A Comparative Evaluation of 3D Keypoint Detectors in a RGB-D Object Dataset

Sílvio Filipe and Luís A. Alexandre
Informatics, IT - Instituto de Telecomunicações, University of Beira Interior, Department of Informatics, 6200-001 Covilhã, Portugal
sfilipe@ubi.pt, lfbaa@ubi.pt

Keywords: 3D Keypoints, 3D Interest Points, 3D Object Recognition, Performance Evaluation.

Abstract: When processing 3D point cloud data, features must be extracted from a small set of points, usually called keypoints. This is done to avoid the computational complexity required to extract features from all points in a point cloud. There are many keypoint detectors and this suggests the need of a comparative evaluation. When the keypoint detectors are applied to 3D objects, the aim is to detect a few salient structures which can be used, instead of the whole object, for applications like object registration, retrieval and data simplification. In this paper, we propose to do a description and evaluation of existing keypoint detectors in a public available point cloud library with real objects and perform a comparative evaluation on 3D point clouds. We evaluate the invariance of the 3D keypoint detectors according to rotations, scale changes and translations. The evaluation criteria used are the absolute and the relative repeatability rate. Using these criteria, we evaluate the robustness of the detectors with respect to changes of point-of-view. In our experiments, the method that achieved better repeatability rate was the ISS3D method.

1 INTRODUCTION

The computational cost of descriptors is generally high, so it does not make sense to extract descriptors from all points in the cloud. Thus, keypoint detectors are used to select interesting points in the cloud on which descriptors are then computed in these locations. The purpose of the keypoint detectors is to determine the points that are different in order to allow an efficient object description and correspondence with respect to point-of-view variations (Mian et al., 2010).

This work is motivated by the need to quantitatively compare different keypoint detector approaches, in a common and well established experimentally framework, given the large number of available keypoint detectors. Inspired by the work on 2D features (Schmid et al., 2000; Mikolajczyk et al., 2005) and 3D (Salti et al., 2011), and by a similar work on descriptor evaluation (Alexandre, 2012), a comparison of several 3D keypoint detectors is made in this work. In relation to the work of Schmid et al. (2000); Salti et al. (2011), our novelty is that we use a real database instead of an artificial, the large number of 3D point clouds and different keypoint detectors. Regarding the paper Filipe and Alexandre (2013), the current paper introduces in the evaluation of 4 new keypoint detectors, makes a computational complexity evaluation through the time spent by each method on the experiments. The benefit of using real 3D point clouds is that it reflects what happens in real life, such as, with robot vision. These never "see" a perfect or complete object, like the ones present by artificial objects.

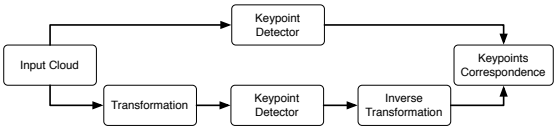


Figure 1: Keypoint detectors evaluation pipeline used in this paper.

The keypoint detectors evaluation pipeline used in this paper is presented in figure 1. To evaluate the invariance of keypoint detection methods, we extract the keypoints directly from the original cloud. Moreover, we apply a transformation to the original 3D point cloud before extracting another set of keypoints. Once we get these keypoints from the transformed cloud, we apply an inverse transformation, so that we can compare these with the keypoints extracted from the original cloud. If a particular method is invariant to the applied transformation, the keypoints ex-

tracted directly from the original cloud should correspond to the keypoints extracted from the cloud where the transformation was applied.

The low price of 3D cameras has increased exponentially and the interest in using depth information for solving vision tasks. A useful resource for users of this type of sensors is the Point Cloud Library (PCL) (Rusu and Cousins, 2011) which contains many algorithms that deal with point cloud data, from segmentation to recognition, from search to input/output. This library is used to deal with real 3D data and we used it to evaluate the robustness of the detectors with variations of the point-of-view in real 3D data.

The organization of this paper is as follows: the next section presents a detailed description of the methods that we evaluate; the results and the discussion appear in section 3; and finally, we end the paper in section 4 with the conclusions.

2 EVALUATED 3D KEYPOINT DETECTORS

Our goal was to evaluate the available descriptors in the current PCL version (1.7 pre-release, on June 2013).

There are some keypoint detectors in PCL which we will not consider in this paper, since they are not applicable to point cloud data directly or only support 2D point clouds. These are: Normal Aligned Radial Feature (NARF) (Steder et al., 2010) that assume the data to be represented by a range image (2D image showing the distance to points in a scene from a specific point); AGAST (Mair et al., 2010) and Binary Robust Invariant Scalable Keypoints (BRISK) (Leutenegger et al., 2011) that only support 2D point clouds.

2.1 Harris3D

The Harris method (Harris and Stephens, 1988) is a corner and edge based method and these types of methods are characterized by their high-intensity changes in the horizontal and vertical directions. These features can be used in shape and motion analysis and they can be detected directly from the grayscale images. For the 3D case, the adjustment made in PCL for the Harris3D detector replaces the image gradients by surface normals. With that, they calculate the covariance matrix Cov around each point in a 3×3 neighborhood. The *keypoints response* measured at each pixel coordinates (x,y,z) is then defined by

$$r(x,y,z) = det(Cov(x,y,z)) - k\,(trace(Cov(x,y,z)))^2\,, \quad (1)$$

where k is a positive real valued parameter. This parameter serves roughly as a lower bound for the ratio between the magnitude of the weaker edge and that of the stronger edge.

To prevent too many keypoints from lumping together closely, a non-maximal suppression process on the keypoints response image is usually carried out to suppress weak keypoints around the stronger ones, followed by a thresholding process.

2.2 Harris3D Variants

In the PCL, we can find two variants of the Harris3D keypoint detector: these are called Lowe and Noble. The differences between them are the functions that define the keypoints response (equation 1). Thus, for the Lowe method the keypoints response is given by:

$$r(x,y,z) = \frac{det(Cov(x,y,z))}{trace(Cov(x,y,z))^2}. \quad (2)$$

The keypoints response from Noble method is given by:

$$r(x,y,z) = \frac{det(Cov(x,y,z))}{trace(Cov(x,y,z))}. \quad (3)$$

2.3 Kanade-Lucas-Tomasi

The Kanade-Lucas-Tomasi (KLT) detector (Tomasi and Kanade, 1991) was proposed a few years after the Harris detector. In the 3D version presented in the PCL, this keypoint detector has the same basis as the Harris3D detector. The main differences are: the covariance matrix is calculated directly in the input value instead of the normal surface; and for the keypoints response they used the first eigen value of the covariance matrix around each point in a 3×3 neighborhood. The suppression process was similar to the one used in the Harris3D method. In this method, they remove the smallest eigen values by thresholding these ones.

2.4 Curvature

Surface curvature has been used extensively in the literature for cloud simplification and smoothing (Desbrun et al., 1999), object recognition (Yamany and Farag, 2002) and segmentation (Jagannathan and Miller, 2007). However, there is a lack of a systematic approach in extracting salient local features or keypoints from an input surface normals using its

local curvature information at multiple scales. The curvature method in PCL calculates the principal surface curvatures on each point using the surface normals. The keypoints response image is used to suppress weak keypoints around the stronger ones and such a process is the same as performed in the method of Harris3D.

2.5 SIFT3D

The Scale Invariant Feature Transform (SIFT) keypoint detector was proposed by Lowe (2001). The SIFT features are represented by vectors that represent local cloud measurements. The main steps used by the SIFT detector when locating keypoints are presented below.

The original algorithm for 3D data was presented by Flint et al. (2007), which uses a 3D version of the Hessian to select such interest points. A density function $f(x,y,z)$ is approximated by sampling the data regularly in space. A scale space is built over the density function, and a search is made for local maxima of the Hessian determinant.

The input cloud, $I(x,y,z)$ is convolved with a number of Gaussian filters whose standard deviations $\{\sigma_1,\sigma_2,\dots\}$ differ by a fixed scale factor. That is, $\sigma_{j+1} = k\sigma_j$ where k is a constant scalar that should be set to $\sqrt{2}$. The convolutions yield smoothed images, denoted by

$$G(x,y,z,\sigma_j), i = 1,\dots,n. \tag{4}$$

The adjacent smoothed images are then subtracted to yield a small number (3 or 4) of Difference-of-Gaussian (DoG) clouds, by

$$D(x,y,z,\sigma_j) = G(x,y,z,\sigma_{j+1}) - G(x,y,z,\sigma_j). \tag{5}$$

These two steps are repeated, yielding a number of DoG clouds over the scale space.

Once DoG clouds have been obtained, keypoints are identified as local minima/maxima of the DoG clouds across scales. This is done by comparing each point in the DoG clouds to its eight neighbors at the same scale and nine corresponding neighborhood points in each of the neighborhood scales. If the point value is the maximum or minimum among all compared points, it is selected as a candidate keypoint.

The keypoints identified from the above steps are then examined for possible elimination if the two local principal curvatures of the intensity profile around the keypoint exceed a specified threshold value. This elimination step involves estimating the ratio between the some eigenvalues of the Hessian matrix (i.e., the second partial derivatives) of the local cloud intensity around each keypoint.

2.6 SUSAN

The Smallest Univalue Segment Assimilating Nucleus (SUSAN) corner detector has been introduced by Smith and Brady (1997). Many corner detectors using various criteria for determining "cornerness" of image points are described in the literature (Smith and Brady, 1997). SUSAN is a generic low-level image processing technique, which apart from corner detection has also been used for edge detection and noise suppression.

The significance of the thresholding step with the fixed value $g = \frac{n_{max}}{2}$ (geometric threshold) is simply a precise restatement of the SUSAN principle: if the nucleus lies on a corner then the Univalue Segment Assimilating Nucleus (USAN) area will be less than half of its possible value, n_{max}. USAN is a measure of how similar a center pixel's intensity is to those in its neighborhood. The gray value similarity function $s(g_1,g_2)$ measures the similarity between the gray values g_1 and g_2. s is meant to be similar in shape to a step function

$$\begin{aligned} X_t : [0,255]^2 &\longrightarrow [0,1] \\ (g_1,g2) &\longmapsto \begin{cases} 1 \; if \; |g_1 - g_2| \leq t \\ 0 \; otherwise \end{cases} \end{aligned} \tag{6}$$

where $t \in [1,256]$ is the brightness difference threshold value. Summing over this kind of function for a set of pixels is equivalent to counting the number of similar pixels, i.e., pixels whose gray value difference is at most t. It can be used to adjust the detector's sensitivity to the image's global contrast level.

SUSAN uses the smooth gray value similarity function

$$\begin{aligned} s_t : [0,255]^2 &\longrightarrow [0,1] \\ (g_1,g2) &\longmapsto e^{-\frac{g_1-g_2}{t}} \end{aligned} \tag{7}$$

which is mentioned to perform better than the step function X_t. The smoothness of s_t plays an important role in noise suppression (Smith and Brady, 1997), since s_t only depends on the difference between g_1 and g_2.

To make the method more robust, points closer in value to the nucleus receive a higher weighting. Moreover, a set of rules presented in Smith (1992) are used to suppress qualitatively "bad" keypoints. Local minima of the SUSANs are then selected from the remaining candidates.

2.7 ISS3D

Intrinsic Shape Signatures (ISS) (Zhong, 2009) is a method relying on region-wise quality measurements.

This method uses the magnitude of the smallest eigenvalue (to include only points with large variations along each principal direction) and the ratio between two successive eigenvalues (to exclude points having similar spread along principal directions).

The ISS $S_i = \{F_i, f_i\}$ at a point p_i consists of two components: 1 – The intrinsic reference frame $F_i = \{p_i, \{e_i^x, e_i^y, e_i^z\}\}$ where p_i is the origin, and $\{e_i^x, e_i^y, e_i^z\}$ is the set of basis vectors. The intrinsic frame is a characteristic of the local object shape and independent of viewpoint. Therefore, the view independent shape features can be computed using the frame as a reference. However, its basis $\{e_i^x, e_i^y, e_i^z\}$, which specifies the vectors of its axes in the sensor coordinate system, are view dependent and directly encode the pose transform between the sensor coordinate system and the local object-oriented intrinsic frame, thus enabling fast pose calculation and view registration. 2 – The 3D shape feature vector $f_i = (f_{i0}, f_{i1}, \ldots, f_{iK-1})$, which is a view independent representation of the local/semi-local 3D shape. These features can be compared directly to facilitate the matching of surface patches or local shapes from different objects.

Only points whose ratio between two successive eigenvalues is below a threshold are considered. Among these points, the keypoints are given by the magnitude of the smallest eigenvalue, so as to consider as keypoints only those points exhibiting a large variation along every principal direction.

3 EXPERIMENTAL EVALUATION AND DISCUSSION

3.1 Dataset

To perform the evaluation of keypoint detectors, we use the large RGB-D Object Dataset[1] (Lai et al., 2011). This dataset is a hierarchical multi-view object dataset collected using an RGB-D camera. The dataset contains clouds of 300 physically distinct objects taken from multiple views, organized into 51 categories, containing a total of 207621 segmented clouds. Examples of some objects are shown in figure 2. The chosen objects are commonly found in home and office environments, where personal robots are expected to operate.

3.2 Keypoints Correspondence

The correspondence between the keypoints extracted

[1]The dataset is publicly available at http://www.cs.washington.edu/rgbd-dataset.

Figure 2: Examples of some objects of the RGB-D Object Dataset.

directly from the original cloud and the ones extracted from transformed cloud is done using the 3D point-line distance (Weisstein, 2005). A line in three dimensions can be specified by two points $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ lying on it, then a vector line is produced. The squared distance between a point on the line with parameter t and a point $p_0 = (x_0, y_0, z_0)$ is therefore

$$d^2 = \begin{array}{l}[(x_1 - x_0) + (x_2 - x_1)t]^2 + [(y_1 - y_0) + \\ (y_2 - y_1)t]^2 + [(z_1 - z_0) + (z_2 - z_1)t]^2\end{array}. \quad (8)$$

To minimize the distance, set $\partial(d^2)/\partial t = 0$ and solve for t to obtain

$$t = -\frac{(p_1 - p_0) \cdot (p_2 - p_1)}{|p_2 - p_1|^2}, \quad (9)$$

where $\cdot$ denotes the dot product. The minimum distance can then be found by plugging t back into equation 8. Using the vector quadruple product ($(A \times B)^2 = A^2B^2 - (A \cdot B)^2$) and taking the square root results, we can obtain:

$$d = \frac{|(p_0 - p_1) \times (p_0 - p_2)|}{|p_2 - p_1|}, \quad (10)$$

where $\times$ denotes the cross product. Here, the numerator is simply twice the area of the triangle formed by points p_0, p_1, and p_2, and the denominator is the length of one of the bases of the triangle.

3.3 Measures

The most important feature of a keypoint detector is its *repeatability*. This feature takes into account the capacity of the detector to find the same set of keypoints in different instances of a particular model. The differences may be due to noise, view-point change, occlusion or by a combination of the above.

The repeatability measure used in this paper is based on the measure used in (Schmid et al., 2000) for 2D keypoints and in (Salti et al., 2011) for 3D keypoints. A keypoint extracted from the model M_h, k_h^i transformed according to the rotation, translation or scale change, (R_{hl}, t_{hl}), is said to be repeatable if the distance d (given by the equation 10) from its nearest neighbor, k_l^j, in the set of keypoints extracted from the scene S_l is less than a threshold ε, $d < \varepsilon$.

We evaluate the overall repeatability of a detector both in relative and absolute terms. Given the set RK_{hl} of repeatable keypoints for an experiment involving the model-scene pair (M_h,S_l), the absolute repeatability is defined as

$$r_{abs} = |RK_{hl}| \tag{11}$$

and the relative repeatability is given by

$$r = \frac{|RK_{hl}|}{|K_{hl}|}. \tag{12}$$

The set K_{hl} is the set of all the keypoints extracted on the model M_h that are not occluded in the scene S_l. This set is estimated by aligning the keypoints extracted on M_h according to the rotation, translation and scale and then checking for the presence of keypoints in S_l in a small neighborhood of the transformed keypoints. If at least a keypoint is present in the scene in such a neighborhood, the keypoint is added to K_{hl}.

3.4 Results and Discussion

In this article, we intend to evaluate the invariance of the methods presented, in relation to rotation, translation and scale changes. For this, we vary the rotation according to the three axes (X, Y and Z). The rotations applied ranged from 5^o to 45^o, with 10^o step. The translation is performed simultaneously in the three axes and the image displacement applied on each axis is obtained randomly. Finally, we apply random variations (between $]1\times, 5\times]$) to the scale.

In table 1, we present some results about each keypoint detector applied to the original clouds. The percentage of clouds where the keypoint detectors successfully extracted (more than one keypoint) is presented in column 2. In the column 3, it appears the mean number of keypoints extracted by cloud. And finally, we present the mean computation time (in seconds) spent by each method to extract the keypoints. These times were obtained on a computer with *Intel®Core™i7-980X Extreme Edition 3.33GHz* with 24 *GB* of RAM memory.

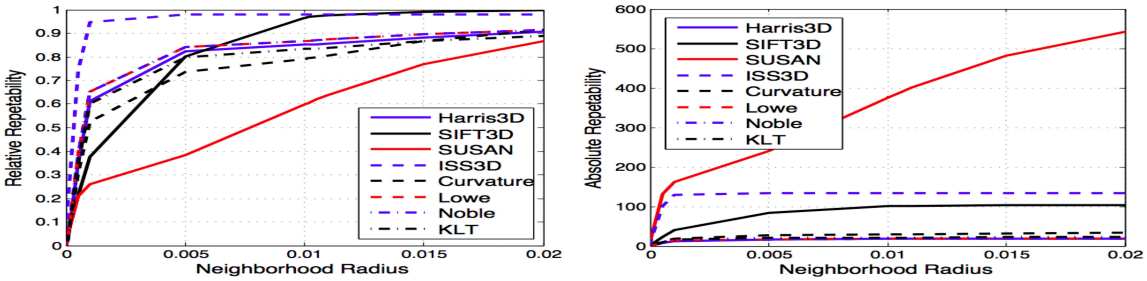
Table 1: Statistics about each keypoint detector. These values come from processing the original clouds.

Keypoint detectors	% Keypoint clouds	Mean of extracted keypoints	Mean time (s)
Harris3D	99.99	85.63	1.05
SIFT3D	99.68	87.46	9.54
ISS3D	97.97	86.24	1.07
SUSAN	86.51	242.38	1.64
Lowe	99.99	85.12	1.02
KLT	100.00	99.16	1.03
Curvature	99.96	119.36	0.70
Noble	99.99	85.12	1.04

To make a fair comparison between the descriptors, all steps in the pipeline (see figure 1) are equal. Figures 3 and 4 show the results of the evaluation of the different methods with various applied transformations. The threshold distances (ε) analyzed vary between $[0,2]$ *cm*, with small jumps in a total of 33 equally spaced distances calculated. As we saw in section 2, the methods have a relatively large set of parameters to be adjusted: the values used were the ones set by default in PCL.

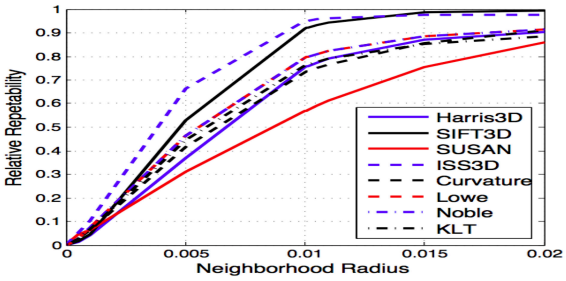
Regarding the relative repeatability (shown in figures 3(a), 3(c), 3(e), 3(g), 3(i), 4(a) and 4(c)) the methods presented have a fairly good performance in general. In relation to the rotation (see figures 3(a), 3(c), 3(e), 3(g) and 3(i)), increasing the rotation angle of the methods tends to worsen the results. Ideally, the method results should not change independently of the transformations applied. Regarding the applied rotation, the method ISS3D is the one that provides the best results. In this transformation (rotation), the biggest difference that appears between the various methods is in the 5 degrees rotation. In this case, the method ISS3D achieves almost total correspondence keypoints with a distance between them of 0.25 *cm*. Whereas for example the SIFT3D only achieves this performance for keypoints at a distance of 1 *cm*. In both the scaling and translation (shown in figures 4(a) and 4(c)), the methods exhibit very similar results to those obtained for small rotations (5^o rotation in figure 3(a)) with the exception of the SUSAN method, that has a relatively higher invariance to scale changes.

Figures 3(b), 3(d), 3(f), 3(h), 3(i), 4(b) and 4(d)

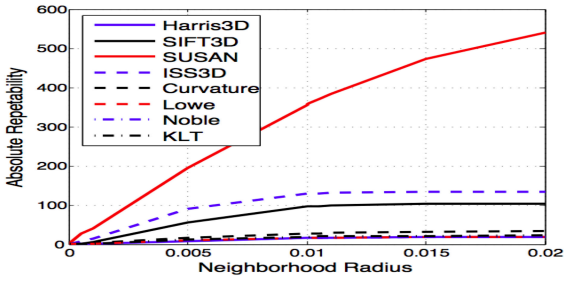


(a) Relative repeatability for 5^o rotation.

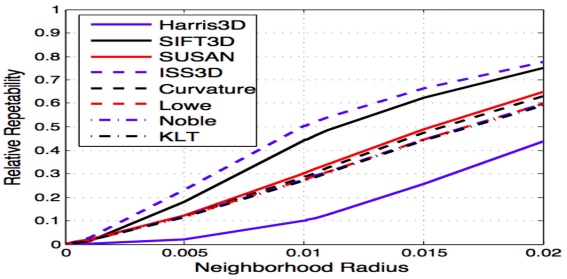
(b) Absolute repeatability for 5^o rotation.



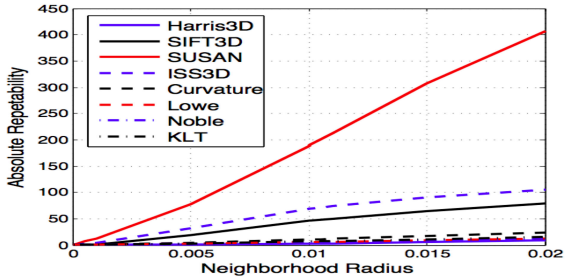
(c) Relative repeatability for 15^o rotation.



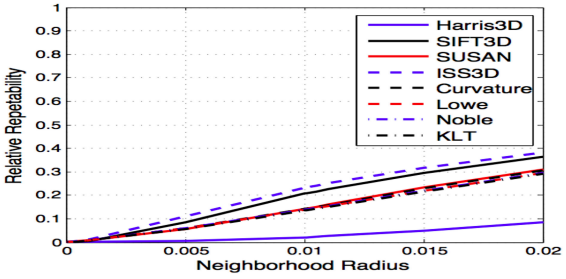
(d) Absolute repeatability for 15^o rotation.



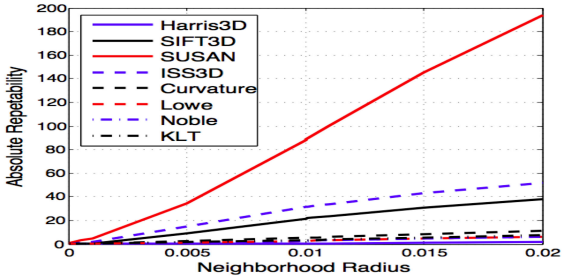
(e) Relative repeatability for 25^o rotation.



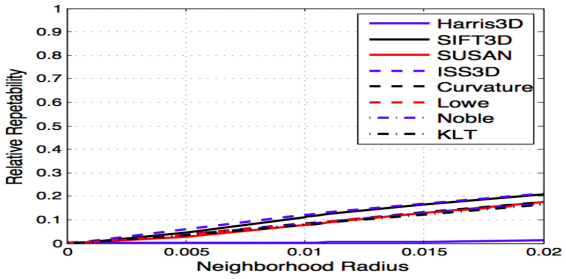
(f) Absolute repeatability for 25^o rotation.



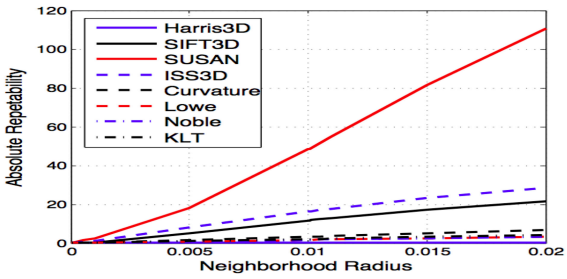
(g) Relative repeatability for 35^o rotation.



(h) Absolute repeatability for 35^o rotation.

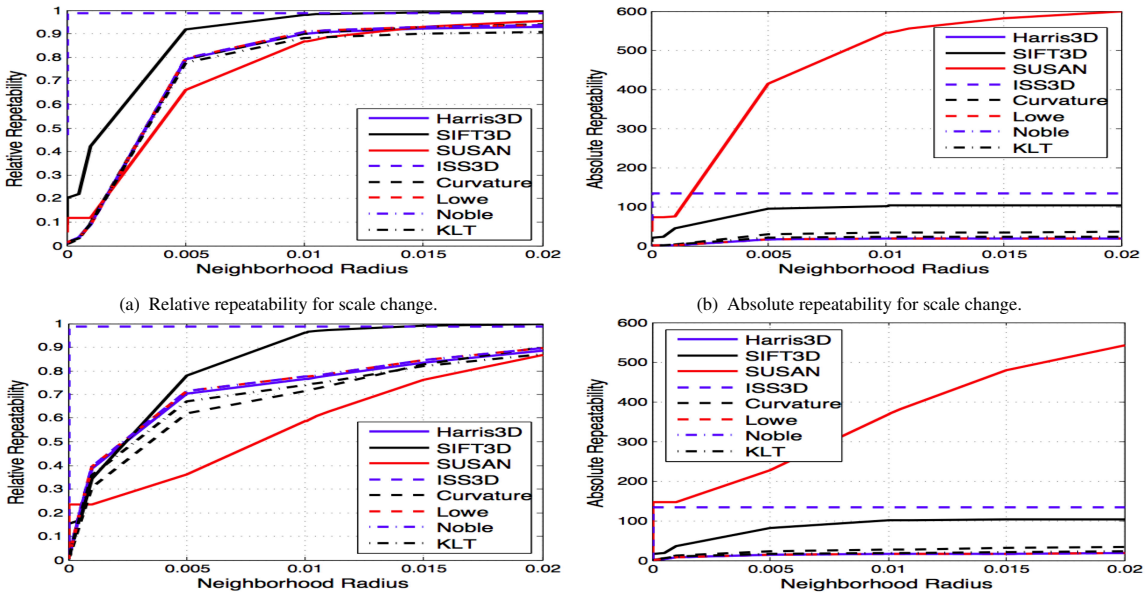


(i) Relative repeatability for 45^o rotation.



(j) Absolute repeatability for 45^o rotation.

Figure 3: Rotation results represented by the relative and absolute repeatability measures (best viewed in color). The relative repeatability is presented in figures (a), (c), (e), (g) and (i), and the absolute repeatability in figures (b), (d), (f), (h) and (j). The presented neighborhood radius is in meters.



(a) Relative repeatability for scale change.

(b) Absolute repeatability for scale change.

(c) Relative repeatability for translation cloud.

(d) Absolute repeatability for translation cloud.

Figure 4: Relative and absolute repeatability measures for the scale change and translation clouds (best viewed in color). The relative repeatability is presented in figures (a) and (c), and the absolute repeatability in figures (b) and (d). The presented neighborhood radius is in meters.

show the absolute repeatability, that present the number of keypoints obtained by the methods. With these results we can see that the method that has higher absolute repeatability (SUSAN) is not the one that shows the best performance in terms of relative repeatability. In terms of the absolute repeatability, the ISS3D and SIFT3D have better results than the SUSAN method regarding the invariance transformations evaluated in this work.

4 CONCLUSIONS

In this paper, we focused on the available keypoint detectors on the PCL library, explaining how they work, and made a comparative evaluation on public available data with real 3D objects. The experimental comparison proposed in this work has outlined aspects of state-of-the-art methods for 3D keypoint detectors. This work allowed us to evaluate the best performance in terms of various transformations (rotation, scaling and translation).

The novelty of our work compared with the work of Schmid et al. (2000) and Salti et al. (2011) is: we are using a real database instead of an artificial, the large number of point clouds and different keypoint detectors. The benefit of using a real database is that our objects have "occlusion". This type of "occlusion" is made by some kind of failure in the infrared sensor of the camera or from the segmentation method. In artificial objects this does not happen, so the keypoint methods may have better results, but our experiments reflect what can happen in real life, such as, with robot vision.

Overall, SIFT3D and ISS3D yielded the best scores in terms of repeatability and ISS3D demonstrated to be the more invariant. Future work includes extension of some methodologies proposed for the keypoint detectors to work with large rotations and occlusions, and the evaluation of the best combination of keypoint detectors/descriptors.

ACKNOWLEDGEMENTS

This work is supported by '*FCT - Fundação para a Ciência e Tecnologia*' (Portugal) through the research grant 'SFRH/BD/72575/2010', and the funding from 'FEDER - QREN - Type 4.1 - *Formação Avançada*', subsidized by the European Social Fund and by Portuguese funds through 'MCTES'.

We also acknowledge the support given by the *IT - Instituto de Telecomunicações* through 'PEst-OE/EEI/LA0008/2013'.

REFERENCES

Alexandre, L. A. (2012). 3D descriptors for object and category recognition: a comparative evaluation. In *Work-*

shop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal.

Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, USA.

Filipe, S. and Alexandre, L. A. (2013). A Comparative Evaluation of 3D Keypoint Detectors. In *9th Conference on Telecommunications, Conftele 2013*, pages 145–148, Castelo Branco, Portugal.

Flint, A., Dick, A., and Hengel, A. (2007). Thrift: Local 3D Structure Recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pages 182–188.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, Manchester.

Jagannathan, A. and Miller, E. L. (2007). Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2195–2204.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view RGB-D object dataset. In *International Conference on Robotics and Automation*, pages 1817–1824.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. In *International Conference on Computer Vision*, pages 2548–2555.

Lowe, D. (2001). Local feature view clustering for 3D object recognition. *Computer Vision and Pattern Recognition*, 1:I–682–I–688.

Mair, E., Hager, G., Burschka, D., Suppa, M., and Hirzinger, G. (2010). Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In *European Conference on Computer Vision*, pages 183–196.

Mian, A., Bennamoun, M., and Owens, R. (2010). On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. *International Journal of Computer Vision*, 89(2-3):348–361.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72.

Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation*, Shanghai, China.

Salti, S., Tombari, F., and Stefano, L. D. (2011). A Performance Evaluation of 3D Keypoint Detectors. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243.

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172.

Smith, S. M. (1992). Feature based image sequence understanding.

Smith, S. M. and Brady, J. M. (1997). SUSAN – A new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

Steder, B., Rusu, R. B., Konolige, K., and Burgard, W. (2010). NARF: 3D range image features for object recognition. In *Intelligent Robots and Systems*, Taipei, Taiwan.

Tomasi, C. and Kanade, T. (1991). Detection and Tracking of Point Features. Technical report, Carnegie Mellon University.

Weisstein, E. W. (2005). *The CRC Encyclopedia of Mathematics*. CRC Press, 3rd edition.

Yamany, S. M. and Farag, A. A. (2002). Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1105–1120.

Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3D object recognition. *International Conference on Computer Vision Workshops*, pages 689–696.