Contour Detection at Range Images Using Sparse Normal Detector

Alejandra Cruz-Bernal(✉), Dora-Luz Alamanza-Ojeda,
and Mario-Alberto Ibarra-Manzano

Departamento de Ingeniería Electrónica, DICIS, Universidad de Guanajuato,
Salamanca, Guanajuato, Mexico
{a.cruzbernal,dora.almanza,ibarram}@ugto.mx
http://www.dicis.ugto.mx

Abstract. The object surfaces on the Range images can be easily treated as elevations, at each point of these surfaces. The Sparse Normal Detector technique focuses in the extraction of keypoints, from homogeneous surface of Range images. Additionally, the contour of the objects in the scene can be represented through these points. First, the homogeneity feature is computed by means of the Sum and Difference Histogram technique, producing the Homogeneity image. Then, the corresponding dense normal vectors of the surface formed by this image are computed. A normal probability density function is used to select the most outstanding dense vectors, yielding the Sparse Normal descriptor. These vectors form new flat directional surfaces. The final detection of interesting point is performed using the Sparse Keypoints Detector technique. The experimental test involves a qualitative analysis, using the Middlebury and DSPLab dataset, and a quantitative evaluation of repeatability.

Keywords: Range image · Sparse Normal Detector · Keypoints detection · Contour detection

1 Introduction

A Range image provides the geometrical information (depth), not only the 2D information of a RGB image. Moreover, the feature extraction task in Range images is generally invariant to scale, rotation and illumination [1]. RGB-D images are acquired by means of a low cost 3D acquisition system, such as the Microsoft Kinect sensor [14].

On the other hand, the interesting point detection is an essential phase to develop a local feature extractor [2]. In this phase, the data, (for instance, texture in intensity images) is obtained to characterize the keypoints. Recently, Steder *et al.* [12] have introduced the Normal Aligned Radial Feature (NARF) for 3D object recognition from a Range image. Some other local detectors and descriptors for 2D or their version for 3D images are: Harris corner detector [5,6], SURF (Speed Up Robust Feature) [7], and FAST (Features from Accelerated Segment Test).

J.F. Martínez-Trinidad et al. (Eds.): MCPR 2016, LNCS 9703, pp. 115–124, 2016.
DOI: 10.1007/978-3-319-39393-3_12

The proposal presented here accomplishes a robust and balanced transformation from dense to a sparse process. First, the surface of the Homogeneity image H_m is constructed from the texture features of the Range image. Moreover, the dense normal vectors referred here as N_D of the Homogeneity image are computed. After in the *sparse process*, a Gaussian distribution is used to select a set with the more representative normal vectors (N_D) at each ***x***-, ***y***-, ***z***-direction, forming a *sparse normal vectors* referred here as $\mathbf{N}_S$. Additionally, is carried out an analysis on the neighborhood of each component of $\mathbf{N}_S$ through the *pdf* (Probability Density Function), in accordance to describe this particular region. Afterwards, the interesting points are obtained from each directional surface highlighting the contour of the objects in the scene. Experimental tests have been performed with two different datasets: (1) the benchmark Middlebury proposed by *Pal et al.* in [3] and *Hirsmüller et al.* in [8], and (2) our DSPLab dataset [9]. Finally, a comparative analysis among our proposal and different proposals for key points detection considered in the-state-of-art demonstrates a high performance at least in almost all the test.

The main contributions of this study are: first, the use of the homogeneity texture feature as a local surface descriptor applied to Range images. In particular, it is proposed to highlight the homogeneous regions because they represent the smooth curvatures of the Range image. Second, this proposal in the sparse process allows a transformation from $\mathbb{R}^3$ to $\mathbb{R}$ space, implying a significant reduction in the cardinality of the descriptor vector ($\mathbf{N}_S$ vector). These descriptors allow the representation of the scene through the separation of the forefront and the background planes. Finally, objects are defined by their keypoints highlighting their contours, with a low computational cost. In this paper, firstly, the proposed technique is described in Sects. 2 and 3. Section 4 discusses the results with a deep qualitative and quantitative analysis. Finally, brief conclusions are discussed in Sect. 5.

2 Methodology

This section describes each phase of our proposal based on the Sparse Normal Detector (SND) technique. Figure 1 illustrates a global block diagram of the proposed strategy. The Range images used are the Middlebury dataset and the Depth images acquired by a Kinect sensor, in which the income image could contain either controlled conditions or real indoor scenes. In the *dense process*, the homogeneity texture feature constructs a dense surface from the Range image, to highlight the uniform regions. Then, the normal vectors (called dense normal vectors) of the Homogeneity image are computed. Later, in the *sparse process*, $\mathbf{M}_x, \mathbf{M}_y$ and $\mathbf{M}_z$ surfaces are built up by selecting the dense normal vectors in accordance to a Gaussian distribution. This process is called Sparse Normal descriptor (SN-descriptor), which carries out a transformation from $\mathbb{R}^3$ to $\mathbb{R}$. The vectors contained in the $\mathbf{M}_x, \mathbf{M}_y$ and $\mathbf{M}_z$ surfaces are the sparse normal vectors, with which are represented the most distinctive values of these surfaces. Finally, the interesting points corresponding to each object in the scene are computed from the sparse vectors, as well as of the pdf information computed.

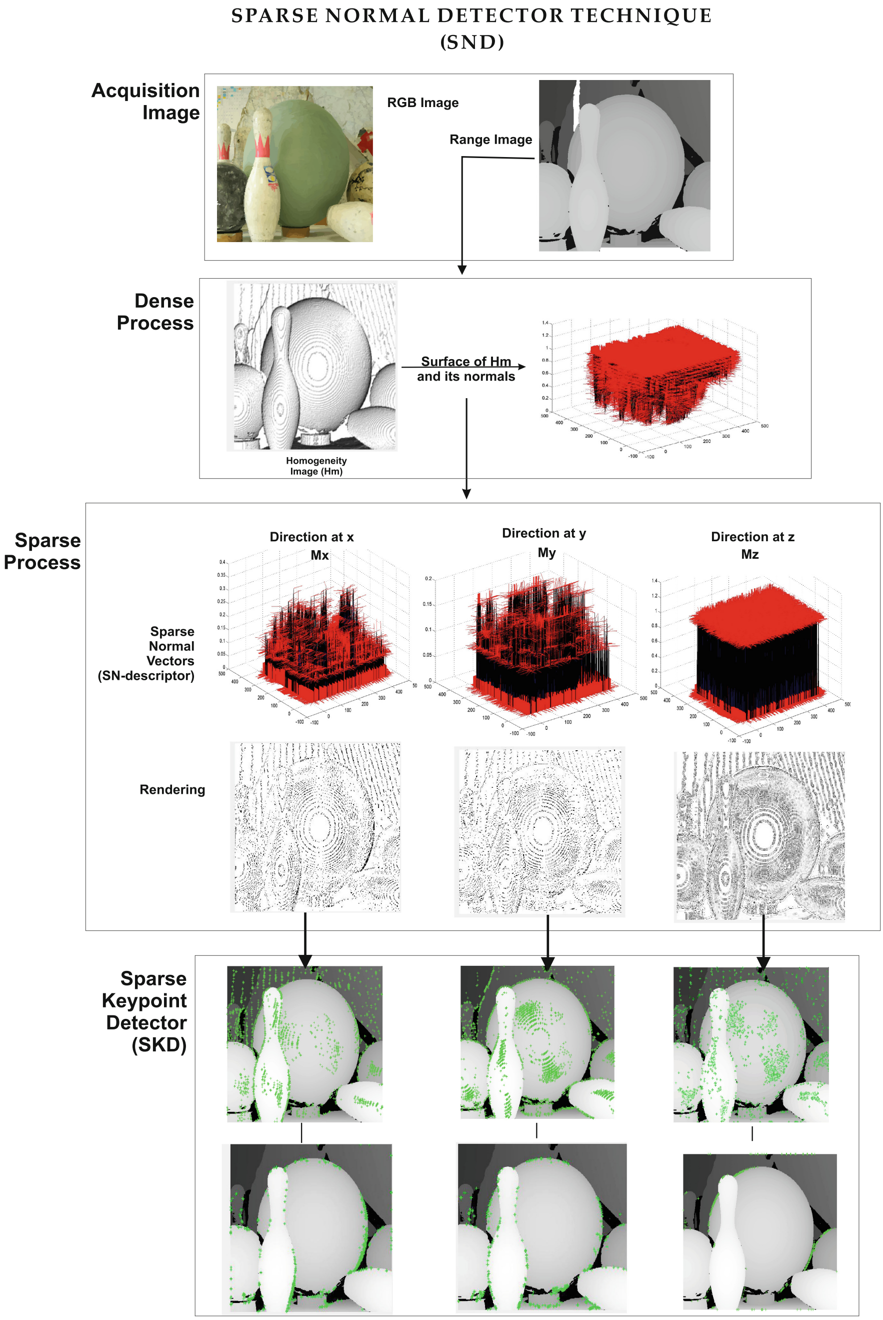


Fig. 1. Global diagram of the Sparse Normal Detector technique (SND).

2.1 Dense Porcess: The Homogeneity Image (H_m)

The Range image is a gray-level image with areas of smooth variation of intensity representing the depth of different objects. Therefore, in this paper we suggest that, the extraction of the homogeneity feature is similar to obtain a surface with a minimal change of energy. Furthermore, the change in the flow obtained from the homogeneity feature forms geodesic curves that highlight the contour and the curvature among regions [13]. In this case, the representation of the geodesic curves is similar to depict the iso-elevation lines (lines that, join of the points of equal value in height), usually used on topographic maps. On another hand, the information obtained from the Homogeneity image allows us represent objects as minimal surfaces (see Fig. 2(c)), with a Gaussian curvature established as a positive hemisphere [4,11], see Fig. 2 top row. Therefore, the extraction of normal vectors for all the points in such a hemisphere allows a dense representation of the Range image surface. These vectors are orthogonal to all tangent vectors in the H_m surface yielding the dense descriptor referred as $\mathbf{N}_D$. The H_m image contains the homogeneity texture feature of the Range image, and it is obtained using the Sum and Difference of Histograms technique (SDH) presented in [10]. A window size of 3×3 pixels is applied to the SDH, yielding an range image of $K \times L$ size and it is defined in the range 0 to 255 grey levels. This process is illustrated in the first block of the Fig. 1. Note that, the equal values of homogeneity are depicted as iso-elevation lines (see first block, Homogeneity image Fig. 1).

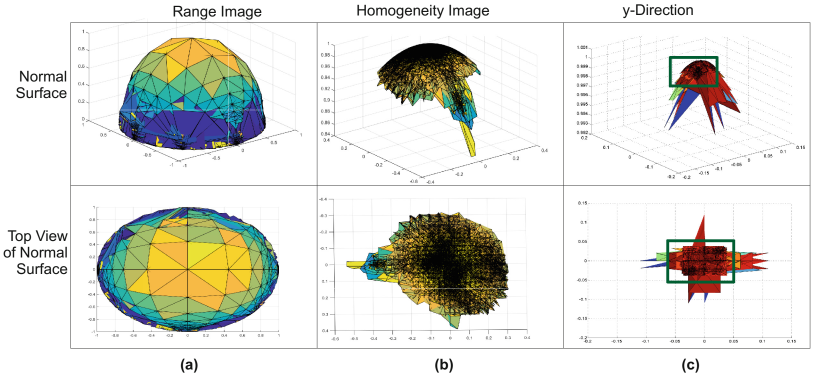


Fig. 2. Column (a) depicts the surface of the Range image as a positive hemisphere, column (b) the H_m surface is shown as a highlight of (a). The surface in **y**-direction is formed through *sparse* vectors. The labelled region is considered a smooth flat.

2.2 Sparse Process: The Sparse Normal Descriptor (SN-Descriptor)

The Sparse Normal descriptor process (referred here as SN-descriptor), divides the dense normal vectors ($\mathbf{N}_D$) into clusters with the most representative values per surface for each $\boldsymbol{x}$-, $\boldsymbol{y}$- and $\boldsymbol{z}$- direction. This process is illustrated in the first step of the Sparse process in Fig. 1. When the surface of H_m is decomposed in its normal vectors, the homogeneous surface is *"broken up"*. That is, new flat surfaces called $\mathbf{M}_x$, $\mathbf{M}_y$ and $\mathbf{M}_z$ are built up at $\mathbb{R}^2$ space. Hence, components of each directional surface are referred as *sparse* normal vectors ($\mathbf{N}_S$). Here, for every $\mathbf{N}_S$ vector on $\mathbf{M}_x$, $\mathbf{M}_y$ or $\mathbf{M}_z$ surfaces are mapping of the H_m surface to $\mathbb{R}^3$. Additionally, the neighborhood corresponding to each $\mathbf{N}_S$ vector is associated with a similar six neighbor connectivity. Then, it is carried out by an analysis of pdf in this neighborhood, to obtain the mean values with respect to intensity level (depth or disparity) and homogeneity feature. Finally, this allows to establish a descriptors set, for every one of the directional surfaces, where each component of this set is a vector of $1 \times m$ size. Therefore, descriptors are into $\mathbb{R}$ space, but, both are inmerse into $\mathbb{R}^2$ space.

2.3 Keypoint Detection: SKD Technique

The Sparse Keypoint Detector technique (referred as SKD) is shown in last block of the Fig. 1. From every one of the directional surfaces, the interesting points are extracted through of a pdf analysis. This is carried out for each neighborhood corresponding to one of the $\mathbf{N}_S$ vector, contained in the $\mathbf{M}_x$, $\mathbf{M}_y$ or $\mathbf{M}_z$ surfaces. Additionally, it is considered a 95 % confidence interval. Thereby, all sparse normal vectors form clusters in accordance to similarity values of their normals corresponding to the same neighborhood. Subsequently, these highlight vectors belong to the contour of objects in the scene.

Algorithm 1. Pseudo code of Sparse Normal Detector Technique.

```
Begin
1. Depth Image → Homogeneity Image $H_m$.
2. Compute Dense Normal Vectors $\mathbf{N}_D = [\mathbf{N}_{Dx}, \mathbf{N}_{Dy}, \mathbf{N}_{Dz}]$ from $H_m$.
3. Normal Vector ($\mathbf{N}_D$) → Sparse-Normal vector:
    a. $[\mathbf{N}_{Dx}, \mathbf{N}_{Dy}, \mathbf{N}_{Dz}] > 0$
    b. Select level of threshold $\mathbf{t}_x$ and $\mathbf{t}_y$ from $\mu(\mathbf{N}_D) \pm \sigma(\mathbf{N}_D)$, for x- and y-
       direction, respectively.
    c. Select level of threshold $\mathbf{t}_z$, from $\mu(\mathbf{N}_{Dz})$ or min ($\mathbf{N}_{Dz}$).
    d. Generate Sparse Normal vectors $[\mathbf{N}_{Dx}, \mathbf{N}_{Dy}, \mathbf{N}_{Dz}]$ using $\mathbf{t}_x$, $\mathbf{t}_y$ and $\mathbf{t}_z$.
4. Build up $\mathbf{M}_x, \mathbf{M}_y$ and $\mathbf{M}_z$ surfaces from $[\mathbf{N}_{Sx}, \mathbf{N}_{Sy}, \mathbf{N}_{Sz}]$.
5. Detect keypoints into $\mathbf{M}_x, \mathbf{M}_y$ and $\mathbf{M}_z$
    a. Select level of threshold to $\mathbf{t}_k$
    b. Sparse-Keypoints = $\mathbf{N}_S > \mathbf{t}_k$
End
```

3 Algorithm of the SND Technique

Algorithm 1 explains the computation of the SND technique. The normal vector $\mathbf{N}_D$ is computed from the H_m surface (first and second steps of the algorithm). Vector $\mathbf{N}_D = [\mathbf{N}_{Dx}, \mathbf{N}_{Dy}, \mathbf{N}_{Dz}]$ contains matrices of $K \times L$ size, which contain dense normal vectors in each one of directions. Moreover, $\mathbf{N}_D$ vector is not normalized and its values are in the range $[-1, 1]$. Here, the surface in H_m is defined as a positive hemisphere surface (see Subsect. 2.1) thus, only the outward-pointing normal vectors of $\mathbf{N}_D$ are considered (step 3(a)). This first selection is used to start the sparse process (steps 3(b-c)).

The sparse normal vectors are formed by selecting the components of $\mathbf{N}_D$ in accordance to a Gaussian distribution represented by the $\mathbf{t}$ vector (see step 3(d) of the algorithm). Here, the $\mathbf{N}_S$ vector is defined as $\mathbf{N}_S = [\mathbf{N}_{Sx}, \mathbf{N}_{Sy}, \mathbf{N}_{Sz}]$. It is built up from the $\mathbf{N}_D$ elements that are into the range $(0, 1)$. Thus, the elements of $\mathbf{N}_S$ are defined as all the normal vectors of $\mathbf{N}_D$ selected by the condition $\mathbf{N}_{Di} < \mathbf{t}$, where i is the *i-th* component of $\mathbf{N}_D$. Additionally, $\mathbf{t}$ is the vector consisting of three threshold levels in the range of $0 < \mathbf{t} < 1$, called $\mathbf{t}_x$, $\mathbf{t}_y$ and $\mathbf{t}_z$. As indicated in step 3(b) and 3(c) of the algorithm, the values of $\mathbf{t}$ are computed from the statistical information, mean and standard deviation, by analyzing each direction of vector $\mathbf{N}_D$. That is, $\mathbf{t}$ is given by

$$t_x = \begin{cases} \mu(N_{Dx}) + \sigma(N_{Dx}) \text{ if } & 0 < N_{Dx} \leq t_u \\ \mu(N_{Dx}) - \sigma(N_{Dx}) \text{ if } & t_u < N_{Dx} < 1 \end{cases} \tag{1}$$

$$t_y = \begin{cases} \mu(N_{Dy}) + \sigma(N_{Dy}) \text{ if } & 0 < N_{Dy} \leq t_u \\ \mu(N_{Dy}) - \sigma(N_{Dy}) \text{ if } & t_u < N_{Dy} < 1 \end{cases} \tag{2}$$

$$t_z = \begin{cases} \mu(N_{Dz}) & \text{if} \quad max(N_{Dz}) < 1 \\ \max(N_{Dz}) - \sigma(N_{Dz}) & \text{if} \quad max(N_{Dz}) = 1 \end{cases} \tag{3}$$

where $\mathbf{t}_u$ is a level of threshold defined within 95 % confidence interval corresponding to mean of all components vectors of $\mathbf{N}_D$. Therefore, the computation of the thresholds in Eqs. (1) and (2), is determined in accordance to value of $\mathbf{t}_u$. The $\mathbf{N}_S$ vector has enough information to represent the most significant normal components of the H_m surface, at each ***x***-, ***y***- and ***z***- direction, respectively (see fourth step). Furthermore, one of the main contributions of this study is the number of elements contained in the $\mathbf{N}_S$ vector, which is established as lower than those of vector $\mathbf{N}_D$ by at least 80 % on average, in any of its three directions. Thus,

$$\begin{array}{llllll} If & C_{Sx} = & card(N_{Sx}) & \text{and} & C_{Dx} = card(N_{Dx}) & \text{then} \quad C_{Sx} << C_{Dx} \\ If & C_{Sy} = & card(N_{Sy}) & \text{and} & C_{Dy} = card(N_{Dy}) & \text{then} \quad C_{Sy} << C_{Dy} \\ If & C_{Sz} = & card(N_{Sz}) & \text{and} & C_{Dz} = card(N_{Dz}) & \text{then} \quad C_{Sz} << C_{Dz} \end{array} \tag{4}$$

Eq. (4) presents how the cardinality of $\mathbf{N}_{Sx}$ and $\mathbf{N}_{Sy}$ is closer to 10 % with respect to the cardinality of $\mathbf{N}_{Dx}$ and $\mathbf{N}_{Dy}$, respectively, and this percentage appears in virtually all the experimental results. In $\boldsymbol{z}$-direction, the cardinality of $\mathbf{N}_{Sz}$ is closed to 30 % with respect to the cardinality of $\mathbf{N}_{Dz}$. The difference between the percentages of the sparse normal vectors in $\boldsymbol{x}$- and $\boldsymbol{y}$- directions with respect to $\boldsymbol{z}$- direction is explained through of the homogeneity level presented at the H_m image. Therefore, at each direction is possible to form a new surface that represents the original H_m surface, with a minimal number of descriptors. Finally in the fifth step, the strategy used to detect the most significant key points is the Sparse Keypoints Detector (SKD). This strategy uses the data contained on the $\mathbf{M}_x$, $\mathbf{M}_y$ and $\mathbf{M}_z$ surfaces to rank the sparse key points based on the statistically defined thresholds $\mathbf{t}_{kx}$, $\mathbf{t}_{ky}$, $\mathbf{t}_{kz}$, corresponding to each $\mathbf{x}-$, $\mathbf{y}-$, and $\mathbf{z}-$ direction, respectively. These thresholds are defined by:

$$\mathbf{t}_{kx} = \mu(N_{Sx}) + \sigma(N_{Sx}); \quad \mathbf{t}_{ky} = \mu(N_{Sy}) + \sigma(N_{Sy}); \quad \mathbf{t}_{kz} = \mu(N_{Sz}) \tag{5}$$

The last block of the Fig. 1 shows a first example of the interesting points detected at each directional surface, using the SKD technique.

4 Experimental Results

The experimental tests have been performed using two datasets: the benchmark Middlebury provided in [3,8], as well as our DSPLab dataset [9]. The importance of testing two different datasets is to establish the robustness and repeatability of our algorithm through offline images obtained under several non-controlled conditions (changes of intensity, exposure and noise in indoor environments). The most representative results respect to DSPLab dataset are depicted in Fig. 3. Also, a qualitative analysis to compare the SKD technique with respect to other common key point detectors is carried out. Finally, the effectiveness of the method is tested using a quantitative analysis of the keypoint repeatability.

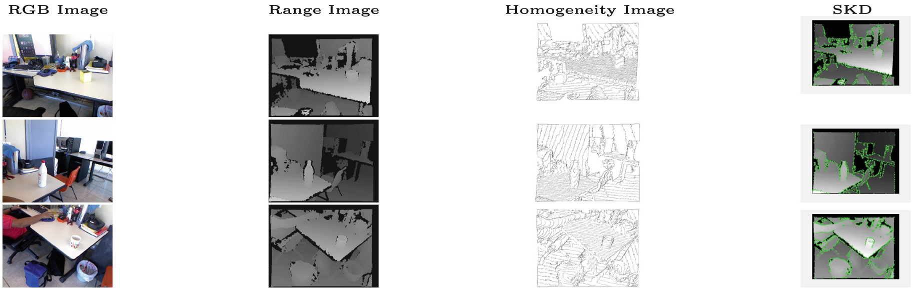
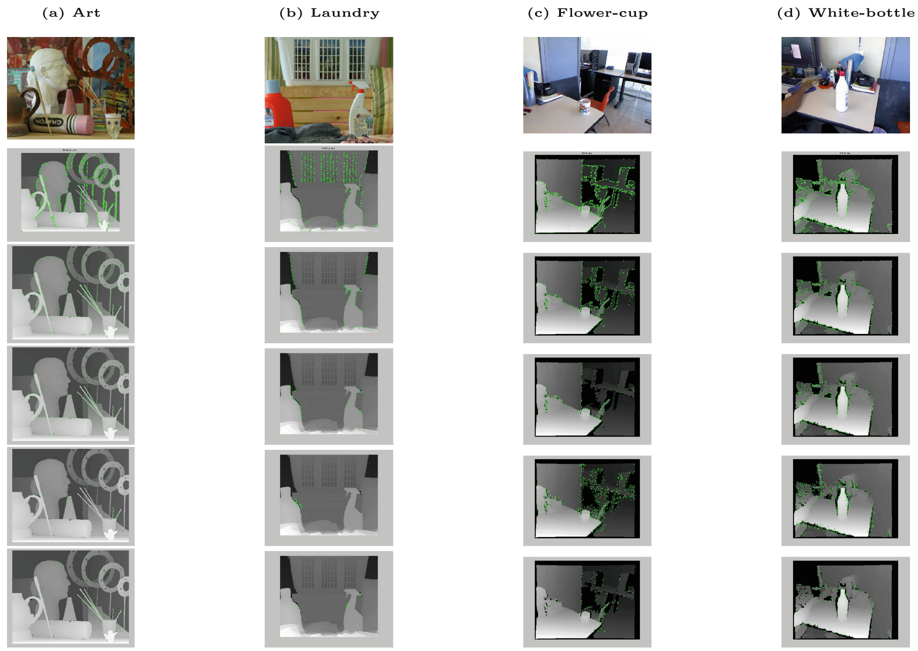


Fig. 3. Results of the homogeneity and extraction of the contour using the DSPLab database. First row, cube-rugby. Second row, white-bottle and last row, flower-cup. Notice that H_m images (third column) show a slightly loss of information. Last column shows the key points with which is defined the contour of the objects in the scene.

Figure 4 shows the most meaningful keypoints detected on the H_m surface using the approach presented in the Sect. 3. Thus, the results using the SKD technique can be compared with those of other techniques proposed in the state-of-the-art applied to range images (third to sixth column of Fig. 4). In particular, the applied techniques are NARF, Harris corner, SURF and FAST. The performance of the SKD shows a better index of repeatability than the other techniques (see Fig. 5). Qualitatively, it can be seen that the keypoints retrieve the contour of all the objects at the scene using e.g., sparse descriptors of the $\mathbf{M}_y$ surface.


(a) Art

(b) Laundry

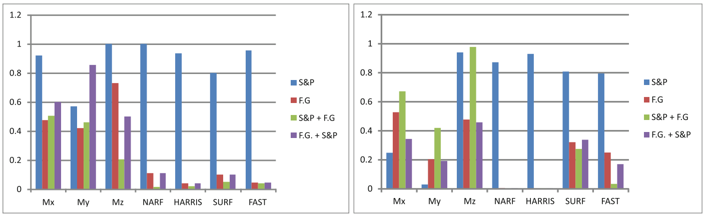
(c) Flower-cup

(d) White-bottle

Fig. 4. Comparison between the proposal Sparse Keypoint Detector (SKD) and other techniques. First row color image. Second row SKD technique. Third row NARF detector. Fourth row Harris detector. Fifth row SURF detector and sixth row FAST detector.

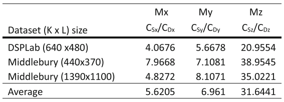
4.1 Quantitative Analysis

In order to evaluate the quality of the proposed key point detector, an analysis of its repeatability was carried out. This analysis was applied to the Middlebury and DSPLab dataset. A Gaussian Filter and salt and pepper noise (**S&P**) were used to generate synthetic and diffused images. To this purpose, the Gaussian filter was generated with a typical mask of 3×3 pixels; in addition, the average level of noise used by the **S&P** was 5 %. The test images contain four types of noises: salt and pepper noise (**S&P**), diffuse image (**D**), salt and pepper combined with diffuse image (**S&P+D**) and diffuse image combined with salt and pepper (**D+S&P**). The repeatability rate is established as the number of interesting points found in the different synthetic images under the same process with



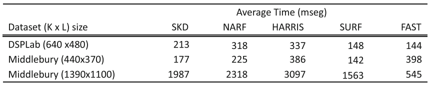
(a)

(b)



Dataset (K x L) size	Mx C_{Sx}/C_{Dx}	My C_{Sy}/C_{Dy}	Mz C_{Sz}/C_{Dz}
DSPLab (640 x480)	4.0676	5.6678	20.9554
Middlebury (440x370)	7.9668	7.1081	38.9545
Middlebury (1390x1100)	4.8272	8.1071	35.0221
Average	5.6205	6.961	31.6441

(c)



	Average Time (mseg)				
Dataset (K x L) size	SKD	NARF	HARRIS	SURF	FAST
DSPLab (640 x480)	213	318	337	148	144
Middlebury (440x370)	177	225	386	142	398
Middlebury (1390x1100)	1987	2318	3097	1563	545

(d)

Fig. 5. Comparative analysis of the keypoints detectors depicted in Fig. 4. Graph (a) illustrates the results for images Middlebury and (b) DSPLab dataset; (c) presents the relation between sparse vectors and dense vectors (see Eq. (4)), and (d) show the performance with respect to time.

respect to the total points initially detected. Thereby, each interesting point of SKD is associated once to an interesting point in the other image. These points are first validated by means of the Sum of Squared Differences technique (SSD) to avoid false positive point detection. Figure 5(a, b) show the results for the Middlebury and the DSPLab dataset, respectively. In general, the best performance of the detectors is given for the synthetic images with **S&P** noise. The worst results are obtained for the diffuse image and their different combinations. On average, the repeatability rate is near to 0.5 for the SKD technique (in particular, $\mathbf{M}_x$ and $\mathbf{M}_y$ surfaces), 0.4 for SURF and FAST techniques, and 0.2 with other techniques. In the case of crowded scenes or similar textures under controlled indoor conditions, such an images Fig. 4(a, b), the SURF and FAST techniques present a low performance (close to Zero). It is important to note that under uncontrolled environmental conditions (see images Fig. 4(c, d) NARF and Harris techniques have shown a low performance (under to 0.2 in Fig. 5(b)). Figure 5(c) presents the relation between $\mathbf{N}_S$ with respect to $\mathbf{N}_D$. For each technique in both datasets, it was obtained the average computation time of 100 executions, see Fig. 5(d). All of our data were obtained by using non-optimized Matlab implementations on an ordinary Intel(R) Core(TM)2 Duo 3.16 GHz CPU with 4 GB of RAM.

5 Conclusions

This study presented a novel approach for the contour detection by means of the interesting points from Range images. Although range images contain crowded or similar texture, the objects in the scene are separated from the background and the object contours are depicted using interesting points, with a minimal loss in the details of the scene. It is important to points out that the quantity of the descriptors is less to 6 % and 8 % of the total dense descriptors at **x-** and **y-**directions, respectively. This represents a reduction of near to 90 % in almost all tested cases. Additionally, the comparison with similar key-point detectors and the proposal presented here demonstrates that the SKD technique produces the best results in the detection of interesting points. Finally, the SND technique could be used in disciplines that imply processing of grey-level images.

References

1. Pears, N.C., Austin, J.: A machine-learning approach to keypoint detection and landmarking on 3d meshes. Int. J. Comp. Vis. **102**, 146–179 (2013)
2. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. Int. J. Comput. Vis. **102**, 128–220 (2013)
3. Pal, J.C., Weinman, J.J., Tran, L.C., Scharstein, D.: On learning conditional random fields for stereo. Int. J. Comput. Vis. **99**(3), 319–337 (2012)
4. Rodolà, E., Albarelli, A., Cremersa, D., Torsello, A.: A simple and effective relevance-based point sampling for 3d shapes. Pattern Recogn. Lett. **59**, 41–47 (2015)
5. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–151 (1988)
6. Rosten, E., Drummond, T.W.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
7. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
8. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2007, pp. 17–22. IEEE Press, Minneapolis, MN, USA (2007)
9. Hernndez-Lpez, J.J., et al.: Detecting objects using color and depth segmentation with Kinect sensor. In: Procedia Technology, editor. The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science, pp. 196–204. Elsevier Science, Mexico (2012)
10. Ibarra-Manzano, M.-A., Almanza-Ojeda, D.-L.: An FPGA implementation for texture analysis considering the real-time requirements of vision-based systems. In: Koch, A., Krishnamurthy, R., McAllister, J., Woods, R., El-Ghazawi, T. (eds.) ARC 2011. LNCS, vol. 6578, pp. 110–117. Springer, Heidelberg (2011)
11. Hedrich, J., Dietrich, P., Francois, G., Marcin, M.: Enhanced surface normal computation by exploiting RGBD sensory information. In: 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, USA, pp. 26–29 (2015)
12. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: NARF: 3D range image features for object recognition. In: International Conference on Intelligent Robots and Systems (IROS), vol. 44 (2010)
13. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press, Cambridge (2006)
14. MICROSOFT: The Kinect effect - how the world is using Kinect [Internet]. http://www.xbox.com/en-GB/kinect/kinect-effect