BIK-BUS: Biologically Motivated 3D Keypoint Based on Bottom-Up Saliency

Sílvio Filipe, *Student Member, IEEE*, Laurent Itti, *Member, IEEE*, and Luís A. Alexandre

***Abstract*— One of the major problems found when developing a 3D recognition system involves the choice of keypoint detector and descriptor. To help solve this problem, we present a new method for the detection of 3D keypoints on point clouds and we perform benchmarking between each pair of 3D keypoint detector and 3D descriptor to evaluate their performance on object and category recognition. These evaluations are done in a public database of real 3D objects. Our keypoint detector is inspired by the behavior and neural architecture of the primate visual system. The 3D keypoints are extracted based on a bottom-up 3D saliency map, that is, a map that encodes the saliency of objects in the visual environment. The saliency map is determined by computing conspicuity maps (a combination across different modalities) of the orientation, intensity, and color information in a bottom-up and in a purely stimulus-driven manner. These three conspicuity maps are fused into a 3D saliency map and, finally, the focus of attention (or keypoint location) is sequentially directed to the most salient points in this map. Inhibiting this location automatically allows the system to attend to the next most salient location. The main conclusions are: with a similar average number of keypoints, our 3D keypoint detector outperforms the other eight 3D keypoint detectors evaluated by achieving the best result in 32 of the evaluated metrics in the category and object recognition experiments, when the second best detector only obtained the best result in eight of these metrics. The unique drawback is the computational time, since biologically inspired 3D keypoint based on bottom-up saliency is slower than the other detectors. Given that there are big differences in terms of recognition performance, size and time requirements, the selection of the keypoint detector and descriptor has to be matched to the desired task and we give some directions to facilitate this choice.**

***Index Terms*— 3D keypoints, 3D interest points, 3D object recognition, performance evaluation.**

Manuscript received May 12, 2014; revised October 8, 2014; accepted November 5, 2014. Date of publication November 20, 2014; date of current version December 9, 2014. This work was supported in part by the Fundação para a Ciência e a Tecnologia, Portugal, under Grant SFRH/BD/72575/2010, in part by the Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal, under Grant PEst-OE/EEI/LA0008/2013, in part by the National Science Foundation under Grant CCF-1317433, in part by the FEDER-QREN-Type 4.1-Formação Avançada, in part by the European Social Fund, and in part by the National Funds through the Portuguese Ministério da Educação e Ciência. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivana Tosic.

S. Filipe and L. A. Alexandre are with the Instituto de Telecomunicações, University of Beira Interior, Covilhã 6200-001, Portugal (e-mail: sfilipe@ubi.pt; lfbaa@ubi.pt).

L. Itti is with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: itti@usc.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material contains all the graphs mentioned in the paper. The total size of the file is 1.18 MB. Contact sfilipe@ubi.pt for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2371532

I. Introduction

THE interest on using depth information on computer vision applications has been growing recently due to the decreasing prices of 3D cameras. Depth information improves object perception, as it allows for the determination of its shape or geometry.

This paper has two main focuses: the first is to present a new keypoint detector; the second an evaluation of our and the state-of-art in 3D keypoint detectors when used for object recognition. Our keypoint detector is a saliency model based on spatial attention derived from the biologically plausible architecture proposed in [1] and [2]. It uses three feature channels: color, intensity and orientation. The computational algorithm of this saliency model has been presented in [2] and it remains the basis of later models and the standard saliency benchmark in 2D images. We present the 3D version of this saliency detector and demonstrate how keypoints can be extracted from a saliency map.

The 3D keypoint detectors and descriptors that we will compare can be found in version 1.7 of the Point Cloud Library (PCL) [3]. PCL is a collection of state-of-art algorithms and tools to process 3D data. With this, we will find what is the best pair of keypoint detector/descriptor for 3D point cloud objects. This is done in order to overcome the difficulty that arises when choosing the most suitable pair of keypoint detector and descriptor for use in a particular task. We propose to answer this question using a public large RGB-D Object Dataset [4], this is composed by 300 real objects.

There are other works that make the evaluation of keypoint detectors and descriptors. In [5] and [6], the evaluation was taken with 2D keypoint detectors, and for 3D were presented in [7] and [8]. A similar work on descriptor evaluation was performed in [9] and [10], where a comparison of several 3D keypoint detectors is made in this work. In relation to the work of [5]–[8], our novelty is that we use a real object database instead of an artificial, large number of 3D point clouds, different keypoint detectors and the evaluation is done based on categories and objects recognition. In [11], we have made a repetability evaluation of the state-of-art in 3D keypoint detectors. 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.

In [9], Alexander focuses on the descriptors available in PCL, explaining how they work and made a comparative evaluation on publicly available data. It compares descriptors based on two methods for keypoint extraction: one is a keypoint detector and the second approach consists on sub-sampling the input cloud with two different sizes, using a voxelgrid with 1 and 2 *cm* leaf size. The sub-sampled points are considered keypoints. One conclusion in this work is that the increased number of keypoints improves recognition results at the expense of size and time. In our study, we will see that it is not enough, the results also depend on the keypoint location. The same author studies the accuracy of the distances both for objects and category recognition and finds that simple distances give competitive results. Our work will use the distance measure with the best accuracy presented in [10].

The paper is organized as follows: the next section presents the evaluated keypoint detectors; in Section III, we describe our keypoint detector; Section IV discusses the recognition pipeline used in this paper and the last two sections will discuss the results obtained and present the conclusions.

II. 3D Keypoints

There are several proposals for 3D keypoint detectors [11]. In that work, the invariance of 3D keypoint detectors according to rotations, scale changes and translations was evaluated. It also contains a more detailed description of the keypoint detectors presented below and we compare our proposal against these ones.

A. Harris 3D

The Harris method [12] is a corner and edge based method and these types of methods are characterized by their high-intensity changes. 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, where the covariance matrix Cov will be calculated. The *keypoints response* measured at each pixel coordinate (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 and a thresholding process is used to suppress weak keypoints around the stronger ones. The keypoint responses are positive in the corner region, negative in the edge regions, and small in flat regions [12]. If the contrast of the point cloud increases, the magnitude of the keypoint responses also increase. The flat region is specified by the $trace$ falling below some selected threshold.

In the PCL we can find two variants of the Harris3D keypoint detector: these are called Lowe [13] and Noble [14]. 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 for Noble method is given by:

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

In the case of the Lowe detector (the differences between the values of the keypoint responses in the corner regions) edge regions and planar regions tend to be closer to zero compared to those of the Noble detector. This means that there are more regions considered flat.

B. Kanade-Lucas-Tomasi

The Kanade-Lucas-Tomasi (KLT) detector [15] 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 using the intensity value instead of the surface normals; and for the keypoints response they used the first eigenvalue of the covariance matrix. Finally, the suppression process is similar to the one used in the Harris3D method.

C. Curvature

The curvature method in the PCL calculates the principal surface curvatures on each point using the surface normals. The keypoints response used to suppress weak keypoints, around the stronger ones is the same as in the Harris3D.

D. Scale Invariant Feature Transform 3D

The Scale Invariant Feature Transform (SIFT) keypoint detector was proposed by [16]. In [17], the original algorithm for 3D data is presented, which uses a 3D version of the Hessian to select the interest points. 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 adjacent clouds are subtracted to yield a small number of Difference-of-Gaussian (DoG) clouds. 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.

E. Smallest Univalue Segment Assimilating Nucleus

The Smallest Univalue Segment Assimilating Nucleus (SUSAN) corner detector was introduced in [18]. SUSAN is a generic low-level image processing technique which, apart from corner detection, has also been used for edge detection and noise suppression. A geometric threshold is applied, which is simply a precise restatement of the SUSAN principle: if the nucleus (center pixel of a circular region) lies on a corner then the Univalue Segment Assimilating Nucleus (USAN) area will be less than half of its possible value. USAN is a measure of how similar a center pixel's intensity is to those in its neighborhood. A gray value similarity function $s(g_1, g_2)$ measures the similarity between the gray values g_1 and g_2. Summing over this kind of function for a set of pixels is equivalent to counting the number of similar pixels. It can be

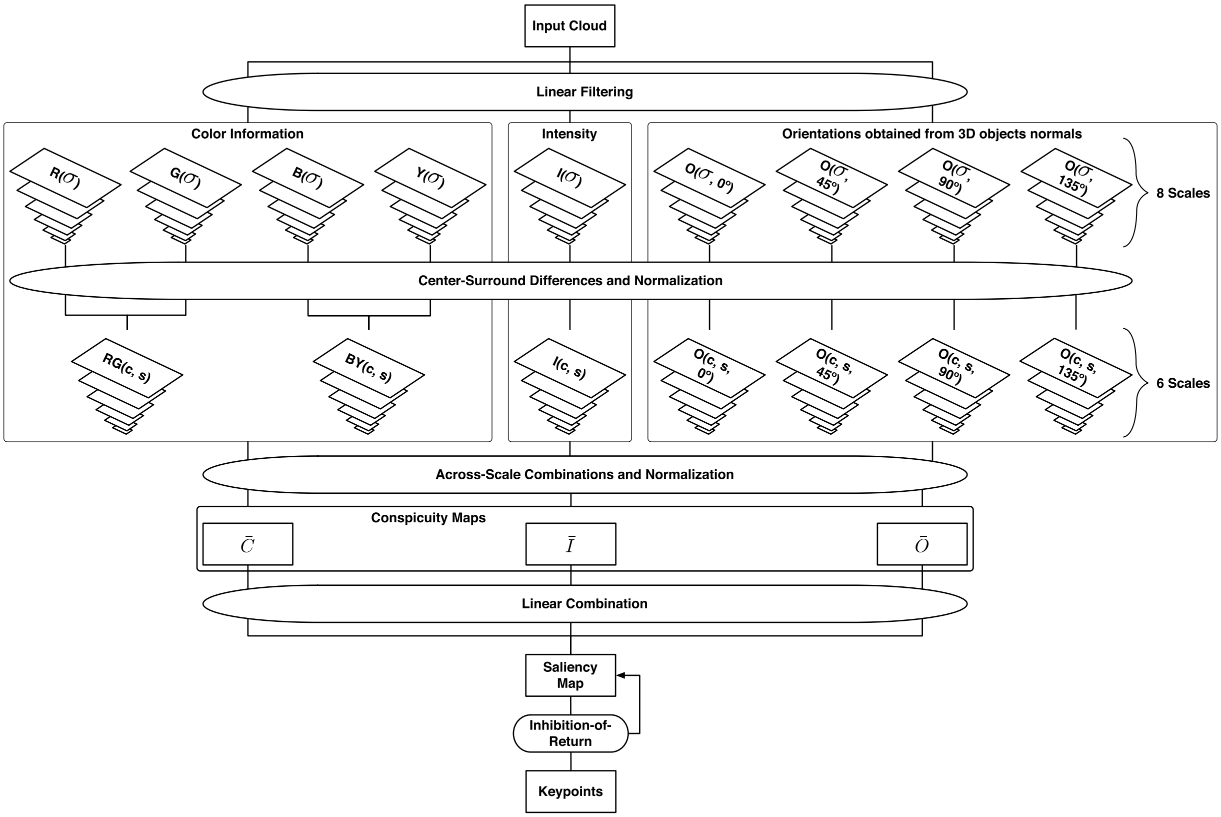


Fig. 1. General architecture of our Biologically Inspired Keypoint Detector based on Bottom-Up Saliency. Our method receives as input a point cloud similar to those shown in Figs. 3 and 4 and a linear filter is applied to obtain the color, intensity and orientations information. The full process is described in the text.

used to adjust the detector's sensitivity to the image's global contrast level. The smoothness plays of $s(g_1, g_2)$ an important role in noise suppression [18], since it 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 [19] are used to suppress qualitatively "bad" keypoints. Local minima of the SUSANs are then selected from the remaining candidates.

F. Intrinsic Shape Signatures 3D

Intrinsic Shape Signatures 3D (ISS3D) [20] 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 ISS3D $S_i = \{F_i, f_i\}$ 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.

III. Proposed 3D Keypoint Detector

The Biologically Inspired 3D Keypoint based on Bottom-Up Saliency (BIK-BUS) is a keypoint detector that is based on the saliency maps. The saliency maps are determined by computing conspicuity maps of the features intensity and orientation in a bottom-up and data-driven manner. These conspicuity maps are fused into a saliency map and, finally, the focus of attention is sequentially directed to the most salient points in this map. Using this theory and following the steps presented in [2] and [21], we will present our keypoint detector (shown in Fig. 1).

A. Linear Filtering

The color channels (r, g, and b) of the input colored point cloud are normalized when $I = (r + g + b)/3$ is larger than $1/10$ of its maximum over the entire image. Other locations yield zero r, g, and b. This is done because large areas with uniform illumination produce very weak signals, and areas with illumination changes (such as object contours) result in strong signals [2]. With these three normalized color channels, we create four broadly-tuned color channels:

$$R = r - (g + b)/2, \tag{4}$$
$$G = g - (r + b)/2, \tag{5}$$
$$B = b - (r + g)/2 \text{ and} \tag{6}$$
$$Y = (r + g)/2 - |r - g|/(2 - b), \tag{7}$$

where R is for the red channel, G for the green, B for the blue and Y for the yellow.

Gaussian pyramids [22] are used in the spatial scales, which progressively low-pass and down-sample the input cloud, producing horizontal and vertical cloud-reduction factors. Five Gaussian pyramids $R(\sigma)$, $G(\sigma)$, $B(\sigma)$, $Y(\sigma)$ and $I(\sigma)$ are created from the color and intensity channels, where σ represents the standard deviation used in the Gaussian kernel.

Each Gaussian pyramid is achieved by convolving the cloud with Gaussian kernels of increasing radius, resulting in a pyramid of clouds. We apply a similar concept to search the density map D over a range of scales, where D can be $\{R, G, B, Y, I\}$. We convolve D with a set of 3D Gaussian kernels to construct a pyramid of density maps, with each layer representing the scale σ. A factor of 2 is used to down-sample the density map and the reduction of the standard deviation of the Gaussian kernel by $\sqrt{2}$. The pyramid creation is a step similar to the DoG presented in the Section II-D.

Let $L(\cdot)$ (one of the five Gaussian pyramids) be a scale space for D:

$$L(x, y, z, \sigma) = D * g(x, y, z, \sigma), \tag{8}$$

where $*$ is the convolution operator and $g(x, y, z, \sigma)$ is a 3D Gaussian with standard deviation σ given by:

$$g(x, y, z, \sigma) = \exp\left(\frac{-x^2 - y^2 - z^2}{2\sigma^2}\right). \tag{9}$$

The orientation pyramids $O(\sigma, \theta)$ are obtained using the normals extracted from the intensity cloud I, where $\theta \in \{0°, 45°, 90°, 135°\}$ is the preferred orientation [22]. In the primary visual cortex, the impulse response of orientation-selective neurons is approximated by Gabor filters [23]. The orientation pyramids are created in a similar way to the color channels, but applying 3D Gabor filters with different orientations θ.

B. Center-Surround Differences

In the retina, bipolar and ganglion cells encode the spatial information, using center-surround structures. The center-surround structures in the retina can be described as *on-center* and *off-center*. The *on-center* use a positive weighed center and negatively weighed neighbors. The *off-center* use exactly the opposite. The positive weighing is better known as excitatory and the negative as inhibitory [24].

Similarly to the visual receptive fields, a set of linear center-surround operations is used to compute each feature. Visual neurons are most sensitive in a small region of the visual space (the center), while stimuli in the surround inhibit neuronal response [2]. Center-surround is computed as the difference between the center pixel at scale $c \in \{2, 3, 4\}$, and the surround is the corresponding pixel at scale $s = c + \delta$, with $\delta \in \{3, 4\}$. The across-scale difference between two maps (represented by '$\ominus$') is obtained by interpolation to the center scale c and point-by-point subtraction.

The first set of feature maps is concerned with intensity contrast. In mammals, this is detected by neurons sensitive either to dark centers on bright surrounds (*off-center*) or to bright centers on dark surrounds (*on-center*) [2], [23]. Here, both types of sensitivities are simultaneously computed in a set of six maps $I(c, s)$:

$$I(c, s) = |I(c) \ominus I(s)|. \tag{10}$$

For the color channels, the process is similar, which, in the cortex, is called 'color double-opponent' system [2]. In the center of their receptive fields, neurons are excited by one color and inhibited by an other, while the converse is true in the surround. The existence of a spatial and chromatic opponency between color pairs in human primary visual cortex is described in [25]. Given the chromatic opponency, the maps $RG(c, s)$ and $BY(c, s)$ are created to take in account the red/green and green/red, and blue/yellow and yellow/blue double opponency, respectively, as:

$$RG(c, s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \tag{11}$$
$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|. \tag{12}$$

Orientation feature maps, $O(c, s, \theta)$, encode, as a group, local orientation contrast between the center and surround scales:

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|. \tag{13}$$

C. Normalization

We cannot combine directly the different feature maps because they represent different dynamic ranges and extraction mechanisms. Some salient objects appear only in a few maps, which can be masked by noise or by less salient objects present in a larger number of maps. In order to resolve that, we use a map normalization operator $\mathcal{N}(.)$. This promotes the maps that contain a small number of strong activity, and suppresses the peaks in the maps that have many of them [2]. $\mathcal{N}(.)$ consists of: 1 – Large amplitude differences are eliminated by normalizing the map values to a fixed range $[0..M]$, where M is the global maximum of the map; 2 – Multiply the map by $(M - \overline{m})^2$, where $\overline{m}$ is the average of all its other local maxima. The lateral cortical inhibition is the biological motivation for this normalization [26].

D. Across-Scale Combination

The conspicuity maps are the combination of the feature maps, for intensity, color and orientation. They are obtained through the reduction of each map to scale four and point-by-point addition '$\oplus$', called across-scale addition. The conspicuity maps for the intensity, $\overline{I}$, and color channels,

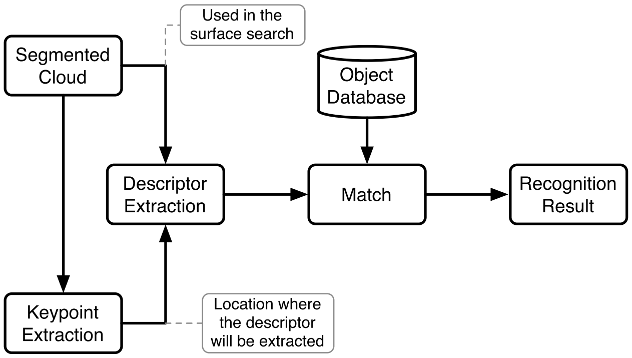


Fig. 2. Block diagram of the 3D recognition pipeline.

$\overline{C}$, are given by:

$$\overline{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(I(c,s)) \text{ and } \tag{14}$$

$$\overline{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(RG(c,s)) + \mathcal{N}(BY(c,s))]. \tag{15}$$

For orientation, we first created four intermediary maps, which are a combination of the six feature maps for a given θ. Finally, they are combined into a single orientation conspicuity map:

$$\overline{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} \mathcal{N}\left[\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(O(c,s,\theta))\right]. \tag{16}$$

The three separate channels ($\overline{I}$, $\overline{C}$ and $\overline{O}$) have an independent contribution in the saliency map and where similar features between them will have a strong impact on the saliency.

E. Linear Combination

The final saliency map is obtained by the normalization and a linear combination between them:

$$S = \frac{1}{3}\left(\mathcal{N}(\overline{I}) + \mathcal{N}(\overline{C}) + \mathcal{N}(\overline{O})\right). \tag{17}$$

F. Inhibition-of-Return

The Inhibition-of-Return (IR) is part of the method that is responsible for the selection of keypoints. It detects the most salient location and directs attention towards it, considering that location a keypoint. After that, the IR mechanism transiently suppresses this location in the saliency map and its neighborhoods in a small radius, such that attention is autonomously directed to the next most salient image location. The suppression was achieved replacing saliency map values with zero. The following iteration will find the most salient point (the maximum) in different location. This iterative process stops when the maximum of the saliency map reaches a certain value (a minimum), which is defined by a threshold. Computationally, the IR performs a similar process of selecting the global and local maximums.

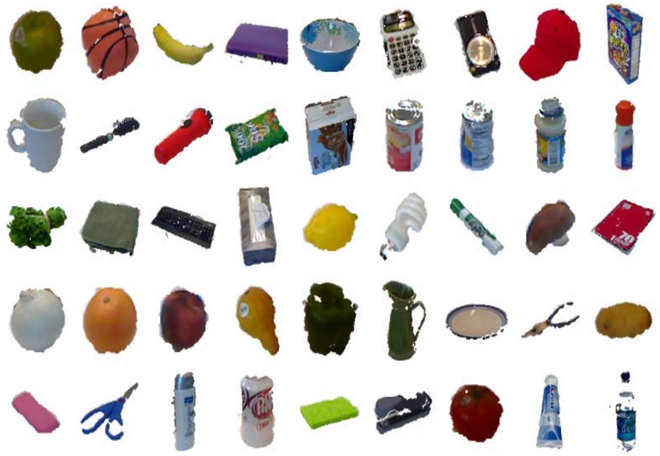


Fig. 3. Examples of point clouds from the RGB-D Object Dataset.

IV. 3D Object Recognition Pipeline

In this section, we present the pipeline used in this work, shown in Fig. 2. As input clouds, we use point clouds with the object previously segmented from the RGB-D Object Dataset [4] presented in the next sub-section. These point clouds will feed the keypoint extraction process (see more details in Section IV-B), which are used to reduce the computational cost of the recognition system. Typically, the largest computational cost of these systems is at the stage of computing the descriptors, so, it makes sense to use only a subset of the input clouds. In Fig. 2, the cloud input also feeds the descriptors extraction, but it is only used to obtain information about the keypoints neighbors (to calculate the normals at the point). A set of object descriptors is compared to those that have been previously computed and which are in the object database. The one that presents the smallest distance is considered as the corresponding object.

A. Segmented Cloud

The evaluation is done using the large RGB-D Object Dataset[1] [4]. This dataset was collected using an RGB-D camera and contains a total of 207621 segmented clouds. The dataset contains 300 physically distinct objects taken on a turntable from 4 different camera poses and the objects are organized into 51 categories. Fig. 3 presents some objects of the this dataset. It's possible to see that there are some errors in the point clouds, due to segmentation errors and sometimes depth sensor noise (some materials do not reflect the infrared pattern used to obtain depth information as well). The chosen objects are commonly found in home and office environments, where personal robots are expected to operate.

In this work, we use 5 point clouds of each physically distinct object, performing a total of 1500 point clouds selected for comparison. In Section V, we explain why we only select 1500 point clouds from this dataset.

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

TABLE I

KEYPOINTS STATISTICS. THE NUMBER OF POINTS, TIME IN SECONDS (S) AND SIZE IN KILOBYTES (KB) PRESENTED ARE RELATED TO EACH CLOUD IN THE PROCESSING OF THE TEST SET

Keypoint	Number of Points		Time (s)		Size (KB)	
Detectors	Mean±Std	Median	Mean±Std	Median	Mean±Std	Median
BIK-BUS	117.43±114.74	72.00	6.66±9.14	3.67	5.83±1.79	5.12
Curvature	116.51±125.56	68.00	0.78±1.43	0.31	5.82±1.96	5.06
Harris3D	83.61±95.64	47.00	1.11±2.01	0.46	5.31±1.49	4.73
ISS3D	84.54±107.34	43.00	1.13±2.01	0.46	5.32±1.68	4.67
KLT	96.45±109.02	54.00	1.16±2.12	0.46	5.51±1.70	4.84
Lowe	83.05±95.43	46.00	1.21±2.48	0.45	5.30±1.49	4.72
Noble	83.05±95.43	46.00	1.18±2.18	0.45	5.30±1.49	4.72
SIFT3D	85.11±103.97	43.00	2.51±3.61	1.20	5.33±1.62	4.67
SUSAN	132.52±483.05	14.00	1.54±2.74	0.64	6.07±7.55	4.22
Average	98.01±190.32	48.00	1.92±4.17	0.63	5.53±2.97	4.75
Original	5740.06±6851.42	3205			316.86±375.73	177.23

B. Keypoint Extraction

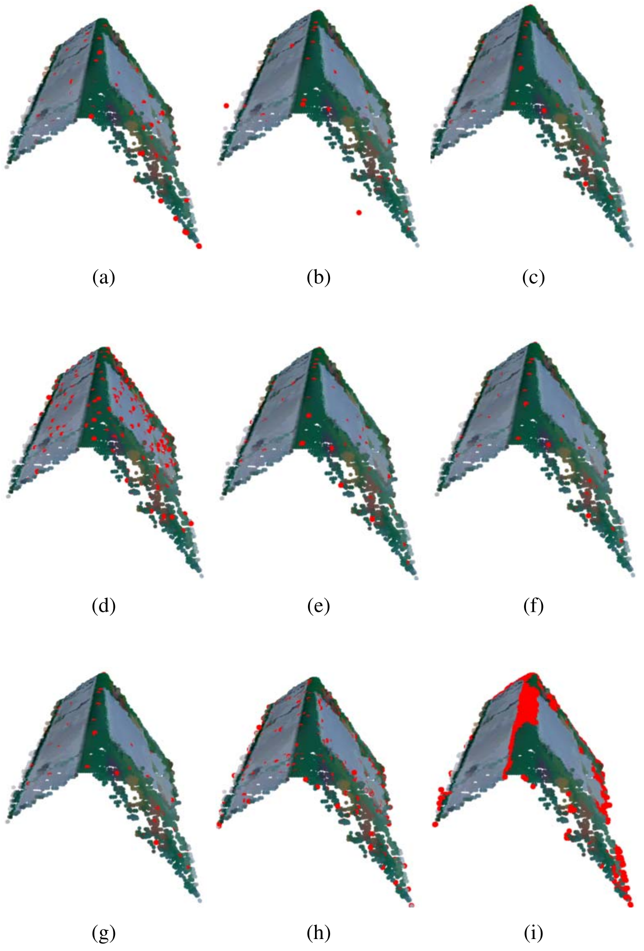
The keypoint detection methods have many parameters to adjust, but normally we use the default values in the PCL. For all the keypoint detectors, we define the same search radius to $1cm$. Where we had to set more parameters was with Susan and SIFT3D methods. For the Susan method, we define two parameters: the $distance\_threshold = 0.01$ cm is used to test if the nucleus is far enough from the centroid; and the $angular\_threshold = 0.01$ cm to verify if the normals are parallel. In the SIFT3D, we define the $min\_scale = 0.002$, $nr\_octaves = 4$, $nr\_scales\_per\_octave = 4$ and the $min\_contrast = 1$. These parameters were adjusted with these values, such that all methods present a similar average number of the keypoints (as can be seen in table I). Fig. 4 presents a cloud of points where the several keypoint detectors were applied with these parameters.

Table I also presents some statistics about the keypoints extracted from the selected point clouds. To get an idea of the reduction between the input points clouds and the keypoints, we include on the last row of the table the statistics information about the input point clouds. All the processing time was calculated based on *Intel Core I7 Extreme Edition X980* (3.3GHz), 24Gb RAM (FSB 1066) and *Fedora Core 14* operating system.

C. Descriptor Extraction

One of our goals was to evaluate the available descriptors in the current PCL version (1.7 pre-release) [3]. There are some descriptors in PCL which we will not consider in this paper, since they are not applicable to point cloud data directly or they are not object descriptors, some of them are pose descriptors (6DoF).

Table II presents some features of the evaluated descriptors and some statistics regarding the descriptors (in the same way as we did for the keypoint extraction methods). The second column contains the number of points generated by each descriptor given an input point cloud with n points. In this work the input cloud will be only the keypoints points. The third column shows the length of each point. The fourth column indicates if the descriptor requires the calculation of the surface normals at each point. In column 5, we present if the method is a global or a local descriptor. Global descriptors require the notion of the complete object


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4. Keypoint detectors applied on a "food_box" point cloud. The red points are the keypoints extracted from each detector and the number of these is presented in the legend of each sub-figure (best viewed in color). (a) BIK-BUS (103 keypoints). (b) Curvature (72 keypoints). (c) Harris3D (59 keypoints). (d) ISS3D (289 keypoints). (e) KLT (72 keypoints). (f) Lowe (59 keypoints). (g) Noble (59 keypoints). (h) SIFT3D (304 keypoints). (i) Susan (2790 keypoints).

while local descriptors are computed locally around each keypoint and work without that assumption. The sixth column indicates if the descriptor is based on the geometry or shape of the object, and if the analysis of a point is done using a sphere. The main ideas of each descriptor are presented in the following subsections.

It's only possible to make a fair comparison between the descriptors if they always use the same parameters in all steps of the pipeline, shown in Fig. 2. In the parametric configuration of the descriptors, we use the default values defined in the PCL. For the descriptors that use normals, we define a radius of $1cm$ for the calculus of normal and for the normal estimation radius search.

1) 3D Shape Context: The 3D Shape Context (3DSC) descriptor [27] is the 3D version of the SC descriptor [28]. It is based on a spherical grid centered on each keypoint. The surface normal estimation is used to orient the grid to the north pole. The grid is defined by bins along the azimuth, elevation and radial dimensions. The bins along the azimuth and elevation dimensions are equally spaced, on the other hand, the radial dimension is logarithmically spaced. The final representation of the descriptor is a 3D histogram, where in each bin contains a weighted sum of the number of points falling on the grid region. These weights are inversely proportional to the bin volume and the local point density.

2) Point Feature Histograms: Descriptors such as Point Feature Histograms (PFH) [29], Fast Point Feature Histograms (FPFH) [30], [31], Viewpoint Feature Histogram (VFH) [32], Clustered Viewpoint Feature Histogram (CVFH) [33] and Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram (OUR-CVFH) [34] can be categorized as geometry-based descriptors [35]. These type of descriptors are represented by the surface normals, curvature estimates and distances, between point pairs. The point pairs are generated by the point p and the points in its local neighborhood q. And they are represented with the angles α, ϕ and θ, which are computed based on a reference frame (u, v, w). The vector u is the surface normal at p, (n_p), v is equal to $u \times \frac{p-q}{||p-q||_2}$ and w is the cross product of these two vectors. With this reference frame, the angles can be computed using: $\alpha = v^T \cdot n_p$, $\phi = u^T \cdot \frac{p-q}{||p-q||_2}$ and $\theta = \arctan(w^t \cdot n_p, u^T \cdot n_p)$.

PFHRGB is an version of PFH in which is included information regarding the color of the object. This variant includes three more histograms, one for the ratio between each color channel of p and the same channel of q.

3) Fast Point Feature Histograms: The FPFH descriptor [30], [31] is a simplification of the PFH. In this case, the normal orientation angles are not computed for all point pairs of p and its neighborhood. The angles are computed only from its k-nearest neighbors. The estimated values are stored into a histogram, since this represents the divisions of the feature space.

4) Viewpoint Feature Histogram: In [32], they proposed an extension of FPFH descriptor, called VFH. The main differences between this and the other two descriptors above are: the surface normal is centered on the centroid c and not in the point p (n_p); instead of computing the angles using all (PFH) or k-nearest neighbors (FPFH), it uses only the centroid of the input cloud; VFH adds a viewpoint variance using the angle $\beta = \arccos(\frac{n_p \cdot c}{||c||})$, wich represents the central viewpoint vector direction translated to each normal; and it only produces one descriptor for the input cloud.

5) Clustered Viewpoint Feature Histogram: The CVFH [33] is an extension to VFH. The idea behind this descriptor is that objects which contains stable regions S. That enable them to be divided into in a certain number of disjoint regions. Stable regions are obtained by first removing the points with high curvature and then applying a smooth region growing algorithm. For each stable regions k, they find the centroid c_k and its normal (n_{c_k}) to compute a local reference frame. It is similar to the VFH descriptor, but instead of using the centroid ant its normal of the input cloud, it is only from the stable region. The final descriptor is given by the concatenated local reference frames (u, v, w, SDC, β), which is a histogram. The Shape Distribution Component (SDC) is equal to $\frac{(c-p_k)^2}{max\{(c-p_k)^2\}}, k = 1, \cdots, |S|$.

6) Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram: The OUR-CVFH [34] is a semi-global descriptor based on Semi-Global Unique Reference Frames (SGURF) and CVFH [33], which exploits the orientation provided by the reference frame to encode the geometrical properties of an object surface. For a specific surface S, it computes N triplets (c_i, n_i, RF_i) obtained from the smooth clustering and the SGURF computation. SGURF aims to solve some limitations of CVFH by defining multiple repeatable coordinate systems on S. This allows to increase the spatial descriptiveness of the descriptor and obtain the 6DoF from the alignment of the reference frames.

For the surface description, it uses an extension of CVFH in the following way: first, c_i and n_i are used to compute the first three components of CVFH and the viewpoint component as presented in [33]. The fourth component of CVFH is completely removed and instead the surface S is spatially described by means of the computed RF_i. To perform this, S is rotated and translated, so that RF_i is aligned with the x, y, z axes of the original coordinate system of S and centered in c_i. To take in account the perturbations on RF_i, an interpolation is performed by associating to each point p_k eight weights. The weights are computed by placing three 1D Gaussian functions over each axis centered at c_i, which are combined by means of weight multiplication. Finally, the weights associated with p_k are added to 8 histograms, its index in each histogram being selected as $\frac{c}{R_i}$, where R is the maximum distance between any point in S and c_i.

7) Point Pair Feature: The Point Pair Feature (PPF) descriptor [36] assumes that both the scene and the model are represented as a finite set of oriented points, where a normal is associated with each point. It describes the relative position and orientation of two oriented points which is similar to the surflet-pair feature from [30] and [37]. If you have two points p_1 and p_2 and their normals n_1 and n_2, the PPF is given by

$$PPF(p_1, p_2) = (d_2, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)), \quad (18)$$

where $\angle(a, b) \in [0, \pi]$ represents the angle between a and b and $d = p_2 - p_1$.

TABLE II

FEATURES AND STATISTICS OF THE EVALUATED DESCRIPTORS IN THIS WORK. n = Number OF POINTS IN INPUT CLOUD; p = NUMBER OF AZIMUTH BINS; m = NUMBER OF STABLE REGIONS; Y = YES; N = NO. THE TIME IN SECONDS (S) AND SIZE IN KILOBYTES (KB) PRESENTED ARE RELATED TO EACH CLOUD IN THE PROCESSING OF THE TEST SET. TO KNOW THE TOTAL TIME OR THE TOTAL SIZE SPENT BY A DATABASE OF ONE OF THIS DESCRIPTOR, WE NEED TO MULTIPLY THAT BY THE NUMBER OF CLOUDS PRESENT IN THE DATABASE

Descriptor	N. Points	Point Size	Normals	Local/Global	Category	Time (s)		Size (KB)	
						Mean±Std	Median	Mean±Std	Median
3DSC	$n \times p$	1980 + 9	Y	Local	Spherical + Shape	9181.56±21135.49	1138.86	725.28±830.02	408.02
CVFH	$m \leqslant n$	308	Y	Global	Geometry + Shape	0.32±0.47	0.17	5.26±0.30	5.20
ESF	1	640	N	Global	Shape	3.15±2.55	2.68	6.50±0.00	6.50
FPFH	n	33	Y	Local	Geometry	25.21±32.82	13.79	15.97±13.77	10.70
OUR-CVFH	1	308	Y	Global	Geometry + Shape	0.32±0.59	0.25	5.47±0.71	5.20
PCE	n	5	Y	Local	Shape	1.23±2.02	0.45	5.81±2.09	5.02
PFH	n	125	Y	Local	Geometry	4157.63±7911.99	1129.62	49.33±52.16	29.39
PFHRGB	n	250	Y	Global	Geometry	8077.11±15432.91	2188.16	95.47±104.70	55.76
PPF	n	5	Y	Global	Geometry	1.26±2.96	0.42	398.50±953.27	57.69
PPFRGB	n	8	Y	Global	Geometry	4.07±5.25	2.22	7.09±3.71	5.56
SHOT	n	352 + 9	Y	Local	Geometric + Spherical	1.60±2.06	0.94	134.85±150.65	77.33
SHOTCOLOR	n	1344 + 9	Y	Local	Geometric + Spherical	1.88±3.04	1.01	494.41±564.62	278.83
SHOTLRF	n	9	N	Local	Geometric + Spherical	0.72±0.80	0.49	7.26±3.75	5.83
USC	n	1980 + 9	N	Local	Spherical + Shape	9125.88±20892.66	1135.20	728.12±831.69	408.02
VFH	1	308	Y	Global	Geometry	0.24±0.43	0.03	5.20±0.00	5.20
Average						2362.27±10257.14	2.23	187.81±504.45	13.22

The model is represented by a set of PPF's, where similar feature vectors being grouped together. This is computed for all the pair points. The distances are sampled in d_{dist} steps and the angles in $d_{angle} = 2\pi/n_{angle}$ steps and the vectors with the same discrete representation are grouped.

An object model descriptor M can be mapped from the sampled space to the model space S. The four dimensional PPF defined at equation 18 are mapped to set A of all pairs $(m_i, m_j) \in M^2$ that define an equal feature vector.

The final local coordinates use a voting scheme, this is done in order to maximize the number of scene points that lie on the model, allowing the recovery of the global object pose. The similarities between their rotations and translations are used to obtain the pose through the voting system.

In PCL, there is also a color version, called PPFRGB. In this version, three new ratios are added, one for each color channel.

8) Signature of Histograms of Orientations: The Signature of Histograms of OrienTations (SHOT) descriptor [38] is based on a signature histograms representing topological features, that make it invariant to translation and rotation. For a given keypoint, it computes a repeatable local reference frame using the eigenvalue decomposition around it. In order to incorporate geometric information of point locations in a spherical grid. For each spherical grid bin, a a 1D histogram is obtained. This histogram is constructed by summing point counts of the angle between the normal of the keypoint and the normal of each point belonging to the spherical grid. Finally, the descriptor override all these histograms according to the local reference frame.

In [39], Tombari et al. propose two variants: one is a color version (SHOTCOLOR), where use the CIELab color space as color information; the second one (SHOTLRF), they encode only the local reference frame information, discarding the shape bins and spherical information.

9) Unique Shape Context: An upgrade of the 3DSC descriptor [27] is proposed in [40], called Unique Shape Context (USC). Tombari et al. reported that one of the problems found in 3DSC is to avoid multiple descriptions for the same keypoint, based on the need to obtain as many versions of the descriptor as the number of azimuth bins. It can cause a possible ambiguity during the successive matching and classification process. To resolve that, they proposed to define only a local reference frame (as defined in [38]) for each keypoint, such that spherical grid associated to a descriptor be directed exclusively by the two main directions in relation to the normal plane. The remaining process for obtaining USC descriptor still the same as the 3DSC.

10) Ensemble of Shape Functions: In [41], they introduced the Ensemble of Shape Functions (ESF) which is a shape function describing feature properties. This is done using the three shape functions presented in [42], that are the angle, the point distance, and the area. To compute this, they use three points randomly selected, where: two of them are used to calculate the distance; the angle is defined by two lines created from all of them; and area of the triangle formed between them. An approximation (voxel grid) of the real surface is used to separate the shape functions into more descriptive histograms. These histograms will represent the point distances, angles, areas and (on, off or both) surface.

11) Point Curvature Estimation: The Point Curvature Estimation (PCE) descriptor calculates the directions and magnitudes of principal surface curvatures (obtained using the cloud normals) on each keypoint, eigenvectors and eigenvalues respectively. For each keypoint, it will produce a descriptor with 5 values. Three values are the principal curvature, which is the eigenvector with the largest eigenvalue and the other two values are the largest and smallest eigenvalues.

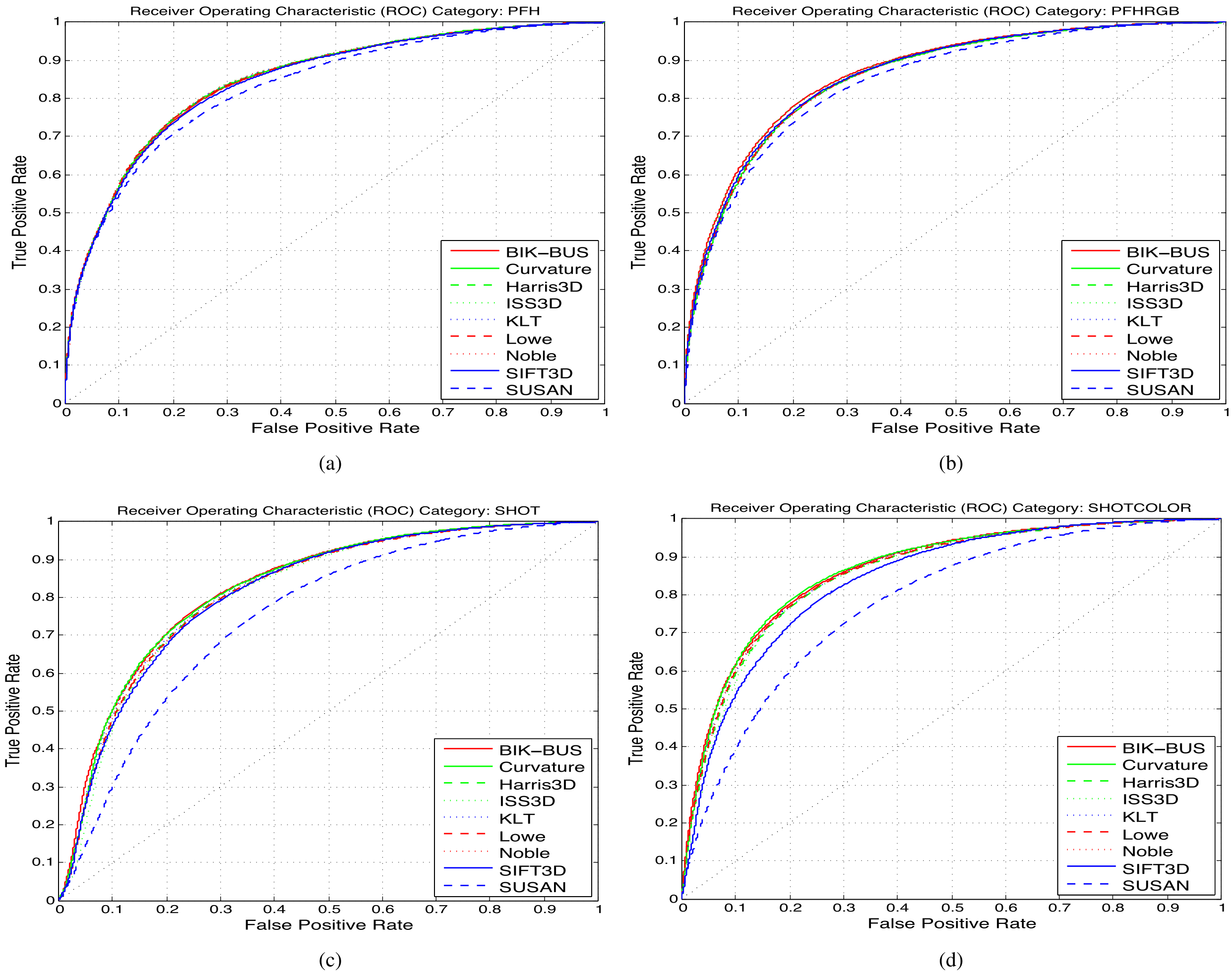
TABLE III

AUC AND DEC VALUES FOR THE CATEGORY AND OBJECT RECOGNITION FOR EACH PAIR KEYPOINT DETECTOR/DESCRIPTOR. WE ALSO PRESENT THE MEAN TIME (IN SECONDS) REQUIRED FOR THE KEYPOINTS AND DESCRIPTORS EXTRACTION. BOLD INDICATES THE BEST (BIGGER) RESULTS IN TERMS OF AUC AND DEC FOR EACH PAIR

Descriptor	Keypoint	Category		Object		Time (s)	Descriptor	Keypoint	Category		Object		Time (s)
		AUC	DEC	AUC	DEC				AUC	DEC	AUC	DEC	
3DSC	BIK-BUS	0.711	**0.519**	0.749	**0.612**	11166.25	PPF	BIK-BUS	**0.646**	**0.475**	**0.673**	**0.552**	8.01
	Curvature	**0.712**	0.491	**0.756**	0.602	10406.55		Curvature	0.555	0.016	0.579	0.008	2.09
	Harris3D	0.706	0.472	0.740	0.539	8402.61		Harris3D	0.561	0.020	0.580	0.008	1.98
	ISS3D	0.706	0.504	0.746	0.603	9581.85		ISS3D	0.640	0.405	0.667	0.479	2.00
	KLT	0.709	0.486	0.748	0.579	9208.83		KLT	0.549	0.012	0.570	0.012	2.13
	Lowe	0.705	0.468	0.746	0.560	8027.55		Lowe	0.574	0.013	0.592	0.028	2.00
	Noble	0.707	0.477	0.749	0.573	7894.47		Noble	0.576	0.021	0.592	0.007	1.92
	SIFT3D	0.700	0.511	0.727	0.568	8745.17		SIFT3D	0.641	0.434	0.666	0.510	3.51
	SUSAN	0.656	0.399	0.682	0.466	12934.55		SUSAN	0.599	0.297	0.602	0.316	11.40
CVFH	BIK-BUS	0.605	0.241	0.633	**0.286**	6.90	PPFRGB	BIK-BUS	0.493	0.042	0.506	0.077	15.12
	Curvature	0.604	**0.258**	0.633	0.283	1.02		Curvature	0.513	0.048	0.526	0.058	5.77
	Harris3D	0.606	0.249	0.632	0.256	1.33		Harris3D	0.522	0.015	0.527	0.084	5.35
	ISS3D	**0.608**	0.235	**0.637**	0.253	1.34		ISS3D	0.480	0.004	0.508	0.024	6.07
	KLT	0.606	0.252	0.633	0.270	1.38		KLT	0.501	0.103	**0.543**	0.106	5.72
	Lowe	0.606	0.248	0.634	0.253	1.42		Lowe	0.509	0.033	0.529	0.020	5.65
	Noble	0.604	0.241	0.626	0.243	1.39		Noble	0.510	**0.108**	0.480	0.066	5.09
	SIFT3D	0.594	0.170	0.635	0.255	2.73		SIFT3D	0.501	0.076	0.510	**0.201**	8.28
	SUSAN	0.560	0.020	0.573	0.038	2.00		SUSAN	**0.537**	0.003	**0.543**	0.051	17.37
ESF	BIK-BUS	0.748	0.843	0.821	1.151	9.85	SHOT	BIK-BUS	**0.827**	**1.281**	0.863	1.513	8.78
	Curvature	0.746	0.817	0.817	1.109	3.83		Curvature	0.823	1.255	**0.866**	**1.532**	2.50
	Harris3D	0.747	0.821	0.822	1.133	4.35		Harris3D	0.817	1.224	0.858	1.490	2.58
	ISS3D	0.747	0.827	0.818	1.130	4.30		ISS3D	0.812	1.168	0.852	1.413	2.69
	KLT	0.745	0.811	0.818	1.110	4.30		KLT	0.820	1.235	0.855	1.448	2.77
	Lowe	0.746	0.815	0.818	1.110	4.32		Lowe	0.818	1.229	0.855	1.462	2.66
	Noble	0.748	0.827	0.819	1.114	4.34		Noble	0.819	1.235	0.860	1.494	2.62
	SIFT3D	0.750	0.847	0.823	1.166	5.63		SIFT3D	0.814	1.207	0.848	1.409	3.94
	SUSAN	**0.751**	**0.854**	**0.826**	**1.184**	4.67		SUSAN	0.749	0.892	0.790	1.075	3.06
FPFH	BIK-BUS	**0.844**	**1.434**	**0.900**	1.833	32.67	SHOTCOLOR	BIK-BUS	**0.867**	**1.571**	**0.916**	**2.012**	9.70
	Curvature	**0.844**	1.433	0.899	1.829	25.93		Curvature	0.865	1.557	0.912	1.972	2.74
	Harris3D	0.836	1.375	0.889	1.730	26.63		Harris3D	0.858	1.519	0.906	1.918	2.72
	ISS3D	0.843	1.429	**0.900**	**1.841**	26.39		ISS3D	0.852	1.465	0.902	1.873	2.92
	KLT	0.840	1.395	0.892	1.746	26.06		KLT	0.861	1.542	0.908	1.935	2.95
	Lowe	0.839	1.391	0.892	1.752	26.00		Lowe	0.860	1.532	0.903	1.903	2.78
	Noble	0.840	1.397	0.893	1.753	26.14		Noble	0.859	1.530	0.907	1.930	2.80
	SIFT3D	0.837	1.377	0.897	1.806	27.60		SIFT3D	0.839	1.394	0.896	1.792	4.18
	SUSAN	0.809	1.236	0.864	1.575	26.04		SUSAN	0.783	1.099	0.839	1.397	3.30
OUR-CVFH	BIK-BUS	0.600	0.222	0.629	**0.274**	6.91	SHOTLRF	BIK-BUS	0.789	**1.096**	**0.822**	**1.265**	7.45
	Curvature	0.605	**0.254**	0.626	0.253	1.04		Curvature	**0.790**	1.062	0.814	1.188	1.54
	Harris3D	0.604	0.233	0.636	0.262	1.33		Harris3D	0.784	1.013	0.810	1.138	1.80
	ISS3D	**0.606**	0.224	**0.635**	0.241	1.36		ISS3D	0.788	1.003	0.817	1.139	1.86
	KLT	**0.606**	0.248	0.634	0.265	1.41		KLT	0.785	1.003	0.815	1.154	1.87
	Lowe	0.604	0.236	0.634	0.264	1.44		Lowe	0.784	1.017	0.812	1.146	1.87
	Noble	0.602	0.225	**0.635**	0.271	1.39		Noble	0.785	1.021	0.811	1.142	1.88
	SIFT3D	0.593	0.159	0.626	0.218	2.73		SIFT3D	0.770	0.924	0.805	1.086	3.20
	SUSAN	0.556	0.009	0.571	0.035	1.89		SUSAN	0.676	0.561	0.710	0.684	2.24
PCE	BIK-BUS	0.614	0.393	0.639	0.470	8.01	USC	BIK-BUS	**0.739**	**0.651**	**0.789**	**0.812**	11041.82
	Curvature	0.618	0.407	0.636	0.460	2.06		Curvature	0.736	0.631	0.786	0.778	10147.67
	Harris3D	0.619	0.411	0.639	0.474	2.18		Harris3D	0.728	0.599	0.778	0.743	8380.70
	ISS3D	0.623	0.427	0.645	0.495	2.28		ISS3D	0.727	0.630	0.777	0.790	9556.12
	KLT	**0.625**	**0.432**	0.646	0.503	2.28		KLT	0.731	0.609	0.784	0.774	9173.02
	Lowe	0.621	0.420	0.642	0.485	2.21		Lowe	0.729	0.604	0.781	0.765	7987.03
	Noble	0.621	0.419	**0.647**	**0.508**	2.23		Noble	0.727	0.597	0.777	0.740	7970.97
	SIFT3D	0.619	0.412	0.640	0.479	3.58		SIFT3D	0.727	0.647	0.774	0.797	8725.92
	SUSAN	0.596	0.336	0.618	0.412	2.80		SUSAN	0.681	0.506	0.717	0.623	11458.16
PFH	BIK-BUS	0.848	1.488	0.893	1.832	4948.23	VFH	BIK-BUS	**0.647**	**0.517**	**0.705**	**0.745**	6.82
	Curvature	0.848	1.489	0.893	1.831	4816.54		Curvature	0.644	0.502	0.703	0.732	0.94
	Harris3D	**0.849**	**1.491**	0.894	1.843	3722.78		Harris3D	0.638	0.483	0.680	0.638	1.27
	ISS3D	0.848	1.489	0.895	**1.855**	4367.38		ISS3D	0.643	0.514	0.687	0.671	1.28
	KLT	0.848	1.489	0.891	1.811	4202.21		KLT	0.644	0.507	0.691	0.680	1.32
	Lowe	0.847	1.483	**0.896**	1.854	3626.09		Lowe	0.638	0.481	0.687	0.670	1.37
	Noble	0.846	1.474	0.894	1.840	3651.77		Noble	0.638	0.480	0.682	0.649	1.32
	SIFT3D	0.843	1.458	0.890	1.801	3920.08		SIFT3D	0.636	0.469	0.686	0.651	2.66
	SUSAN	0.828	1.363	0.866	1.625	6642.93		SUSAN	0.584	0.295	0.615	0.404	1.70
PFHRGB	BIK-BUS	**0.867**	**1.586**	**0.948**	**2.397**	9567.81							
	Curvature	0.859	1.535	0.938	2.267	9315.20							
	Harris3D	0.859	1.533	0.941	2.303	7233.37							
	ISS3D	0.866	1.585	**0.948**	2.394	8488.44							
	KLT	0.859	1.536	0.941	2.302	8206.08							
	Lowe	0.860	1.539	0.942	2.314	7047.42							
	Noble	0.861	1.548	0.939	2.275	7116.42							
	SIFT3D	0.861	1.546	0.946	2.373	7628.79							
	SUSAN	0.845	1.445	0.934	2.205	12815.49							

D. Object Database

Using the 1500 point clouds selected, the experiments use the Leave-One-Out Cross-Validation (LOOCV) method [43]. As the name suggests, LOOCV involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as



(a)

(b)

(c)

(d)

Fig. 5. ROCs for the category recognition experiments (best viewed in color). (a) PFH. (b) PFHRGB. (c) SHOT. (d) SHOTCOLOR.

the validation data. This is the same as a K-fold cross-validation with K being equal to the number of observations in the original sampling. With 1500 point clouds and LOOCV method, we perform more than 1600000 comparisons for each pair keypoint detector/descriptor and we have a total of 135 pairs (9 keypoint detectors × 15 descriptors).

E. Distance Measure and Matching

One of the stages in 3D object recognition is the correspondence between an input cloud and a known object cloud (stored in the database). The correspondence is typically done using a distance function between the sets of descriptors. In [10], multiple distance functions were studied. In this work, we will use the distance D_6 that presents good results in terms of recognition and run time. Consider two point clouds each represented by a set of descriptors A and B, then the distance D_6 between the point clouds is given by

$$D_6 = L_1(c_A, c_B) + L_1(std_A, std_B), \tag{19}$$

where c_A and c_B are the centroids of the sets A and B, respectively, and

$$std_A(i) = \sqrt{\frac{1}{|A|-1}\sum_{j=1}^{|A|}(a_j(i) - c_A(i))^2}, i = 1, \ldots, n, \tag{20}$$

$a_j(i)$ refers to the coordinate i of the descriptor j, and likewise for std_B. The L_1 distance is between two descriptors (not sets) x, y:

$$L_1(x, y) = \sum_{i=1}^{n} |x(i) - y(i)|. \tag{21}$$

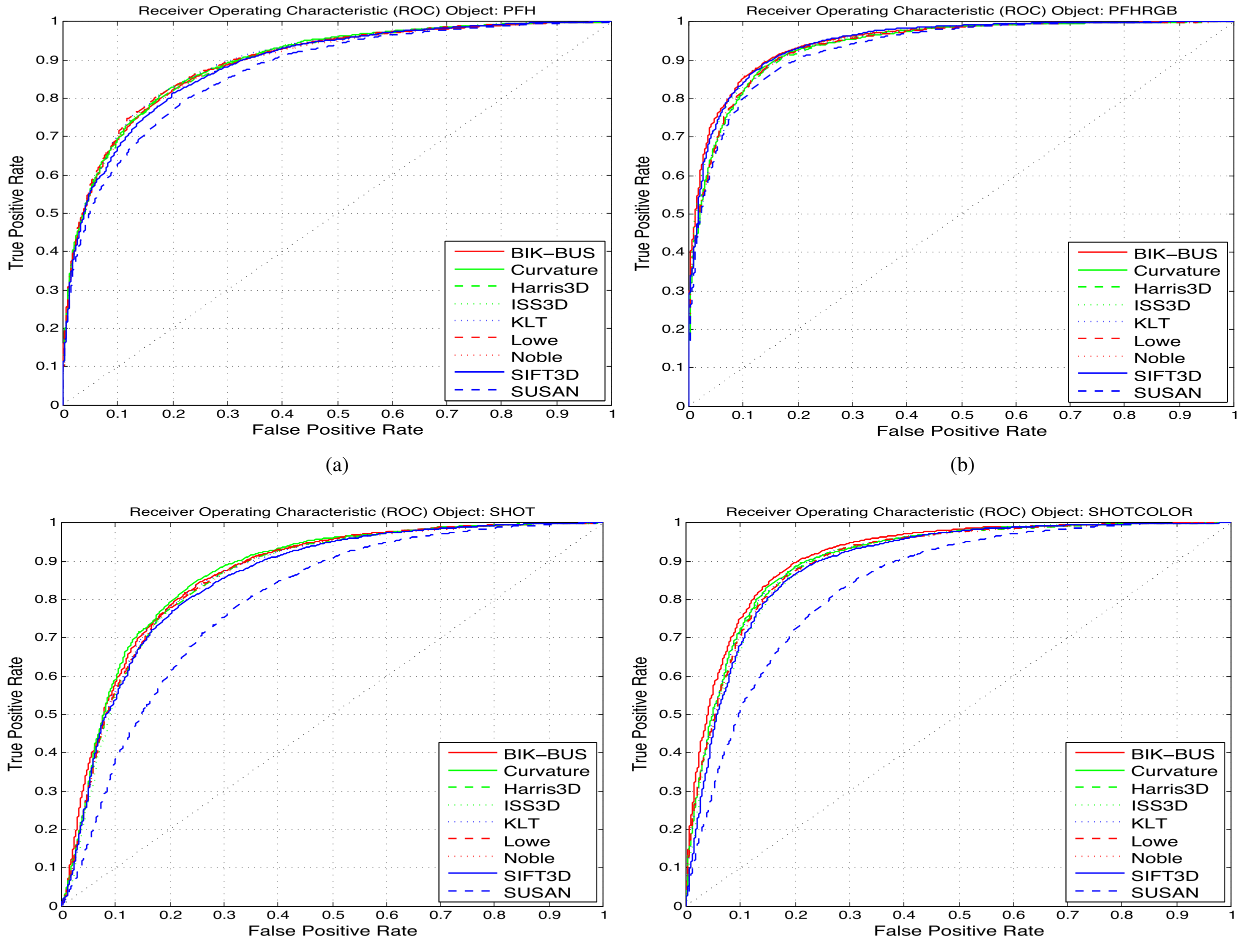
V. Experimental Evaluation and Discussion

In order to perform this evaluation, we will use three measures: the Receiver Operator Characteristic (ROC) Curve, the Area Under the ROC Curve (AUC) and the decidability (DEC). The decidability index [44] (equation 22) represents the distance between the distributions obtained for the two classical types of comparisons: between descriptors extracted from the same (*intra-class*) and different objects (*inter-class*).

$$DEC = \frac{|\mu_{intra} - \mu_{inter}|}{\sqrt{\frac{1}{2}(\sigma_{intra}^2 + \sigma_{inter}^2)}}, \tag{22}$$

where μ_{intra} and μ_{inter} denote the means of the intra- and inter-class comparisons, σ_{intra}^2 and σ_{inter}^2 the respective standard deviations and the decidability can vary between $[0, \infty[$.

The obtained AUC and DEC are given in table III, while the ROCs for category and object recognition are presented in



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

Fig. 6. ROCs for the object recognition experiments (best viewed in color). (a) PFH. (b) PFHRGB. (c) SHOT. (d) SHOTCOLOR.

Figs. 5 and 6, respectively. Table IV presents the information about the number of times that each keypoint detector achieved the best result in the category and object recognition and the sums of these counts (Total column). When there is a tie between two methods both methods score. Figs. 5 and 6 present only the four best descriptors of the table III, two use color information and the other two don't. The source code and the others ROC curves are available online.[2]

Analyzing the descriptors in a generic way, the best results were obtained with the PFHRGB. It is interesting to compare it to the PFH: improvement can only be attributed to the incorporation of color information. The same is true for the SHOTCOLOR versus the SHOT descriptor. The two best results in terms of category and object recognition are presented in the descriptors that use color information. The ROCs, in Figs. 5 and 6, also show the superiority of these two descriptors (that use color) versus the remaining. FPFH is an extension of PFH and it has a performance slightly worst than the original descriptor, but it is faster to extract and uses about half the space (shown in table II), as the authors of the descriptor suggested. An interesting result is the one obtained by PPFRGB which is an color extension of PPF: in this case the none color version is better than the color version.

TABLE IV

COUNTING THE NUMBER OF TIMES A KEYPOINT DETECTOR HAS THE BEST RESULT IN TABLE III. IN CASE OF A TIE BOTH METHODS SCORE

Keypoint	Category		Object		Total
	AUC	DEC	AUC	DEC	
BIK-BUS	7	9	7	9	32
Curvature	3	2	2	1	7
Harris3D	1	1	0	0	2
ISS3D	2	0	4	2	8
KLT	2	1	1	0	4
Lowe	0	0	1	0	1
Noble	0	1	2	1	4
SIFT3D	0	0	0	1	1
SUSAN	2	1	2	1	6

The USC was proposed as an upgrade to the 3DSC and our results confirm that in fact it improves the 3DSC results. Only when we used the SUSAN keypoint detector in both recognition tasks, the 3DSC beats the USC in most of the cases.

Considering OUR-CVFH an upgrade of CVFH and this one an extension of VFH, we are not able to see where are

[2] http://socia-lab.di.ubi.pt/&#8764;silvio/

improvements because both have lower scores and the processing times are slightly higher than the original descriptor.

In terms of computational time and space, the descriptor's requirements varies a lot. If the application needs real-time performance or when we are using embedded devices with limited resources there are some descriptors that cannot be considered.

Considering only the accuracy, the best combination for the category recognition is BIK-BUS/PFHRGB, closely followed by BIK-BUS/SHOTCOLOR, ISS3D/PFHRGB and ISS3D/SHOTCOLOR both in terms of AUC and DEC. The pairs BIK-BUS/PFHRGB and BIK-BUS/SHOTCOLOR have exactly the same AUC, the difference is in the DEC where it is slightly higher in the case of PFHRGB. BIK-BUS turns out again the best performer among detectors: FPFH, PPF, SHOT, SHOTCOLOR, USC and VFH. In relation to the 3DSC and SHOTLRF descriptors, our keypoint detector obtains the best DEC while the AUC is better when using Curvature keypoint detector in both descriptors.

If we consider a threshold for the AUC t_{AUC} and another for the DEC t_{DEC}, where $t_{AUC} = 0.8$ and $t_{DEC} = 1.0$. With these thresholds will keep only two original descriptors (PFH and SHOT) and four of its variants (FPFH, PFHRGB, SHOTCOLOR and SHOTLRF). In the case SUSAN/SHOT both thresholds fail and for SHOTLRF only the threshold t_{DEC} is satisfied in seven keypoint detectors. In these descriptors, our detector only in a single case does not have the best results in both measures, and this in the case of PFH where only has a difference of 0.1%. In the other four descriptors, the recognition accuracy varies between 2.2% and 8.4%.

In terms of object recognition, the best pair is BIK-BUS/PFHRGB, but only beats the second best combination, ISS3D/PFHRGB, because it presents a better DEC. For SHOT and SHOTCOLOR descriptors if we compare our keypoint detector with the ISS3D we obtain improvements for both of 1.5% in the case of category recognition, and 1.1% and 1.4% in object recognition, respectively. The only point against our keypoint detector is relation to the processing time, since it is approximately 6 times slower than ISS3D. The processing time can be reduce by a parallel implementation or by an implementation in GPU. The architecture of the BIK-BUS, shown in Fig. 1, shows that the parallel implementation would be a good strategy to solve this problem.

VI. Conclusions

In this paper we presented a novel 3D keypoint detector biologically motivated by the behavior and the neuronal architecture of the early primate visual system. We also made a comparative evaluation of several keypoint detectors plus descriptors on public available data with real 3D objects. The BIK-BUS is a keypoint detector on a computational technique to determine visual attention, which are also known as saliency maps. The saliency maps are determined by sets of features in a bottom-up and data-driven manner. The fusion of these sets produced the saliency map and the focus of attention is sequentially directed to the most salient points in this map, representing a keypoint location.

In the evaluation, we used the 3D keypoint detectors and the 3D descriptors available in the PCL library. The main conclusions of this paper are: 1) a descriptor that uses color information should be used instead of a similar one that uses only shape information; 2) the descriptor should be matched to the desired task, since there are differences in terms of recognition performance, size and time requirements; 3) in terms of keypoint detectors, to obtain an accurate recognition system we recommend the use of the BIK-BUS, since its performance was better in 32 tests, in a total of 60 tests. When the second best detector only obtained the best performance 8 times (see table IV); 4) for a real-time system, the ISS3D or Curvature detectors are good choices, since they have a performance that is only surpassed by BIK-BUS and are faster; 5) in terms of descriptors, if the focus is on accuracy we recommend the use of PFHRGB and for real-time a good choice is the SHOTCOLOR because it presents a good balance between recognition performance and time complexity.

In further work, we will select a small number keypoint detectors and descriptors (those with the best results) in order to analyze which are the best pair to do the recognition of a particular category or object. We also consider a parallelization of the code or an implementation on the GPGPU in order to reduce the computational time of BIK-BUS. This parallelization is possible because of the architecture of the method, shown in Fig. 1.

References

[1] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Human Neurobiol.*, vol. 4, no. 4, pp. 219–227, Jan. 1985.

[2] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[3] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[4] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.

[5] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, 2000.

[6] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Oct. 2005.

[7] S. Salti, F. Tombari, and L. Di Stefano, "A performance evaluation of 3D keypoint detectors," in *Proc. Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss.*, 2011, pp. 236–243.

[8] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 198–220, Jul. 2013.

[9] L. A. Alexandre, "3D descriptors for object and category recognition: A comparative evaluation," in *Proc. Workshop Color-Depth Camera Fusion Robot. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1–6.

[10] L. A. Alexandre, "Set distance functions for 3D object recognition," in *Proc. 18th Iberoamer. Congr. Pattern Recognit.*, 2013, pp. 57–64.

[11] S. Filipe and L. A. Alexandre, "A comparative evaluation of 3D keypoint detectors in a RGB-D object dataset," in *Proc. 9th Int. Conf. Comput. Vis. Theory Appl.*, Jan. 2014, pp. 476–483.

[12] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–152.

[13] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Dec. 2001, pp. I-682–I-688.

[14] J. A. Noble, *Descriptions of Image Surfaces*. London, U.K.: Oxford Univ. Press, 1989.

[15] C. Tomasi and T. Kanade, "Detection and tracking of point features," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-91-132, 1991.

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

[17] A. Flint, A. Dick, and A. Van Den Hengel, "Thrift: Local 3D structure recognition," in *Proc. 9th Biennial Conf. Austral. Pattern Recognit. Soc. Digit. Image Comput. Techn. Appl.*, Dec. 2007, pp. 182–188.

[18] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, 1997.

[19] S. M. Smith, "Feature based image sequence understanding," Ph.D. thesis, Robot. Res. Group, Dept. Eng. Sci., Oxford Univ., Oxford, U.K., 1992.

[20] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, Sep. 2009, pp. 689–696.

[21] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vis. Res.*, vol. 40, nos. 10–12, pp. 1489–1506, Jan. 2000.

[22] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C. H. Anderson, "Overcomplete steerable pyramid filters and rotation invariance," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 222–228.

[23] A. G. Leventhal, *The Neural Basis of Visual Function: Vision and Visual Dysfunction*. Boca Raton, FL, USA: CRC Press, 1991.

[24] J. L. Bermúdez, *Cognitive Science: An Introduction to the Science of the Mind*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[25] S. Engel, X. Zhang, and B. Wandell, "Colour tuning in human visual cortex measured with functional magnetic resonance imaging," *Nature*, vol. 388, no. 6637, pp. 68–71, 1997.

[26] M. W. Cannon and S. C. Fullenkamp, "A model for inhibitory lateral interaction effects in perceived contrast," *Vis. Res.*, vol. 36, no. 8, pp. 1115–1125, 1996.

[27] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, pp. 224–237.

[28] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[29] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.

[30] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[31] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz, "Fast geometric point labeling using conditional random fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 7–12.

[32] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2155–2162.

[33] A. Aldoma *et al.*, "CAD-model recognition and 6DOF pose estimation using 3D cues," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 585–592.

[34] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, "OUR-CVFH—Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation," in *Proc. Joint 34th DAGM, 36th OAGM Symp.*, 2012, pp. 113–122.

[35] A. Aldoma *et al.*, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 80–91, Sep. 2012.

[36] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 998–1005.

[37] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification," in *Proc. 4th Int. Conf. 3D Digit. Imag. Modeling*, Oct. 2003, pp. 474–481.

[38] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 356–369.

[39] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 809–812.

[40] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. ACM Workshop 3D Object Retr.*, 2010, pp. 57–62.

[41] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Karon Beach, Phuket, Dec. 2011, pp. 2987–2992.

[42] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proc. Int. Conf. Shape Modeling Appl.*, May 2001, pp. 154–166.

[43] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, vol. 2. San Francisco, CA, USA, 1995, pp. 1137–1143.

[44] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

Sílvio Filipe (S'13) received the B.Sc. degree in computer science and the M.Sc. degree in computer science with a minor in computing and intelligent systems from the University of Beira Interior, Covilhã, Portugal, in 2008 and 2010, respectively. His current research interests include pattern recognition, image processing, and biometrics. He has authored research papers in journals and conferences, and is a member of the Research Laboratory with the University of Beira Interior, where he is currently pursuing the Ph.D. degree. He is also a member of the Portuguese Association for Pattern Recognition and the Institute for Systems and Technologies of Information, Control and Communication.

Laurent Itti received the M.S. degree in image processing from the École Nationale Supérieure des Télécommunications, Paris, France, in 1994, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, CA, USA, in 2000. Since then, he has been an Assistant and Associate Professor with the University of Southern California, Los Angeles, CA, USA, where he is currently a Full Professor of Computer Science, Psychology, and Neuroscience. His research interests are in biologically inspired computational vision, in particular, domains of visual attention, scene understanding, control of eye movements, and surprise. This basic research has technological applications to, among others, video compression, target detection, and robotics. He has co-authored over 150 publications in peer-reviewed journals, books and conferences, three patents, and several open-source neuromorphic vision software toolkits.

Luís A. Alexandre received the B.Sc. degree in physics/applied mathematics, the M.Sc. degree in industrial informatics, and the Ph.D. degree in electrical engineering and computers from the University of Porto, Porto, Portugal, in 1994, 1997, and 2002, respectively. His current research interests include pattern recognition, neural networks, image processing, and 3D object recognition. He has authored over 80 research papers in journals and conferences and is a Leader of the Research Laboratory with the University of Beira Interior, Covilhã, Portugal, where he is currently an Associate Professor. He is also a member of the Portuguese Association for Pattern Recognition and the International Neural Network Society, and served as a member of the Executive Committee of the European Neural Network Society from 2008 to 2010.