Robustness Analysis of 3D Feature Descriptors for Object Recognition Using a Time-of-Flight Camera

Levente Tamas[1] and Bjoern Jensen[2]

***Abstract*— In this paper we propose to analyze characteristics of the feature descriptors in terms of robustness against typical disturbances in the context of the object recognition pipeline for depth data with intensity information. In terms of robustness the focus was on the occlusion handling, segmentation errors, sub-sampling of data as well as the presence of Gaussian noise in data. For this analyses we considered a set of real life data captured in an indoor environment using a time-of-flight sensor returning depth and intensity data. According to our test results the intensity spin estimator and the ensemble of shape functions type of feature descriptors proved to be the most suitable variant for such object recognition tasks.**

I. INTRODUCTION

The use of 3D perception sensors in the mobile robotics application became popular during the last few years. This is mainly due to the appearance of the affordable depth sensors such as the time-of-flight or projected infrared (e.g Kinect) ones. The research focus in the field of object descriptors, keypoints and classification [1], [2] grow especially for the Kinect like sensors, while for the time-of-flight sensors such as the SwissRanger it is rather limited. Although there are similarities in the characteristics of the two sensors, the quality of depth data acquired with the SR is smoother compared to the Kinect like senors, and it can used daytime in outdoor environment [3]. Instead of RGB data in the case of the SR camera family for each 3D point the intensity and the confidence information for that point is available. Thus a different approach and behavior is expected for the 3D point descriptors for the data emerging from the SR cameras.

In this paper we propose a thorough analyses for the already implemented feature descriptors in the PCL [4] library available with a BSD type license. This analyze is focused on the data acquired with a Swiss Ranger 4000 (SR4K) time-of-flight camera [5] in a indoor environment for a large range of objects. As typical error sources the occlusion, subsampling, boundary errors and signal noise are considered as separate test cases. For performance metrics the receiver operating characteristic (ROC) [6] curve is considered based on the true positives and false positives during the recognition tests. Further on, the size and runtime of the different descriptors are considered. Based on these evaluations the best descriptor can be considered for the application dealing with time-of-flight camera data recognition.

*This work was supported by Sciex-NMS project nr. 12.239

[1] Technical University of Cluj-Napoca, Robotics Research Group, Dorobantilor st. 71-73, 400609, RO; Levente.Tamas@aut.utcluj.ro

[2] Bern University of Applied Sciences, Institute for Human Centered Engineering, roboticsLab, Quellgasse 21, Biel/Bienne, CH-2502; bjoern.jensen@bfh.ch

II. DATA PREPROCESSING AND RECOGNITION

This section presents in detail the data base building as well as the recognition pipeline used in testing phase. The details regarding the data filtering, object segmentation and recognition are highlighted as well.

A. Data set acquisition

In order to build up a test database for the feature descriptor analysis 15 different sized objects were considered. The list of objects captured with the SR4K camera are listed in Figure 1.

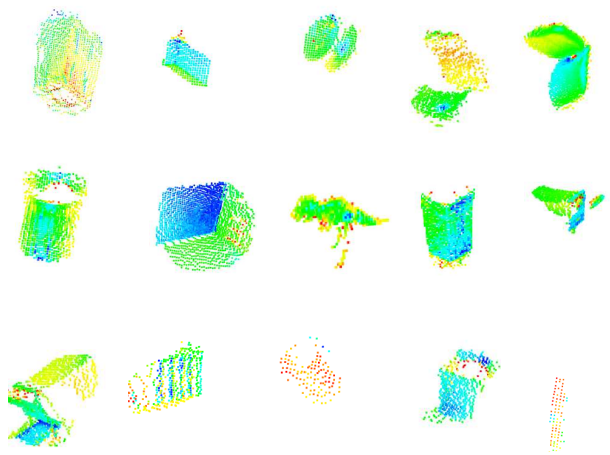


Fig. 1: Test objects considered during the evaluation (top-left to bottom-right): box, small box, cable-holder, chair1, chair2, cylinder, drawer, dyno, hexagon, open-drawer, open-drawer2, paper-holder, small-cylinder, trash-bin, tube.

These objects vary in size from $0.1m$ to $1.5m$ and are well observable with the time-of-flight sensor, i.e they are not transparent nor lucid surfaces. Also note that, objects from the same category were considered, i.e two different chairs or a drawer with different configurations (with one and with two open doors), which makes the recognition and thus the feature descriptor evaluation challenging. There were more than 300 different captures made for these objects from different views, distances and positions.

B. Filtering

The first step of the object recognition pipeline was the filtering of the raw data. The main role of this step was to reduce the outliers with a statistical filtering and to get a compact representation of the data with a voxel-grid type

filter. Also a pass-through filter was considered in order to cancel out the false readings from the SRK4 sensor, which are often present like far shadows for the objects. All these filters are part of the PCL library. In our setup the tuned parameters for the filters were as follows: for the voxel grid we considered a grid size of $0.01m$, which is close to the actual resolution of the sensor, while for the statistical outliers we considered 50 points and with the standard deviation threshold of 0.8.

C. Object segmentation

In the next step of the data preprocessing the major planes are segmented out in order to get the objects from the scene. This is an important step in order to end up with the data containing only the objects from a scenario. In order to achieve this, planar models are fitted to the acquired data, and the largest planes up to a tuned percentage are removed. The plane fitting is performed in a standard sampling consensus (SAC) approach, and for the plane removal parameter we set to 25 percentage of the original data.

D. Object recognition

The object recognition domain is a well evaluated one especially in the 2D space, special contests being organized for this topic such as the Pascal VOC. In the last few years the recognition in the 3D domain got into the focus with major contributions such as the work [7]. These ones often make use of techniques and principles developed in the 2D domain and extrapolated to the 3D case. This is valid also for the point-correspondence based recognition. This technique usually makes use of feature descriptors for the points in order to perform an efficient recognition.

After the feature descriptors are extracted, for the feature comparison different distance metrics can be used also such as the Euclidean, L_1, L_2, Hellinger or χ^2. In our approach we used the χ^2 metric of two feature vectors $\mathbf{x}$ and $\mathbf{y}$ with length N such as follows:

$$\chi_{\mathbf{x},\mathbf{y}} = \sqrt{\frac{\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{y}_i)^2}{\sum_{i=1}^{N}(\mathbf{x}_i + \mathbf{y}_i)}} \tag{1}$$

This metric proved to be stable and good weighting for the different types of features used in our evaluation tests. For the feature correspondence search the nearest-neighbors (NN) search was used with the above mentioned metric. The details regarding the used feature descriptors are presented in the next section.

III. DEPTH FEATURE DESCRIPTORS

The depth feature descriptors as well as the image descriptors are compact representation of data. Beside the fact that the data is represented in a compact form, the features tend to contain rich information extracted from the data. Another important general characteristics is the invariance with respect to certain transformations and disturbances. This invariance is essential in order to use them as discriminators in a recognition type applications.

Two main categories of descriptors are distinguished based on the range of data on which are computed: local and global ones [8]. Global descriptors are meant to be used to describe a larger set of data containing objects of interest, and they capture the geometric properties of these objects. Local descriptors in contrast are usually computed on a small subset of representative data (keypoints) for already segmented scenes, and are capturing properties around these keypoints. Both types of descriptors have their advantages/disadvantages, which we analyzed in our robustness test benches.

A. Rotation invariant feature transform

The rotation invariant feature transform (RIFT) can be applied to 3D data containing intensity information too. Originally it was proposed in the work of [9] as an extension to the SIFT image descriptors [10]. The algorithm iterates over each point P_i in the input point cloud and and within a sphere of radius r all the points are considered to belong to a subset P_{ik}. An imaginary circle with 4 bins (rings) are considered perpendicular to the normal at the point P_i. All neighbors of the selected point are assigned to a ring based on the relative distance based on gradient histogram computed with 8 bins using a thresholding. Thus a total number of 32 histograms are computed with this technique, which describe circular features of a point P_i.

B. Intensity Gradient Estimator

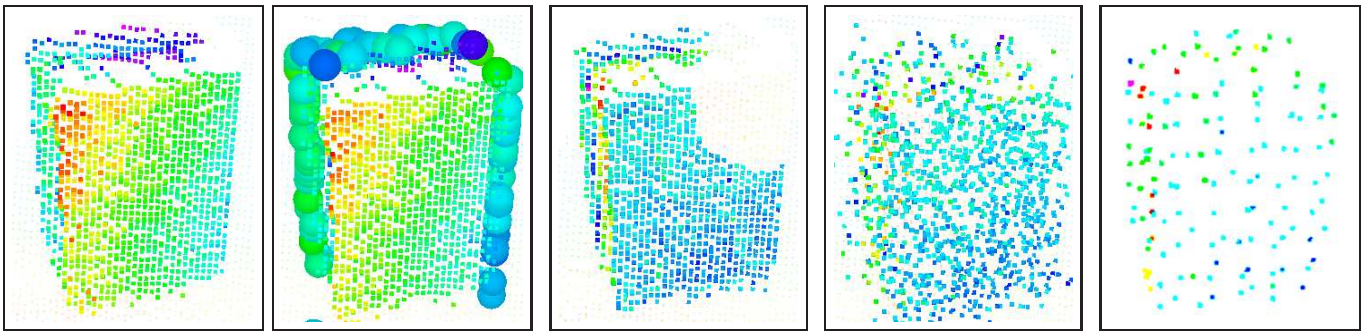
The intensity gradient estimator (IGE) uses as input depth data with intensity information. In the first step the normals of the 3D points are computed, and for each point the analytical intensity gradient is computed along the surface direction of the considered region. This technique is similar to the 2D image intensity computation [11]. In the final step the computed gradient is projected onto the surface, and the output of the estimator is the projection gradient vector containing the intensity information computed for each input point.

C. Intensity Spin Estimator

This type of descriptor is based on the work of [9], however the idea of using intensity information as a descriptor was already present in the earlier work [12]. In contrast to the IGE type of descriptor in this case there is no need for explicit normal pre-computation at the input point cloud, which gives a considerable speed-up for this algorithm. As tuning parameters the point distance and intensity distance bins can be set, having the same meaning as in the case of the RIFT descriptor.

D. Spin Image Estimator

The original idea for the spin image (SI) estimation is presented in the work [12] and can be applied to depth data with pre-computed normals. The algorithm computes two types of distances: the distance of the normals computed at a point and the source normal n and the between the from the considered point along n. The distances larger than a tuning



(a) a (b) b (c) c (d) d (e) e

Fig. 2: 3D data corrupted with different noise (best viewed in color): (a) the nominal data, (b) the contour segmentation noise, (c) occlusion noise, (d) Gaussian noise, (e) sampling noise

threshold are rejected. From the remaining distance pairs a histogram is built, which represents the occurrence of the discrete distance pairs.

E. Point Feature Histogram

The local point feature histogram (PFH) [13] descriptor extends the original surflet-pair relation histograms suggested in the work of [14]. The input for this feature descriptor is a pointcloud with normals. In the first step for each point P_i the neighbors within a search radius are computed, denoted with sets P_{ik}. Within these sets point pairs are considered denoted with P_s and P_t with the meaning source and target. For these pairs, the difference of normals are computed and, described with 3 angles around the axis and a distance. As the distance if varying with viewpoint, this can be left out. Finally, these angles are considered to be sorted in the 125 binned histogram, which is the output of the algorithm for each point.

F. Fast Point Feature Histogram

The fast point feature histogram (FPFH) [13] is an extension of the PFH yielding to computationally less expensive variant of the PFH. The major difference between PFH and FPFH is that while in the case of PFH all pairs of points are considered in the subsets P_{ik}, in this case only the point pairs between P_i and the rest of the point within P_{ik} are considered. Thus the computation cost drops from $O(nk^2)$ to $O(nk)$. The three angles in this case are binned into a 11 bin histogram, the total length of the obtained histogram is 33 for each point.

G. Viewpoint Feature Histogram

The global viewpoint feature histogram (VFH) [15] describes the pointcloud P as containing two components: a component representing the viewpoint of the scene and one containing the FPFH features. In this case the FPFH features are binned into a 45 bin histogram, and the distance between the points is also taken into account, thus a total number of 4 extended FPFH features are stored. The additional view point feature is computed by taking the centroid of the pointcloud denoted with P_c and computing the FPFH for each neighbors. The later histogram is represented using 128 bins, thus the total number of bins for a pointcloud is 308 for this descriptor for the entire pointcloud.

H. Clustered Viewpoint Feature Histogram

The clustered viewpoint feature histogram (CVFH) [16] is an extension of the VFH in order to handle occlusion or other types of sensor noise. This is mainly important for the VFH, as in case of an occluded view of the same object the histogram of the descriptor varies considerably. The basic idea of the CVFH is the construction of stable regions (clusters) S which step is done by computing compact regions using the region growing approach with thresholding on the normal values. Once these S regions are computed, the VFH for each of them is determined, and an additional shape distribution (SD) is computed as $SD = \frac{(c-p_i)^2}{sup(c-p_i^2)}$, where c is the centroid of the cluster S and the p_i represents the points from this region. This component is also stored in a binned histogram, the total number of descriptor histogram bins for a single point being equal to 308.

I. Ensemble of Shape Functions

The ensemble of shape functions (ESF) type descriptor was proposed in the work [17] which is based on the A3, D2, D3 shape descriptor functions and extends the D2 type description presented in [18].

The algorithm starts with selecting a subset of 20000 points from the pointcloud, and samples three random points from this P_a, P_b, P_c. The D2 distance is based on the metric distance between the points P_a and P_b. In the next step is verified weather the line connecting the two points are on the surface (in), or out the surface (out) or both (mixed). The corresponding bin for the D2 distance is incremented at the computed bin. This procedure is repeated for the remaining two point-pairs.

J. Radius-based Surface Descriptor

The Radius-based Surface Descriptor (RSD) [19] has as input an oriented pointcloud, i.e. with normals and describes the local geometry of a point with respect to its neighbors in terms of radii. Each point pair is supposed to lie on a sphere,

and the distance of the points d and the angle between the normals at the two points has the relation:

$$d(\alpha) = \sqrt{2r} \cdot \sqrt{1 - cos(\alpha)} \approx r\alpha + r\alpha^3/24 + O(\alpha^5) \quad (2)$$

The equation holds for $\alpha \in (0, \pi/2)$, while for an accurate estimation of the radii a linear regression is applied on the extremas of the (α, r) pairs. Also an intuitive geometric interpretation of the obtained radii makes it usable for surface categorization, i.e. the large radii denotes planar surfaces, while small radii is for cylindric objects. For the recognition test we used the radii as histograms computed for individual objects.

K. Principal Curvature Descriptor

The principal curvature (PC) descriptor as its name suggest computes the curvature for an oriented pointcloud, and uses this information as descriptor. Beside the curvatures, it computes the magnitudes and directions for each point along the XYZ axes, and stores the largest and smallest values for these ones. As tuning parameter the maximum radii for neighbors search is considered.

IV. ROBUSTNESS TEST

In our robustness tests we investigated the sensibility of the different feature descriptors with respect to common disturbances encountered during the object recognition pipelines (see Figure 2). For qualitative evaluation we considered the ROC curve, which incorporates the information from a confusion matrix, while for the quantitative comparison the AC_d metric was used defined as [6]:

$$AC_d = 1 - \sqrt{W \cdot (1 - TP)^2 - (1 - W) \cdot Fp^2} \quad (3)$$

where TP and FP denote the true positives respectively the false positives and W is a weighting factor giving an application specific weighting to Tp in favor of FP.

A. Test methodology

The test methodology followed in this paper is similar to the generic test procedures in the main literature with the focus on the recognition applications. This means that we divided our captured dataset into $1/3$ and $2/3$ ratio used for training and testing purposes. The training set we denote with *templates* while the testing objects we consider as *observations* of the template objects.

In the training phase we compute for each object each aforementioned descriptor, and store the descriptors in a FLANN [20] data structure and store it for later queries. This can be viewed also as an offline phase in a real life recognition application.

In the test phase each of computed descriptors for the considered object is searched in the FLANN database containing the specific template feature descriptors. The object is compared against each other object in the database and the one with the smallest metric defined in eq. (1) is considered to be a match. In this manner are determined the indicators for the recognition such as the T_p expressing the true positive rate and F_p the incorrect classification rate for the objects.

Another important aspect of the test procedure is that for each type of tested disturbances, the added artificial disturbance is considered both for the template and for the observation side in two separate phases, thus the total object-object comparison exceeds half million test cases.

For the nominal case, i.e. no added artificial noise, the best performance f meaning closest to the point (0,1) is achieved by the PFH and ISE descriptors, while the poorest is the FPFH in this case.

B. Occlusion test results

The occlusion test tackles one of the most common issues in the object recognition pipeline, the object occlusion problem. This practically means that the template is only partially visible, i.e. an occluded observation is available for the recognition. In our experimental setup we considered variable occlusion to be tested, by removing a radial zone of an object at a random position. The amount of removed points is given as a tuning parameter expressing the removed points as a percentage of pointcloud size of the template. The results are shown in Figure 3 part (a) and part (b) for the occlusion noise at observation and at the template side.

For the observation side occlusion test case the best behavior is obtained with the CVFH. ISE and SI type of descriptors, while the poorest results are given by VFH and FPFH. Also on the template side added disturbance the best robustness is achieved with the CVFH, which proves the efficient approach of the clustered viewpoint approach for the occlusion handling.

C. Segmentation error test results

Another important source of error in the object recognition applications are emerging from the segmentation phase. The segmentation errors can be viewed also as contour noise around the object, which is present due to the inaccuracy in the segmentation process. In order to test this kind of error source, first we determined the boundary of the test object, and than we removed a tunable region for each point from the boundary. The results for this test case are shown in Figure 3 part (c) and part (d) for the observation occlusion noise and for the template side noise respectively. The best results are achieved with VFH and ISE respectively.

D. Sensor noise test results

The most common problem with the real data recognition setups is the sensor noise, i.e. measurement inaccuracies due to the physical sensor limitations. Usually the non-systematic part of the noise is approximated with a Gaussian probability distribution. In our case we used also this assumption, and we considered a system with variable added Gaussian noise covariance along the XYZ coordinates added independently.

This kind of disturbance affects drastically the performance of the CVFH and VFH at the observation side noise, while the SI is vulnerable for the template side Gaussian noise disturbances. Still in this case the ISE tolerates well the noise both on observation and template sides.

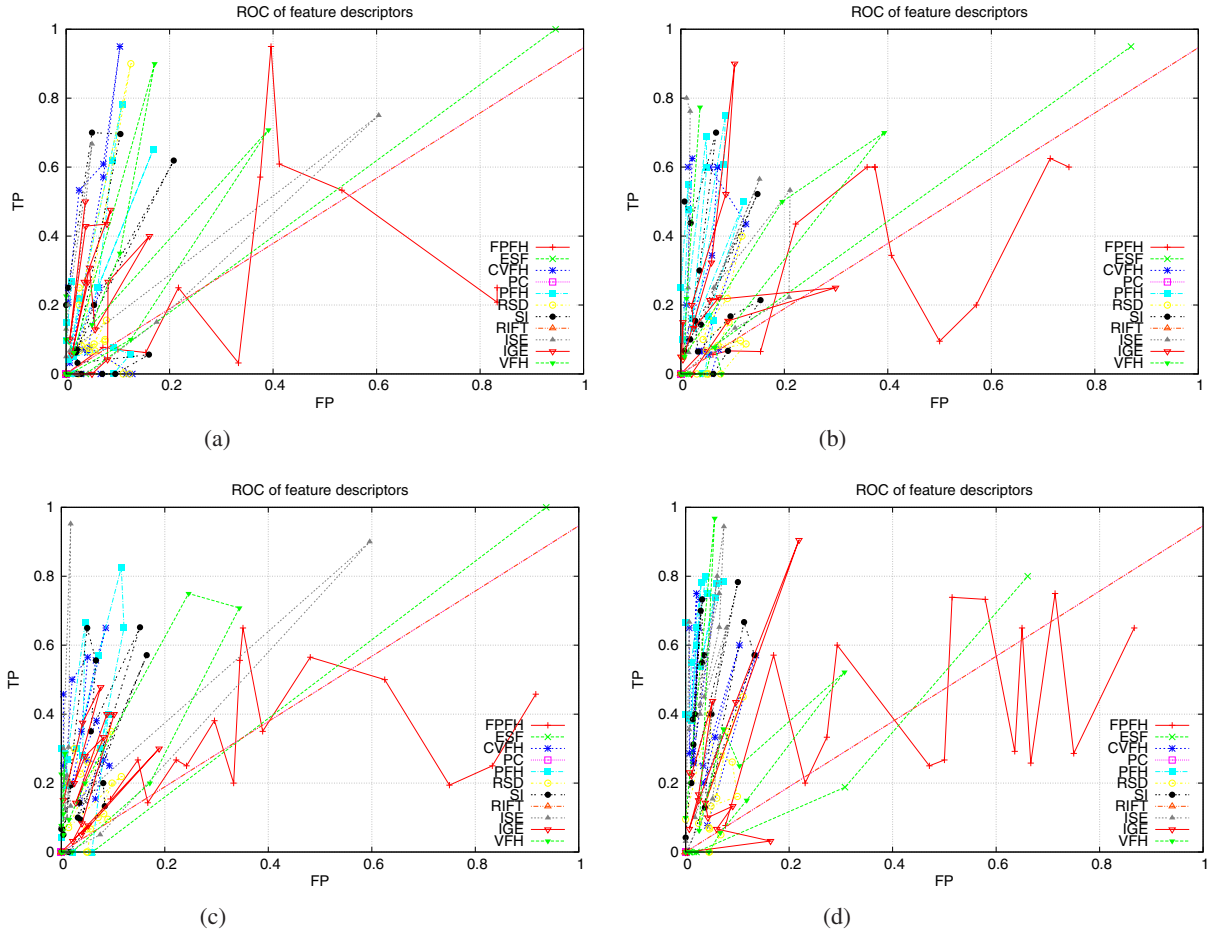


Fig. 3: 3D data corrupted with occlusion and contour noise (best viewed in color): (a) occlusion noise at the observation side, (b) occlusion noise at the template side, (c) contour noise at the observation side, (d) contour noise at the template

E. Subsampling test results

The last test case deals with the varying sampling type error. This kind of disturbance is present in the case of the time-of-flight cameras, as the same object captured at different distances from the sensor has a varying sampling ratio. Thus this is a disturbance which is important to be analyzed for such type of sensors. In our test cases we computed different subsampling by applying a tunable sized voxel grid on the original data.

The majority of the feature descriptors are robust against this type of disturbance, however the ISE giving the best performance till now fails at this test phase. Also the FPFH get worse in this case, getting close to point (1,1) n ROC.

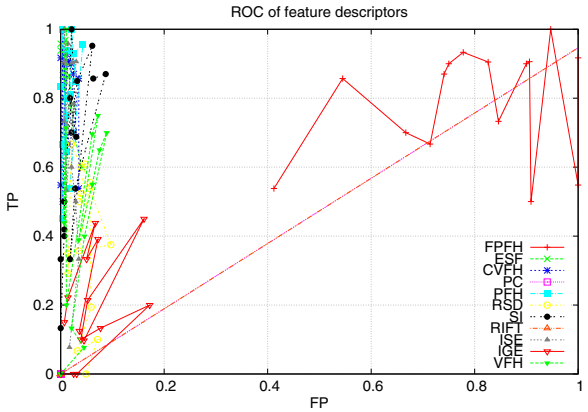


Fig. 4: Feature descriptor test result for the nominal data (best viewed in color)

F. Quantitative results

In order to have a quantitative evaluation of the test bench we considered the AC_d metrics defined in eq. (3) for the analyzed descriptors. Each descriptor varies also with the test object considered for the recognition problem, hence in order to reduce the dimensionality of the evaluation, we considered the average values for the metrics over all test object for each descriptor. This gives a good performance indicator for the different test cases without loosing the important characteristics of the test outputs. The evaluation was done for both the observation and template side disturbances.

The results for the test cases shown also on the ROC curves are summarized in the Table I for the object side noise disturbances and in the Table II for the template side

Descr.	Nom.	Occ.	Cont.	Gauss	Sampl.
VFH	0.75	0.63	0.64	0.52	0.58
CVFH	0.61	0.61	0.57	0.51	0.59
PFH	0.82	0.56	0.71	0.68	0.71
FPFH	0.41	0.46	0.42	0.46	0.40
RSD	0.73	0.52	0.62	0.62	0.77
RIFT	0.59	0.55	0.53	0.51	0.49
PC	0.48	0.44	0.45	0.39	0.46
SI	0.83	0.54	0.67	0.74	0.83
ISE	0.87	0.59	0.67	0.87	0.63
IGE	0.64	0.57	0.65	0.62	0.60
ESF	0.83	0.57	0.59	0.80	0.74

TABLE I: Observation side robustness tests containing the nominal, occlusion, contour, Gaussian and sampling noise test cases

Descr.	Nom.	Occ.	Cont.	Gauss	Sampl.
VFH	0.71	0.65	0.64	0.52	0.62
CVFH	0.81	0.69	0.75	0.54	0.74
PFH	0.78	0.71	0.73	0.61	0.690
FPFH	0.88	0.79	0.77	0.74	0.63
RSD	0.73	0.64	0.63	0.63	0.69
RIFT	0.59	0.49	0.47	0.50	0.49
PC	0.59	0.51	0.49	0.55	0.48
SI	0.83	0.66	0.75	0.62	0.84
ISE	0.87	0.66	0.76	0.86	0.64
IGE	0.64	0.65	0.65	0.61	0.63
ESF	0.89	0.59	0.59	0.79	0.77

TABLE II: Template side robustness tests containing the nominal, occlusion, contour, Gaussian and sampling noise test cases

case. The level of added noise in this test bench was $4cm$ for the contour noise removal band, 10% for the occlusion percentage of the original object, $5cm$ voxel grid for the sampling and the Gaussian noise with $0.03m$ covariance.

In average the best performance is achieved in this case with the ISE type feature descriptor and ESF for the object and template side respectively.

In average the best performance in these cases is achieved with the ISE type feature descriptor and ESF for the object and template side respectively, however the performances of the descriptor lowered considerably with the noise.

In terms of run-times, the best performance was achieved with FPFH, while computationally the most intensive ones were the ISE and IGE type of descriptors.

V. CONCLUSIONS

In this paper the problem of feature descriptor robustness was tackled for depth and intensity type of data. The main aim was to give both a qualitative and quantitative evaluation of the existing feature descriptors for the 3D domain on this type of specific data, and to test against a large variety of disturbances including the occlusion, segmentation error, sensor noise and sub-sampling cases too. The result were presented using the ROC curves and the AC_d metric for the different test benches. In the future the evaluation on outdoor data is planned to be performed as well as the extension of the recognition pipeline to handle multi-object scenarios too.

REFERENCES

[1] H. Ali, F. Shafait, E. Giannakidou, A. Vakali, N. Figueroa, T. Varvadoukas, and N. Mavridis, "Contextual object category recognition for RGB-D scene labeling," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 241–256, Feb. 2014.

[2] L. Tamas and A. Majdik, "Heterogeneous feature based correspondence estimation," in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, IEEE. Munich, Germany: IEEE, September 2012, pp. 89–94.

[3] A. Dopfer, H.-H. Wang, and C.-C. Wang, "3D Active Appearance Model alignment using intensity and range data," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 168–176, Feb. 2014.

[4] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1 –4.

[5] A. Belhedi, S. Bourgeois, V. Gay-Bellile, P. Sayd, A. Bartoli, and K. Hamrouni, "Non-parametric depth calibration of a tof camera," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 549–552.

[6] T. Fawcett, "An introduction to roc analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, June 2006. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2005.10.010

[7] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3D Scene Reconstruction and Class Segmentation," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 97–104, June 2013.

[8] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation." *IEEE Robot. Automat. Mag.*, vol. 19, no. 3, pp. 80–91, 2012.

[9] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] R. Shams, R. A. Kennedy, P. Sadeghi, and R. I. Hartley, "Gradient intensity-based registration of multi-modal images of the brain." in *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*. IEEE, 2007, pp. 1–8.

[12] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[13] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1848–1853.

[14] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification," in *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, Oct 2003, pp. 474–481.

[15] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in *Proceedings of the 23rd IEEE International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[16] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 585–592.

[17] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, Dec 2011, pp. 2987–2992.

[18] C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli, "Using shape distributions to compare solid models," in *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, ser. SMA '02. New York, NY, USA: ACM, 2002, pp. 273–280.

[19] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D-3D categorization and classification for multimodal perception systems," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1378–1402, Aug. 2011.

[20] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.