CHAPTER X

DISTINGUISING BETWEEN SIMILAR OBJECTS BASED ON GEOMETRICAL FEATURES IN 3D PERCEPTION

J. GARCÍA BUENO[1,2], A. MARTÍN CLEMENTE[1,2], M. GONZÁLEZ-FIERRO[1,3], L.MORENO[1] and C. BALAGUER[1]

[1]Robotics Lab, Universidad Carlos III de Madrid; [2]beMee Technology S.L.; [3]Samsamia Technologies S.L.,
{jgbueno, aimartin, mgpalaci, moreno, balaguer}@ing.uc3m.es

In this study, we propose a meta-classifier to identify and distinguish between objects with confusable geometry, similar in appearance, colors and dimensions. The suggested classifier is able to differentiate successfully between 3D point clouds with a high degree of similarity extracting parameters such as 3D features from the surface of the cluster, absolute geometrical boundaries and also color distributions. Objects are until now detected based on its Viewpoint Feature Histograms (VFH) with the downside of being scale independent, making difficult the segmentation of items with simile geometry. Furthermore, this paper discusses several distance metrics to choose which one fits most with VFH descriptors. We present several experiments that validate the model and corroborate the meta-classifier here proposed.

1 Introduction

During the last years, object recognition has turned into a prior research line in robotics. The idea of designing robots capable of detecting and recognizing objects in their vicinity make more natural and prosperous their integration in humanlike scenarios such as offices, kitchens (Rusu, 2010) or living rooms (Bueno, 2012; Liu, 2001). With the aim of detecting and interacting with these objects, robot perception abilities have to be mostly

enhanced. Time-Of-Flight technology and structured-light have become decisive in perception sensorial devices such as Kinect camera or Asus Xtion Pro Live. Those instruments are capable of capturing RGB-D point clouds at high frame rates allowing robots to interpret and analyze with clarity what is coming about (Rusinkiewicz, 2002) and consequently react to these perturbations.

Objects are intended to be detected mostly for grasping. This fact leads to a consequent challenge: items have to be not only successfully detected but also favorably located in order to determine the path to reach and grasp them (Bueno, 2011). This requirement demands two different goals: object pose by means of its 6 DOF and object labeling based on its inherent features such as color, texture, shape, 3D key-points, geometry, center of gravity, etc. For the first purpose, there exist multiple strategies such as 3D recognition based on correspondence grouping using SHOT32 features, ICP-like algorithms (Rusinkiewicz, 2001) or 3D matching (Bueno, 2012; Scovanner, 2007; Knopp, 2010). For the last, 3D features are necessary. Some of the perception features used nowadays are SIFT 3D (Scovanner, 2007), NARF (Bueno, 2012), SURF 3D (Knopp, 2010; Bay, 2006) or FPFH (Rusu, 2009) among others. Without exception, they extract consistent features from a 3D point cloud cluster such as edges, normal variations on the surfaces or intrinsic geometric models such as lines, planes or spheres.

This study is focused on FPFH (Rusu, 2010), a histogram of values that represents with a single histogram the variations and most remarkable variations for a specific surface. FPFH histogram might be used to search similarities among different objects with scale invariability. This feature becomes an issue when databases are composed by scaled versions of similar objects at different scales (this circumstance is quite common for kitchen items such as spoons, mugs, glasses or dishes). Classifiers are always supported by a distance metric in charge of comparing candidates with the observed guest. The selection of a correct metric can affect directly to the recognition rates as discussed in (Martin, 2009).

Subsequent sections of this paper are organized as follows. Section 2 is a formal description of FPFH histogram. Then section 3 presents the proposed meta-classifier algorithm to detect similar histogram. Afterwards in section 4 the experimental results are discussed and later section 5 addresses the highlighted conclusions.

2 Point Feature Histogram

There exist multiple alternatives to represent and interpret surfaces. As a general rule those techniques analyze point cloud curvatures and normals along the surfaces to reduce a high dimensional problem into something more manageable. Most scenes will contain 3D clusters that may represent dissimilar objects or not, making challenging to create a wide representation for 3D point clouds into a single feature list. (Rusu, 2009; Rusu, 2010) proposes a novel point features representation named Point Feature Histogram (PFH) based on the curvature and normal orientation between neighbors in a certain point cloud. Since surface normals and curvature estimations are able to capture precise details of the geometry around a point, most algorithms base their intention focusing on them to deal with false correspondences.

In order to enhance recognition approaches, there exist many different feature representations for a surface. In general it would be ideal to represent each point with an information label that contains the geometry class it belongs to: edge point, spherical surface point, etc.

Following the above, there is a need of finding a multi-dimensional feature space which separates surfaces in different categories. In terms of point-wise analysis, the concept of a dual-ring neighborhood is introduced for any point $p_i \in \wp$ as

$$(\exists) r_1, r_2 \in \Re, r_1 < r_2 \text{ such that} \begin{cases} r_1 \Rightarrow \wp^{k_1} \\ r_2 \Rightarrow \wp^{k_2} \end{cases}, \text{with } 0 < k_1 < k_2$$

where $\wp$ represents the complete set of 3D points. Both radii r_1, r_2 have a specific target. While r_1 represents the surface normal at the query point p_i, obtained from the Principal Component Analysis of the neighborhood patch $\wp^{k_1}$. The radius r_2 delimits the PFH representation itself.

As it has been stated before, the main goal of the PFH formulation is to encode the $\wp^{k_2}$ neighborhood's geometrical properties by generalizing the mean curvature around p_i using a multi-dimensional histogram of values. Assuming that normal values of neighbors of p_i have been already

computed, it is possible to state that having two different points p_i and p_j the relative difference between them can be defined as follows:

$$\overrightarrow{p_{ij}} = p_i - p_j, \overrightarrow{p_{ji}} = p_j - p_i$$

$$if\ \cos^{-1}(\vec{n_i} \cdot \overrightarrow{p_{ji}}) \leq \cos^{-1}(\vec{n_j} \cdot \overrightarrow{p_{ij}}) \rightarrow \begin{cases} p_s = p_i, n_s = n_i \\ p_t = p_j, n_t = n_j \end{cases}$$

$$else \rightarrow \begin{cases} p_s = p_j, n_s = n_j \\ p_t = p_i, n_t = n_i \end{cases}$$

being p_s the source and p_t the target. As the above condition takes place, p_s is chosen such that the angle between its associated normal and the line connecting the two points is minimal. It is then defined a Darboux coordinates frame at one of the points as shown in Fig. 1.

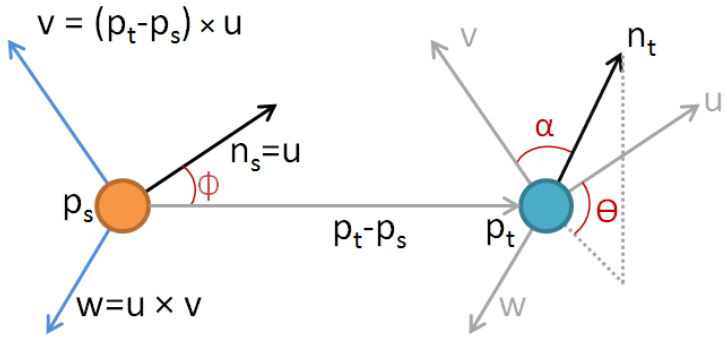

Fig. 1: Darboux coordinates frame at one of the point when computing PFH

where $u = n_s, v = u \times \frac{(p_t - p_s)}{\|p_t - p_s\|_2}$ and $w = u \times v$. With this in mind, the difference between the two normals n_s and n_t is defined with

$$\alpha = v \cdot n_t \quad (1)$$

$$\phi = u \cdot \frac{(p_t - p_s)}{\|p_t - p_s\|_2} \quad (2)$$

$$\theta = tg^{-1}\left(\frac{w \cdot n_t}{u \cdot n_t}\right) \tag{3}$$

The quadruplet $\langle \alpha, \phi, \theta, d \rangle$ reduces the 12 values x, y, z, n_x, n_y, n_z (for both points) to 4 with $d = \|p_t - p_s\|_2$. The influence region diagram is represented in Fig. 2 where p_q represents a query point and its k neighbors in a 3D sphere of radius r_1 are referred as p_{ki} .

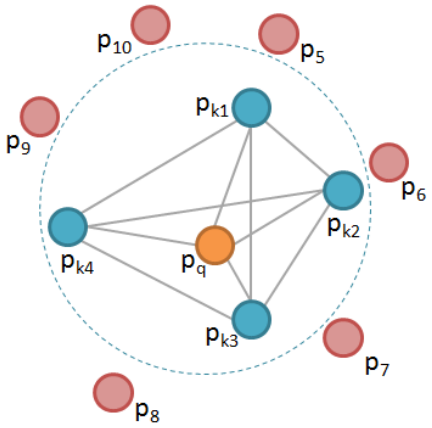

Fig. 2: Influence region diagrams for 3D neighbors in PFH

In order to encode the quadruplet for the whole point cloud, a histogram is created. The quadruplet values are divided on bins forming a histogram including on each bin the number of coincidences for this interval. Seeing that three of the four values are angles it is straightforward to divide the resulting values into equi-spaced bins.

2.1 Fast Point Feature Histogram

A first improvement for PFH consist on taking into account the histogram of the enclosing neighbors for each point and take them into account creating an averaged histogram averaged,

- For each point p_q a set of tuples $\langle \alpha, \phi, \theta \rangle$ are computed between a query point and its neighbors using the equations described above. This is called SPFH (Simplified PFH).

- then, for each point p_q its histogram is recomputed taking into account a weighted value of neighbors SPFH

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{k}\sum_{i=0}^{k}\frac{1}{w_i}\cdot SPFH(p_k) \tag{4}$$

where w_i represents a distance between the query point p_q and a neighbor point p_k in some given metric space (for this study euclidean distance metric will be computed). FPFH includes additional point pairs outside the r_1 radius sphere with a re-weighting strategy that smooth local discontinuities and define a better geometry around the query point.

2.2 View Point Feature Histogram

The second enhancement applied to FPFH concerns to the view point of the object. FPFH is view point dependent because it does not take into account the position of the sensor and therefore it is not suitable for dynamic scenarios such as mobile robot manipulators. To convert the feature estimator view point independent this viewpoint information has to be extracted from the quadruplet somehow. (Rusu, 2009) proposes to maintain viewpoint invariance while retaining invariance to scale computing additional statistics between the viewpoint direction and the normals estimated at each point.

The best way to remove this dependence is straight mixing the viewpoint direction directly into the relative normal angle calculation in the FPFH. A histogram of the angles that the viewpoint direction makes with each normal is computed. This is done from the viewpoint direction at the centroid of the cluster point cloud achieving the feature scale invariant. The histogram VFH then is created as in Fig.3 where there is a viewpoint component and then the FPFH component previously computed.

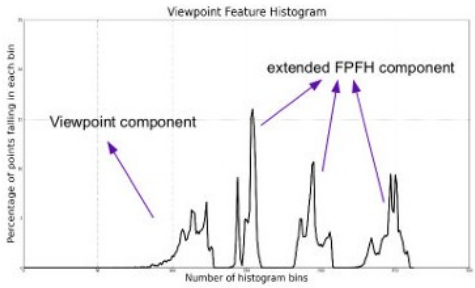

Fig. 3: Example of a VFH histogram with the extended PFH component

3 Proposed algorithm

As it has been stated in the previous section, VFH is scale invariant. In order to classify correctly objects with similar surfaces at different scales, it has been proposed a meta-classifier that not only takes into account the VFH histogram but also the absolute geometry of the object (height, width and depth) and color texture (Red, Green and Blue). The following Fig. 4 contains some of the items included in the database created specifically for this study.

Fig. 4: 3D and color representations for the different objects that form the database. Notice that most of the items are geometrically similar.

As it can be noticed, there exist objects with similar surface shape but different size or color. This becomes uncertain and miscalculated if only FPFH features are taken into account during the categorization. For this

reason, a meta-classifier has been proposed as it is represented in the following Fig.5.

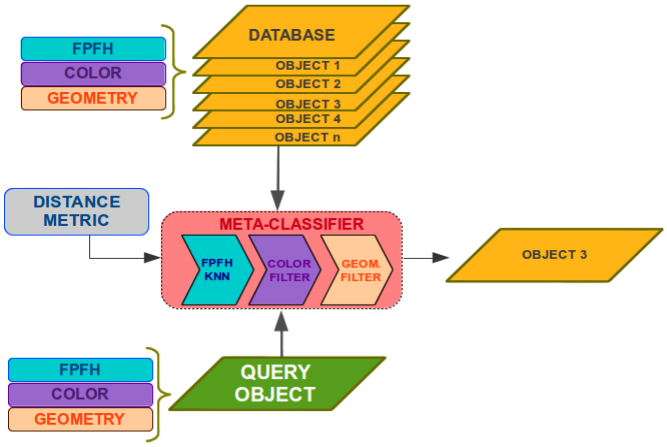

Fig. 5: Diagram of the classifier proposed for this study. Distance metric can be changed before filtering is done. The meta-classifier takes into account 3D surface descriptors, color and geometry for each candidate.

Geometry features contains absolute measures in the three axes giving more significance to the height value. That condition is imposed to maintain the view-point invariance since width and depth depend on the rotation of the object while height maintains constant relative to the supporting plane, maintain detached from yaw angle.

The next subsections explain each filter part of the complete meta-classifier. Firstly, the VFH filter will extract those objects from the database with analogous surfaces. Secondly, geometric filter will pick out those candidates which contain similar global geometry values. Lastly color filter will be in charge of selecting the most color-like remaining candidate from the database.

3.1 Distance metric

In order to measure distances between distributions, several distance metrics have been proposed. One of the aims of this research is to determine the degree of reliability for several *f-functions* to distinguish between

different VFH histograms and color histograms. The list of functions and their equations can be found in the following Table 1. It will be assumed that all the distributions are normalized in order to satisfy the metrics conditions.

Table. 1: Distance metrics used for this study. Name of the function and the discrete-form equation

Distance metric	**Distance function**
Euclidean	$d = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2}$
Manhattan	$d = \sum_{i=1}^{N} \lvert p_i - q_i \rvert$
Bhattacharyya	$d = -\ln \sum_{i=1}^{N} \sqrt{p_i - q_i}$
Kullback-Leibler	$d = \sum_{i=1}^{N} p_i \ln \frac{p_i}{q_i}$
Chi-Square	$d = \sum_{i=1}^{N} \frac{(p_i - q_i)^2}{p_i + q_i}$
Histogram Intersection Kernel	$d = \sum_{i=1}^{N} \min(p_i, q_i)$

3.2 VFH filter

The earlier filter in the classifier extracts the VFH descriptor for the observed cluster and compares it with the database. Because this operation might spend a lot of time, a KNN tree is previously built with the catalog of objects already added to the database. This advancement achieves faster execution times.

The aim of this process is to determine which the ten most similar candidates are taking into account exclusively VFH descriptors. Those aspirants could have various scales or textures and the same surfaces. Fig. 6 represents a list of the 5 nearest neighbors for a single candidate that it is also shown (first of the second row).

Candidates are sorted by the metric function distance desired and selected. Notice that distance is zero for the first guess because it is already in the training database. This will never happen due to systematic errors in observation and sensors.

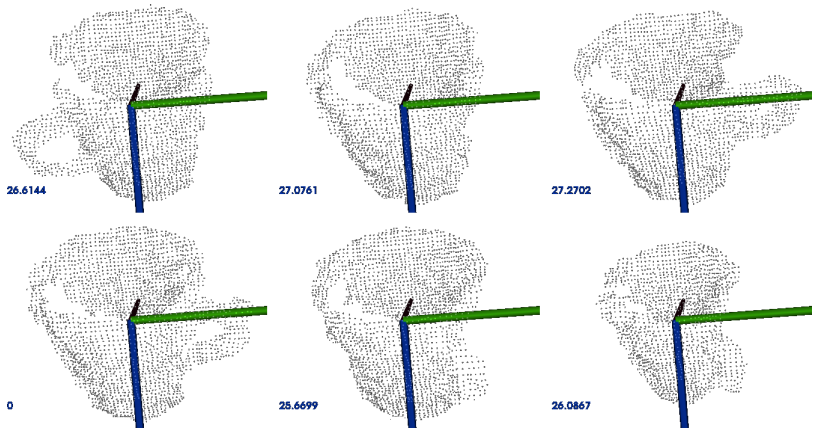

Fig. 6: Result of searching for similar objects in the database. The fourth item is the query and the VFH distance for this match is 0 because it is included in the database.

The KNN tree modeled contains, for each leaf, the VFH histogram of a single view for a specific object. VFH implementation uses 45 binning subdivisions for each of the three extended FPFH values, plus another 45 binning subdivisions for the distances between each point and the centroid and 128 binning subdivisions for the viewpoint component.

Upcoming Fig. 7 represents a set of views of the same object and their individual VFH values making it easy to understand the structure of the database.

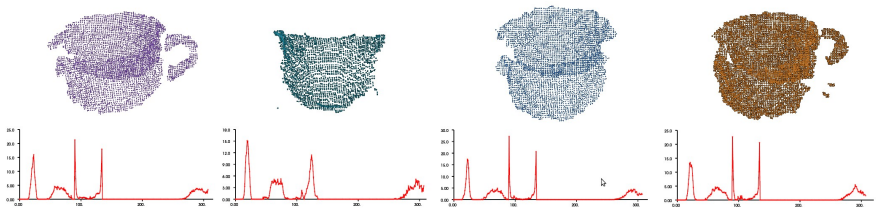

Fig. 7: Representation of several items in 3D and their corresponding View-Point feature Histogram in red. Histograms look quite similar due to their real look-alike.

3.2 Geometry filter

Subsequent filter step of the classifier is focused on geometry absolute geometric values: height, depth and width for each view. The most valuable measure is height because both width and depth are weak independent from view point and rotation angle. This makes sense because the higher part of the point cloud will always be observed by the camera sensor, and therefore successfully determine, while depth or width can be mistaken due to occlusions.

One arising problem when capturing these three values is determining the absolute position of the object according to the supporting plane global position. For this reason, it is necessary to extract the plane equation before computing the point cloud bounding box.

The parametric equation of the supporting plane has been extracted applying RANSAC over the whole point cloud and considering that only 30% of the points are outliers. Fig. 8 shows the supporting plane segmentation and object clustering and also geometry parameters (height, width and depth) computed on the projected point cloud.

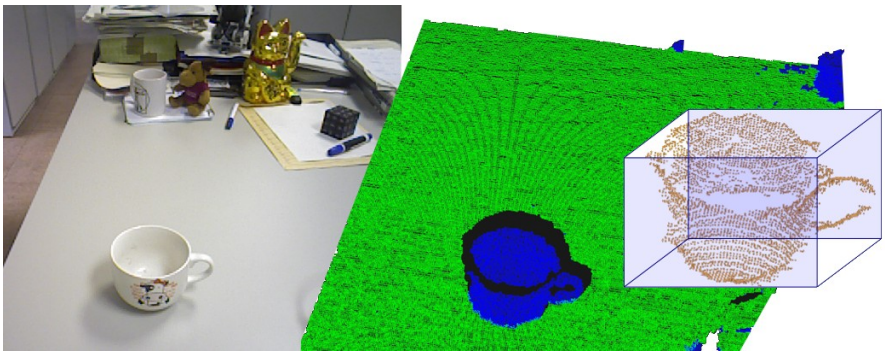

Fig. 8: Display of a real scenario with noise items in the back. Right image shows the supporting table segmentation and projected bounding box.

3.3 Color filter

Last step of the classifier is determining differences in texture between similar candidates. This last condition makes the complete architecture able to distinguish between objects with similar surface shapes and identical geometrical sizes but different colors.

There exist multiple ways of comparing textures among objects. For this paper, each color histogram R, G and B will be assumed as an independent distribution and the distance metric between objects will be chosen from one of the previously mentioned in Table 1. Each color distribution will be normalized to remove cluster size dependency.

4 Experimental results

A good way for discussing the accuracy of an object recognition algorithm with similar candidates is creating a confusion matrix that collects all of them. This kind of studies compares the whole range of classes and explains the correlation between them, intrinsic similarities and the degree of segmentation among them.

For this study, several confusion matrices will be shown. The first experiments will cover the dependency of distance matrices when comparing VFH features and RGB histograms. The last investigations will focus on

the performance of the meta-classifier and its accuracy depending which single filters are applied.

4.1 Distance metric performance

In this experiment, several distance metrics have been evaluated in order to measure their performance and select the most suitable function to evaluate the distance between feature histograms. As it has been mentioned in the previous section, six classic distance functions will be used: Mahattan distance (L1), Euclidean (L2), Bhattacharyya, Kullback-Leibler divergence, Chi-Square distance and Histogram Intersection Kernel.

Their performance has been measured with a single candidate (tinymug) in different poses and distances relative to the sensor view point with the aim of making the trials more realistic and the experiment richer.

Table. 2: Recognition rates for different distance metrics.

L1	**L2**	**Bhattacharyya**	**KL**	X^2	**HIK**
63%	66%	46%	84%	74%	23%

The above results have been computed using the following equation

$$Performance = 100 \cdot \frac{\sum_{i=1}^{N} 1 - |o_i - r_i|}{\sum_{i=1}^{N} r_i} (\%) \quad (5)$$

where o_i corresponds to each bin of the observed VFH histogram and r_i represents the real and stored value for this view.

Norm L1 and L2 are quite similar for this kind of distributions while HIK does not fit properly. Results show up that the best achievement is performed using Kullback-Leibler when contrasting feature histograms. This distance metric will be used in the following test.

4.2 Confusion matrices

Next experiment is focused on measuring the improvement of the filter as long as it is activated internally. That is, by means of confusion matrix it is achievable to compare detection rates for similar candidates and therefore understand the degree of confusion or precision of the filter as it is applied each of the previous stages. Three matrices have been therefore computed. Each one represents a different version of the filter. First version includes only the VFH filter, next version includes the geometric filter and the last differentiates among colors and textures.

VFH filter

Table. 3: Confusion matrix for VFH filter. Notice the high degree of confusion among scaled versions of the same objects.

	mug -tiny	**mug -medium**	**mug -big**	**mug-robot-white**	**mug-robot-red**	**mug-cow**	**teddy**
mug-tiny	**7**	41	3	19	30	0	0
mug-medium	0	**29**	26	18	27	0	0
mug-big	0	0	**84**	1	8	5	2
mug-robot-white	0	0	0	**26**	74	0	0
mug-robot-red	0	0	0	49	**51**	0	0
mug-cow	0	0	34	0	0	**59**	7
teddy	0	0	0	0	0	0	**100**

The average recognition rate for this filter is 50.85%. Notice that it takes uniquely in account the 3D surface shape of each object, confusing objects with similar surfaces such as mug-robot-white and mug-robot-red, which are highly mixed up. The same is happening between mug-medium and mug-big, where the recognition rates are below 35%.

VFH+ Geometry filter

Table. 4: Confusion matrix for VFH+Geometry filter. Objects with similar geometry but different textures are still confused

	mug -tiny	**mug -medium**	**mug -big**	**mug-robot-white**	**mug-robot-red**	**mug-cow**	**teddy**
mug-tiny	**49**	29	0	10	12	0	0

mug-medium	0	**40**	11	11	38	0	0
mug-big	0	0	**99**	1	0	0	0
mug-robot-white	0	0	0	**77**	23	0	0
mug-robot-red	0	0	0	30	**67**	0	0
mug-cow	0	0	1	0	0	**99**	3
teddy	0	0	0	0	0	0	**100**

Second experiment takes into account not only the surface features VFH but also the global geometry of the object. With this addition, it is expected to decrease the confusion error between objects with similar shape but different size such as the mug-tiny, mug-medium and mug-big, where confusion has been demonstrated to decrease markedly. Notice that mug-cow and mug-big are now much better segmented due to their distinction. The average recognition rate for this test has been raised to 75.85%.

VFH+ Geometry + Color filter

Table. 5: Confusion matrix for VFH+Geometry+Color filter. Confusion matrix is higher diagonal stating the degree of success for the filter.

	mug -tiny	**mug -medium**	**mug -big**	**mug-robot-white**	**mug-robot-red**	**mug-cow**	**teddy**
mug-tiny	**84**	12	0	4	0	0	0
mug-medium	20	**60**	0	20	0	0	0
mug-big	3	0	**96**	0	0	1	0
mug-robot-white	0	0	0	**100**	0	0	0
mug-robot-red	0	0	0	11	**88**	0	1
mug-cow	0	0	0	0	0	**100**	0
teddy	0	0	0	0	0	0	**100**

Last filter compares VFH features, geometric features and color texture for each object with respect to the observation. In this case, confusion matrix's diagonal is strictly higher and therefore correlation between candidates is well diminished. Mug-robot-red and mug-robot-white, which are geometrically identical but in different color, are well segmented and recognized while the rest of the distinctions rates are maintained constant as

expected. The average recognition rate for this last experiment, including the three filter steps is notoriously higher reaching 89.71%.

5 Conclusions

This paper proposes an improvement of View-Point Feature Histogram based on geometrical and color texture features for 3D objects. Several experiments have been performed concluding Kullback-Leibler divergence as the most suitable selection for VFH comparisons. Furthermore, confusion matrices for the three states of the filter have been studied, correcting from the initial 50.85% of recognition rate for single VFH features to 89.71% for with confusable 3D objects.

Acknowledgements

The authors would like to thank the Spanish Government for the funds provided by the HYPER CONSOLIDER-INGENIO 2010 project and the FPU grant provided by the Spanish Education Ministry AP2009-3445. The research leading to these results has received funding from the ARCADIA project DPI2010-21047-C02-01, funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness and from the RoboCity2030-II project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006* (pp. 404-417). Springer Berlin Heidelberg.

Bueno, J.G.; Slupska, P.J.; Burrus, N.; Moreno, L. (2011) "Textureless object recognition and arm planning for a mobile manipulator," *ELMAR, 2011 Proceedings* , vol., no., pp.59,62, 14-16.

Bueno, J. G.; Fierro, M. G.;Moreno, L.; Balaguer, C. (2012) Facial Gesture Recognition using Active Appearance Models based on Neural Evolution . *2012 Conference on Human-Robot Interaction (HRI 2012)*. Boston. USA. Mar.

Knopp, J., Prasad, M., Willems, G., Timofte, R., & Van Gool, L. (2010). Hough transform and 3D SURF for robust three dimensional classification. In Computer Vision–ECCV 2010 (pp. 589-602). Springer Berlin Heidelberg.

Liu, Y., Emery, R., Chakrabarti, D., Burgard, W., & Thrun, S. (2001). Using EM to learn 3D models of indoor environments with mobile robots. In-*MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE*-(pp. 329-336).

Martin, F., Munoz, M. L., Garrido, S., Blanco, D., & Moreno, L. (2009). L1-norm global localization based on a Differential Evolution Filter. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on* (pp. 229-234). IEEE.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In*3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on* (pp. 145-152). IEEE.

Rusinkiewicz, S., Hall-Holt, O., & Levoy, M. (2002). Real-time 3D model acquisition. In *ACM Transactions on Graphics (TOG)* (Vol. 21, No. 3, pp. 438-446). ACM.

Rusu, R. B., Marton, Z. C., Blodow, N., Holzbach, A., & Beetz, M. (2009). Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (pp. 3601-3608). IEEE.

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 3212-3217).

Rusu, R. B., Bradski, G., Thibaux, R., & Hsu, J. (2010). Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots*

and Systems (IROS), 2010 IEEE/RSJ International Conference on (pp. 2155-2162).

Rusu, R. B. (2010). Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. KI-Künstliche Intelligenz, 24(4), 345-348.

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia* (pp. 37-360). ACM.

Steder, B., Rusu, R. B., Konolige, K., & Burgard, W. (2010). NARF: 3D range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* V. 44