. The similarity measures are defined like distance metrics. Four type of distance metrics, $d_s = \{d_{L1}, d_{L2}, d_{\chi^2}, d_H\}$ are used to compare the CAD-model, C_j, which represents a object category with the object view in the scene. The definitions for the four distances are:

$$d_{L1}(p,q) = \sum_{i=1}^{n} p_i - q_i \tag{1}$$

$$d_{L2}(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2}$$

$$d_{\chi^2}(p,q) = \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i + q_i} \tag{3}$$

$$d_H(p,q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2} \tag{4}$$

where d_{L1} defines the Manhattan distance, d_{L2} is Euclidean distance, d_{χ^2} defines Chi-squared distance and d_H is Hellinger distance. And n is point dimensions, being p and q two arbitrary points.

Each CAD-model, C_j is defined by a set of views $C_j = \{c_{j1}, c_{j2} \ldots, c_{jr}\}$ where r is the number of viewpoints from where the CAD-model is observed with a virtual camera. Furthermore, each view is represented by a set of descriptors defined as following, $c_{jl} = \left\{m_1^{jl}, m_2^{jl}, m_3^{jl}, m_4^{jl}, m_5^{jl}\right\}$ where l represents the view identifier and j the object class defined in the CAD-model. This set represents a hybrid descriptor composed of five components. A component for each type of descriptor: *PFH*, *FPHF*, *SHOT*, *VFH* and *CVFH*. Similarly, for each object, O_i is defined by a set of views $O_i = \{o_{i1}, o_{i2}, \ldots, o_{in}\}$ where n is the number of viewpoints from where the object in scene is captured using a virtual or real camera. As well, each view is represented by a set of descriptors defined as following, $o_{ik} = \left\{v_1^{ik}, v_2^{ik}, v_3^{ik}, v_4^{ik}, v_5^{ik}\right\}$ where k represents the view identifier, and i is the object identifier.

Then, the difference between each component of the CAD-model descriptor and object descriptor, is calculated according to equations (1), (2), (3) and (4).

The similarity, d_c, between object category, C_j in the database and the object in scene, is computed by using the minimum distance for each type of descriptor, following equation (5). The comparison is done for all models saved in the database.

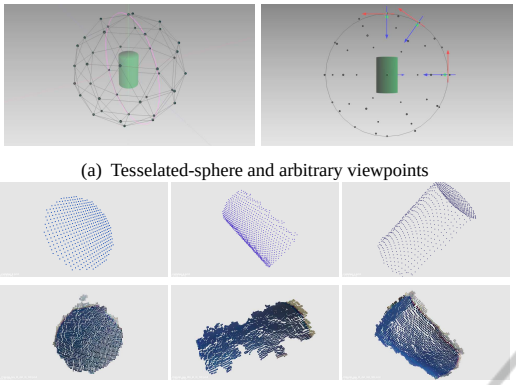
$$d_c(O_i, C_j) = \min_{o^{ik} \in O_i \wedge c^{jl} \in C_j} \left\{ d\left(o^{ik}, c^{jl}\right) \right\} \tag{5}$$

$$d\left(o^{ik}, c^{jl}\right) = \sqrt{d_s(o^{ik}, c^{jl})^2 + d_s(c^{jl}, o^{ik})^2} \tag{6}$$

where s represents the kind of distance defined in equation (1), (2), (3) and (4).

4 EXPERIMENTS

Test data were created to analyse the 3D descriptors behaviour. They were created like a dataset of the 5 basic shapes which are used like models of objects. They are a sphere, cube, cone, cylinder and triangular prism (Figure 1). These models represent different surfaces without colour, texture or another characteristic different to geometry. Each CAD-model was



(a) Tesselated-sphere and arbitrary viewpoints

(b) Top, side and another.

Figure 2: (a) Camera poses to obtain views. (b) Virtual and real objects views from three arbitrary poses, respectively.

created as a point cloud from CAD software. Each CAD-model represents an object category in order to recognize. They are represented by a point cloud with variable number of points, with regards to the view and the kind of shape.

The correspondence process between model and object must be consistent. For this reason, in this paper, we have evaluated this process using CAD-models. In addition, we did not use keypoints computed from the surface and so the noise due to inaccuracy in its location is almost eliminated. Therefore, factors like the repeatable of keypoints with respect to viewpoint variations cannot be produced. We have used all points in the surface to analyse and evaluate the descriptors behaviour, thoroughly. If we had only evaluated the descriptors with a number of points chosen from surface, i.e. keypoints, the analysis had been limited to effectiveness of those. The keypoints must be chosen to avoid redundant or sparse information (keypoints close or too far themselves, respectively). Generally, the descriptors based on keypoints are efficient but they are little descriptive and they are not robust to noise. Other descriptors, such as local/regional or global descriptors are more suitable to noise. Moreover, they are useful to handle partial/complete surface information and so they are more descriptive on objects with poor geometric structure. Therefore, they are more suitable to categorize objects in a recognition process, as can be seen here.

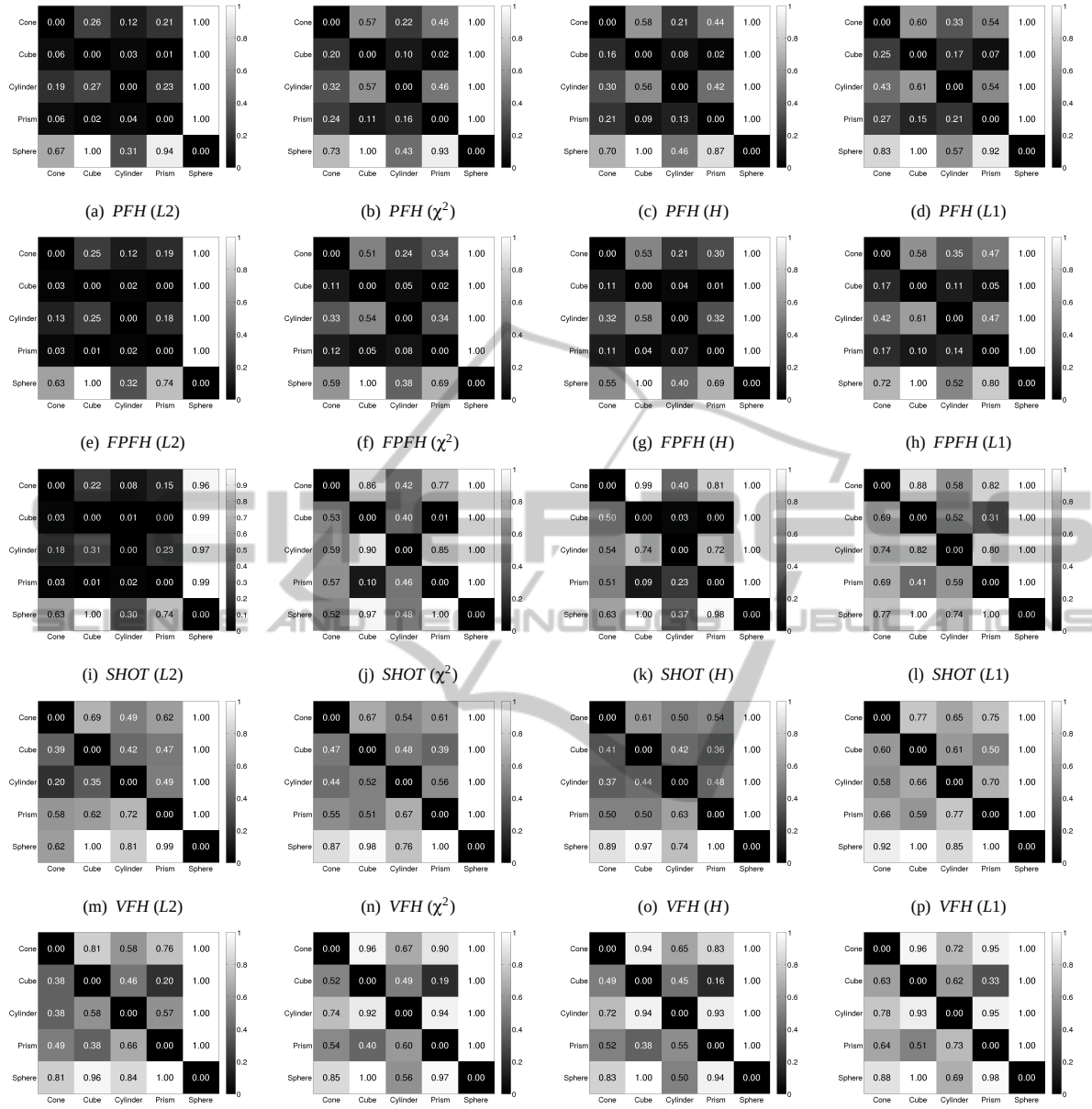
In the experiments, geometric transformations are applied to the point cloud of CAD-models shown in Figure 1. Geometric transformations simulate viewpoint of the objects in scene of real world. Geometric transformations applied were rotations, translations and scale changes from different camera poses (Figure 2). The recognition process consists of a matching process among CAD models and objects in order to associate and identify the object category. The object category is given by the object greatest similarity between the object and the geometric shape of a model (Figure 3, Figure 4 and Figure 5), applying Equation 5.

In order to evaluate the behaviour descriptors and find which works best in recognition process, we have planned two type of experiments. Firstly, virtual objects are created from CAD-models selecting views to build the test database (Figure 3). Thus, at least, we guarantee that all views created for the test database are equals to one view of a CAD-model. Secondly, virtual objects are created from CAD-models applying one or more transformation on those (Figure 4). These transformations are chosen to provide different views to any view used within a model so we ensure a total difference between test database and models. In this case, we have worked with 42 and 38 different views of the test and model database, respectively.

Figure 3 shows a comparison in which the matching process is done combining all descriptors with all distances for virtual object views without transformations. This comparison allows us to determine the capacity of similarity measures for classification of object views in categories according to a CAD-model. The obtained results report better recognition when the matching process is done using $L1$ distances and the worst results are generated by $L2$ distance, in both case is independent from the used 3D descriptor. In addition, $L2$ distance causes confusion in the recognition as distance matrices of *PFH*, *FPFH* and *SHOT* demonstrate.χ^2 and H provide similar results although H is slightly better.

Figure 4 shows an interesting additional experiment. It consists in reporting recognition results with regard to the transformation level. The difficulty in the matching process is increased due to the loss of similarity among the virtual object views with transformation and the models. In this case, both distance matrices, *VFH* and *SHOT*, report about a growth of confusion level in the recognition regardless of distance metric. Furthermore, both *PFH* and *FPFH* are not practically changed their behaviour. Summarizing, *CVFH* is the most stable descriptor although the chosen distance metric is different or the object views are not exactly equal to any model views.

Finally, we have tried out the behaviour of the two best descriptors using the two best similarity measures when the recognition process is realized from real physical objects. In this case, the views for the test database are obtained by means of acquisition process from Kinect. In this last experiment, CAD-models are used to create 5 real physical objects using a 3D printer. They were created using PLA (PLA:



(a) *PFH* (*L*2) (b) *PFH* (χ^2) (c) *PFH* (*H*) (d) *PFH* (*L*1)

(e) *FPFH* (*L*2) (f) *FPFH* (χ^2) (g) *FPFH* (*H*) (h) *FPFH* (*L*1)

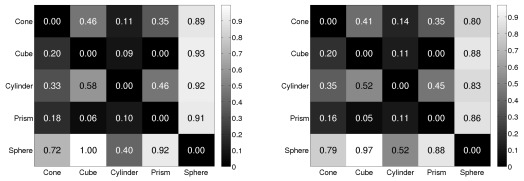
(i) *SHOT* (*L*2) (j) *SHOT* (χ^2) (k) *SHOT* (*H*) (l) *SHOT* (*L*1)

(m) *VFH* (*L*2) (n) *VFH* (χ^2) (o) *VFH* (*H*) (p) *VFH* (*L*1)

(q) *CVFH* (*L*2) (r) *CVFH* (χ^2) (s) *CVFH* (*H*) (t) *CVFH* (*L*1)

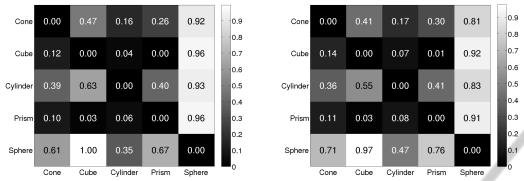
Figure 3: Distance matrix when model set is compared with itself (Model vs Model).

PolyLactic Acid or PolyLActide) filament of 3mm diameter. The print allowed us a precisely controlling of the size, exact shape and the building material that objects would have in the scene. This is done because we would not have an appropriated error handling, if household objects similar to (Rusu, 2009) or (Alexandre, 2012) had been used in our experiments. Perhaps, in those cases, the errors in the recognition process were influenced by the properties of building material, the capture and digitalized process when the shapes are not exactly like the CAD-model, etc. For this reason, we have built our own objects for the test database. After we have captured from Kinect these real physical objects using different pose cameras in the scene. In particular, the test data set has a total of 32 camera views for each object. These viewpoints represent rotations and translations. The object has been rotated from 4 different angles $(0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2})$rad in two different axis (in relation of the main axis and minor axis of the object). In addition, the object has been translated to 4 different positions which represent (origin, near, left and right). This way the scale



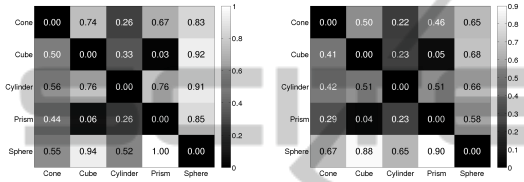
(a) *PFH* (χ^2)

(b) *PFH* (*L*1)



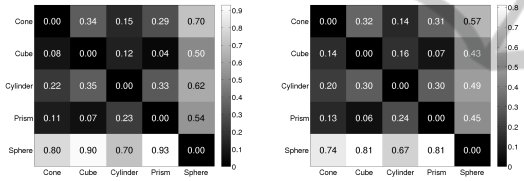
(c) *FPFH* (χ^2)

(d) *FPFH* (*L*1)



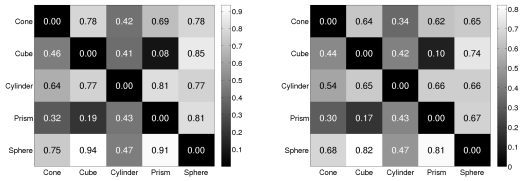
(e) *SHOT* (χ^2)

(f) *SHOT* (*L*1)



(g) *VFH* (χ^2)

(h) *VFH* (*L*1)



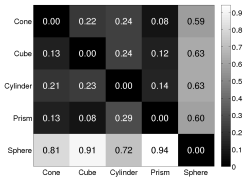
(i) *CVFH* (χ^2)

(j) *CVFH* (*L*1)

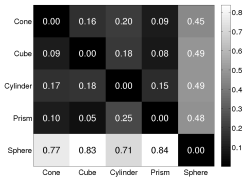
Figure 4: Distance matrix when model set is compared with test set (Model vs Test).

changes have also been considered. The result can be seen in Figure 5 which shows the matching process between all objects and all CAD-models.

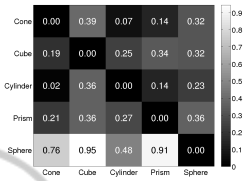
As the above Figures 4 clearly shown, *CVFH* is the most effective to recognize virtual objects. Therefore, it turns out a good choice to apply it to recognize real physical objects using similar views to those were registered for the virtual objects as is shown in Figure 5. A comparison of Figures 4(i)- 4(j) and Figures 5(c)- 5(d) demonstrate that the presence of variations, such as present noise, lacking of points to define the surface when the view is captured from camera or loosing of smoothing surface due to noise points in the acquisition process, have worsened the matching



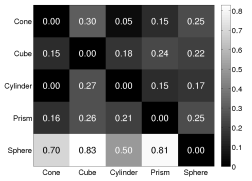
(a) VFH (χ^2)



(b) VFH (*L*1)



(c) CVFH (χ^2)



(d) CVFH (*L*1)

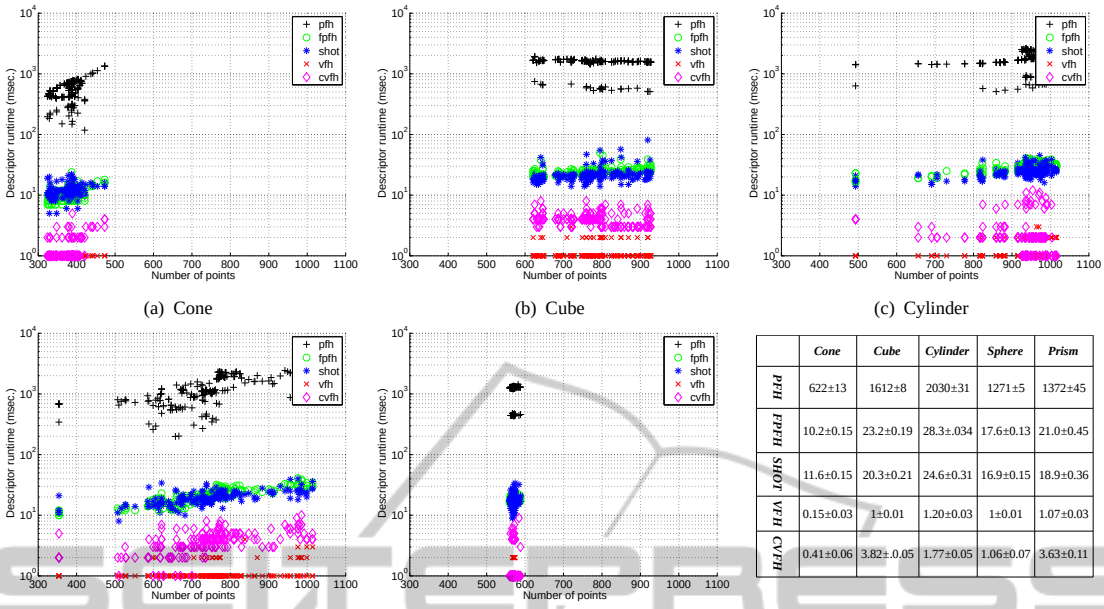
Figure 5: Distance matrix for matching process among models and real scenes.

process. Consequently, the distance between a view and false model are closer to zero. This fact is clearly observed between cylinder and cone.

5 ANALYSIS AND EVALUATION OF TIME AND ACCURACY

The recognition process behaviour have been evaluated with regards to the relation between runtime and accuracy. A complete set of experiments were designed. Summarizing, the recognition process consisted of three steps: a) Building database: Calculation of descriptors for each view in each model saved in the database. b) Calculation of descriptors for real and virtual (test) views. c) Matching between test views by means of computing difference among all models views saved in the database and arbitrary test view.

The runtime of steps a) and b) on the recognition process is changing and it depends on amount of points in the view, the number of views per model, the number of models and the descriptor characteristics. Thus, we have to measure the runtime cost depending on detail level of its representation in each point cloud. Figure 6 shows the runtime for each descriptor depending on the shape. Each graph represents the runtime of all descriptors for each shape (for each shape were used 162 views with different amount of points). On the one hand, as observed, the runtime dependency with shape complexity is least-significant than computational complexity of feature descriptor. It is because all shapes keep the following relation: *PFH* >> *FPFH* >> *SHOT* >> *CVFH* >> *VFH*. Although, the shape complexity affects to stability of local feature descriptors runtime (Figure 6(f)). *VFH*



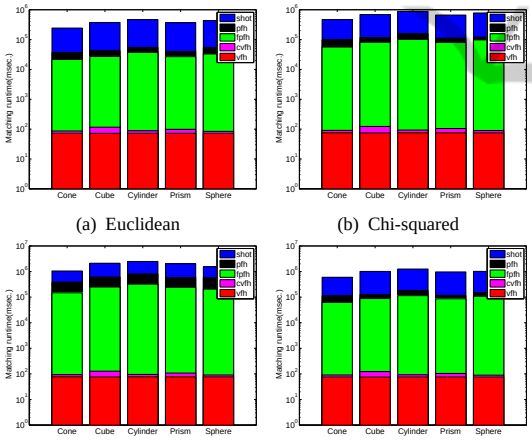
(a) Cone (b) Cube (c) Cylinder

(d) Prism (e) Sphere

	Cone	*Cube*	*Cylinder*	*Sphere*	*Prism*
PFH	622±13	1612±8	2030±31	1271±5	1372±45
FPFH	10.2±0.15	23.2±0.19	28.3±.034	17.6±0.13	21.0±0.45
SHOT	11.6±0.15	20.3±0.21	24.6±0.31	16.9±0.15	18.9±0.36
VFH	0.15±0.03	1±0.01	1.20±0.03	1±0.01	1.07±0.03
CVFH	0.41±0.06	3.82±.0.05	1.77±0.05	1.06±0.07	3.63±0.11

(f) Mean and standard errors

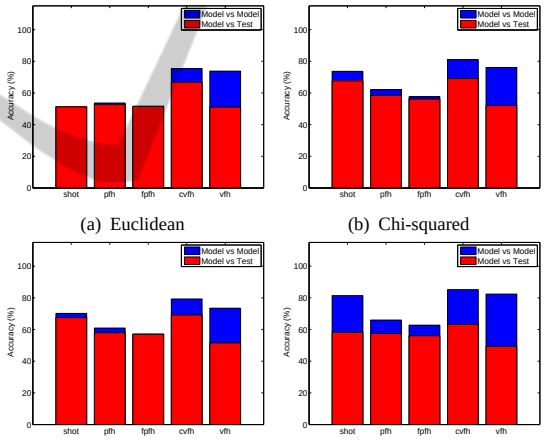
Figure 6: Descriptor runtime depending on the shape.



(a) Euclidean (b) Chi-squared

(c) Hellinger (d) Manhattan

Figure 7: Matching runtime for each descriptor depending on the shape.



(a) Euclidean (b) Chi-squared

(c) Hellinger (d) Manhattan

Figure 8: Accuracy rates for descriptors depending on metric used in matching process.

and *CVFH* are the fastest in this comparison.

On the other hand, a study of the balance between runtime and accuracy is realized in step c). Firstly, Figure 7 shows the mean runtime in matching process between a test view and models database. Again, the set of global descriptors (*VFH* and *CVFH*) is faster than others (10^3 times), independently for the high dimensionality of its signatures. Secondly, Figure 8 shows the difference between accuracy when the matching process is made using models such as test views and when it is made using test views. In addition, accuracy is less using local descriptors than global descriptors. Although *CVFH* has the best accuracy rate, another important issue is the metric selection. In terms of runtime, this selection is not outstanding (Figure 7), but it is important in terms of accuracy (Figure 8). In the experiments, model vs model represented in Figure 3, a 20% increase of accuracy rate is obtained. When $L1$ is used as observed in Figure 8(a) - 8(d). Nevertheless, the best result is obtained using χ^2 in th experiment, model vs test represented in Figure 4. In this case, a 5% increase of accuracy is achieved.

6 CONCLUSIONS

This paper discusses the effectiveness of using 3D descriptors based on normals to surfaces in order to recognize geometric objects. 3D descriptors were used for real physical and virtual objects recognition by means of matching with virtual geometric models. A total of 6028 tests have been done. Where 3800 tests (4 different distances, 5 descriptors, 5 shapes and 38 views per shape) are from the model-vs-model experiment, 2100 tests (2 different distances, 5 descriptors, 5 shapes and 42 views per shape) are from the model-vs-test experiment and 128 tests (2 different distances, 2 descriptors, one shape and 32 views) are from the model-vs-real-physical-object experiment. *SHOT* and *FPFH* are run in CPU-based parallel implementation. The computer specification is Intel Core i7-4770k processor, equipped with 16GB of system memory and GPU is Nvidia GeForce 770GTX. The effectiveness of recognition process is evaluated by measuring the runtime and the precision to achieve success rate of the recognition process. Those are depending on the type of descriptor, resolution of the point cloud which represents each object, and the level of accuracy required for the recognition.

ACKNOWLEDGEMENTS

The research leading to these result has received funding from the Spanish Government and European FEDER funds (DPI2012-32390) and the Valencia Regional Government (PROMETEO/2013/085).

REFERENCES

Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S., and Vincze, M. (2012a). Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. In *IEEE Robot. Automat. Mag.*, volume 19, pages 80–91.

Aldoma, A., Tombari, F., Rusu, R. B., and Vincze, M. (2012b). Our-cvfh - oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *DAGM/OAGM Symposium*, pages 113–122.

Aldoma, A., Vincze, M., Blodow, N., Gossow, D., Gedikli, S., Rusu, R. B., and Bradski, G. R. (2011). Cad-model recognition and 6dof pose estimation using 3d cues. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pages 585–592.

Alexandre, L. A. (2012). 3D descriptors for object and category recognition: a comparative evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal.

Lai, K. (2013). *Object Recognition and Semantic Scene Labeling for RGB-D Data*. PhD thesis, University of Washington, Washington, USA.

Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Van Nguyen, H., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., and Vandermeulen, D. (2013). A comparison of methods for non-rigid 3d shape retrieval. *Pattern Recogn.*, 46(1):449–461.

Marton, Z.-C., Pangercic, D., Blodow, N., and Beetz, M. (2011). Combined 2d-3d categorization and classification for multimodal perception systems. *I. J. Robotic Res.*, 30(11):1378–1402.

Rusu, R. B. (2009). *Semantic 3D object maps for everyday manipulation in human living environments*. PhD thesis, Technical University Munich.

Rusu, R. B., Bradski, G. R., Thibaux, R., and Hsu, J. (2010). Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, pages 2155–2162. IEEE.

Tombari, F., Salti, S., and Stefano, L. D. (2010). Unique signatures of histograms for local surface description. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, pages 356–369, Berlin, Heidelberg. Springer-Verlag.

Wohlkinger, W., Aldoma, A., Rusu, R. B., and Vincze, M. (2012). 3dnet: Large-scale object class recognition from cad models. In *ICRA*, pages 5384–5391. IEEE.

Wohlkinger, W. and Vincze, M. (2011a). Ensemble of shape functions for 3d object classification. In *ROBIO*, pages 2987–2992. IEEE.

Wohlkinger, W. and Vincze, M. (2011b). Shape distributions on voxel surfaces for 3d object classification from depth images. In *ICSIPA*, pages 115–120. IEEE.