



# Salient Spin Images: A Descriptor for 3D Object Recognition

Jihad H'roua<sup>1</sup>(✉), Michaël Roy<sup>2</sup>, Alamin Mansouri<sup>2</sup>, Driss Mammass<sup>1</sup>,  
Patrick Juillion<sup>2</sup>, Ali Bouzit<sup>1</sup>, and Patrice Méniel<sup>3</sup>

<sup>1</sup> IRF-SIC Laboratory, Ibn Zohr University, Agadir, Morocco  
jihad.hroua@gmail.com

<sup>2</sup> LE2I, Université de Bourgogne Franche-Comté, Auxerre, France

<sup>3</sup> ARTEHIS, Université de Bourgogne Franche-Comté, Dijon, France

**Abstract.** In the last decades a wide range of algorithms have been devoted to recognize 3D free-from objects under real conditions such as occlusions, clutters, rotation, scale and translation. Spin image is one of these algorithms known to be robust to rotation, translation, occlusions up to 70% and clutters up to 60%, but still suffer from scaling, resolution changes and it is time consuming. In this paper we present a novel approach based on spin images, called salient spin images (SSI). This method enhances spin images algorithm based on its limits. Particularly, it decreases significantly the complexity of the algorithm using DoG detector, it shows a higher performance due to the relevant localization of salient vertices on the scene, and its robustness to occlusions reaches 80%.

**Keywords:** 3D object recognition · Salient vertex spin image · DoG Clutter · Occlusion · True positives

## 1 Introduction

In the recent past, the use of 3D data is becoming increasingly important which affects different domains. The increasing abundance of 3D data boosts the need for trustworthy analysis techniques, ranging from reconstruction to registration. In this work, we focus on recognition task in cluttered and occluded scenes. To this end, pattern recognition approaches are known to be the most suitable, due to their good robustness to clutters and occlusions. Pattern recognition approaches are low-level methods, they exploit local features either directly on the 3D surface of the object: 3D/3D local approaches, or first by giving a 2D representation of the object: 3D/2D local approaches, which allows the utilization of simple mathematical concepts. The state of the art puts into our reach different survey on 3D object recognition methods [5, 8]. Here we cite some of them by category.

**3D/3D Local Approaches:** Maes et al. in [10] fit SIFT descriptor [9] to 3D meshes. Similar to SIFT, MeshSIFT involves three stages: (1) Point of interest

detection using average curvature, (2) The assignment of orientation using a spherical region to compute the neighborhood, (3) Extraction of local descriptor. MeshSIFT shows its robustness to rigid and non-rigid transformations, missing data and occlusions. But still requires a uniform sampling of meshes and also it doesn't provide information about the overall shape of the object. For the same purpose, Nouri et al. [11] present a multi-scale approach to detect salient regions on the surface mesh using patches of adaptive sizes. For each vertex a patch is constructed by first estimating its tangent plane, then defining a support region on the plan. The plan is filled with projection heights of neighborhood to form thus the patch of the vertex corresponding. To define the multi-scale saliency, they compute the average of all single-scale saliencies weighted by their respective entropies. Shah et al. in [12] present a novel descriptor KSR for keypoints-based surface representation. As a first step keypoints are detected using DoG detector. Next they compute geometric distances between keypoints. The main advantages of this descriptor is its invariance to mesh resolution changes and noise. And since it doesn't extract local features around detected keypoints, the algorithm shows a low complexity.

**3D/2D Local Approaches:** Authors in [13] propose a novel 3D representation of objects from 2D images called 3DVP for 3D Voxel Pattern. It encodes 3D properties in a triplet of (appearance, 3D shape, occlusions). Using The KITTI detection benchmark [3] and 3D CAD dataset<sup>1</sup>, authors represent the appearance by the image of the object. Occlusions are coded using a 2D segmentation mask. This mask is associated with visibility labels built from a depth ordering mask, which informs either a pixel is visible, occluded or truncated. While 3D shape is represented by the voxelised 3D CAD model associated to the object. Therefor objects are recognized using a classifier such as SVM. In [14] instead of computing just one feature for a view, they adopt multiple features such as 2D Zernike moments, 2D Fourier descriptor and 2D Krawtchouk moments. Next using Hausdorff distance function, three graphs corresponding to features are generated. Then authors proposed a feature fusion framework based on multi-modal graph learning.

In this paper, a novel 3D object recognition method is proposed based on spin images [6], know to be one of the most robust descriptors to occlusions and clutters. In this approach, by the mean of the saliency concept, we enhance significantly the complexity of spin image algorithm, and its performance by increasing the number of true positives.

The paper is laid out as follows. In Sect. 2, we give a brief review of spin image algorithm, then we introduce some details about the proposed method. Experiments are conducted in Sect. 3. Finally, we conclude this paper in Sect. 4.

---

<sup>1</sup> <http://3dwarehouse.sketchup.com>.

## 2 Proposed Method

### 2.1 Background: Spin Images

Spin images is a 3D shape descriptor proposed by Johnson and Hebert in [6]. The idea behind spin image is to represent the 3D surface mesh by a set of 2D images obtained through projections of 3D vertices on local 2D coordinate systems. Each local base is determined by an oriented point  $o$  and two cylindrical coordinates  $\alpha$  and  $\beta$ . An oriented point  $o(p, \mathbf{n})$  is defined by the 3D coordinates of a vertex  $p$  on the surface of the mesh and a surface normal  $\mathbf{n}$ . The surface normal is the plane tangent to the vertex  $p$  and perpendicular to the normal vector  $\mathbf{n}$ . And  $\alpha$  and  $\beta$  are given by equation:

$$\alpha = \sqrt{\|x - p\|^2 - (\mathbf{n} \cdot (x - p))^2} \quad (1)$$

$$\beta = \mathbf{n} \cdot (x - p) \quad (2)$$

With  $x$  is other vertices to project. Thus to get a spin image for an oriented point, first all vertices of the surface mesh are projected on the local base associated to it, according to the projection function below:

$$S_O : R^3 \mapsto R^2$$

$$S_O(x) \mapsto (\alpha, \beta) = (\sqrt{\|x - p\|^2 - (\mathbf{n} \cdot (x - p))^2}, \mathbf{n} \cdot (x - p)) \quad (3)$$

The selection of vertices to project is controlled by two parameters: angles between the normal of each vertex and the normal of the oriented point, it is called angle-support, and the width  $W$  of the spin image to create. Second points  $(\alpha, \beta)$  are accumulated into discrete bins using Eq. (4), and to ensure robustness to noise a bilinear interpolation is performed to four surrounding bins, Eq. (5).

$$i = \frac{\frac{W}{2} - \beta}{b} \quad j = \frac{\alpha}{b} \quad (4)$$

$$a = \alpha - ib \quad b = \beta - jb \quad (5)$$

Figure 1 represents some spin images and their corresponding oriented points on the surface mesh of horse's skull.

Then a surface matching algorithm is implemented for 3D object recognition in distinct scenes (see Fig. 2).

### 2.2 Salient Spin Images (SSI)

As depicted in the section above, spin image descriptor is proposed by [6]. This descriptor shows its robustness to translation, rotation, occlusions (less than 70%) and clutter (less than 60%). Nevertheless it is sensitive to scale, resolution of the mesh (density) and it is time consuming. In this current work we propose a contribution to reduce the complexity of the algorithm and to improve its performance in occluded and cluttered scenes. The algorithm starts by extracting

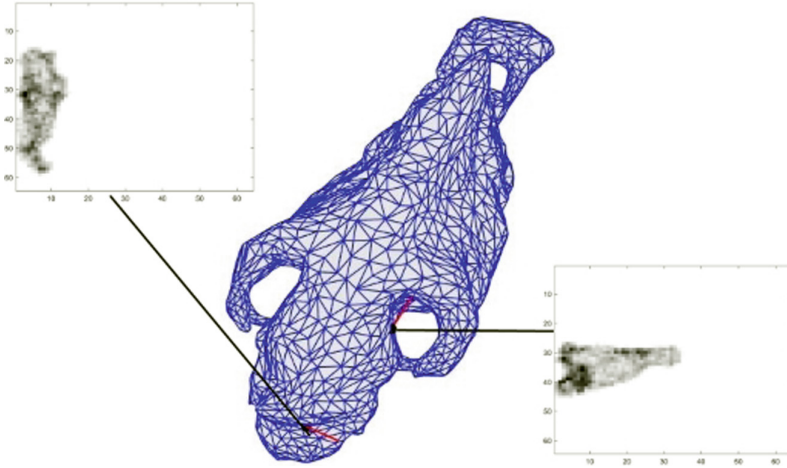


Fig. 1. Two oriented points and their corresponding spin images for skull model

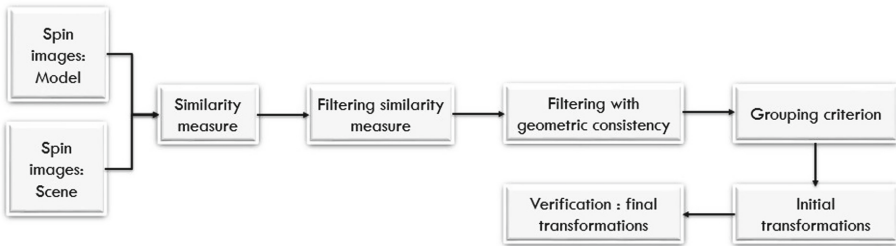
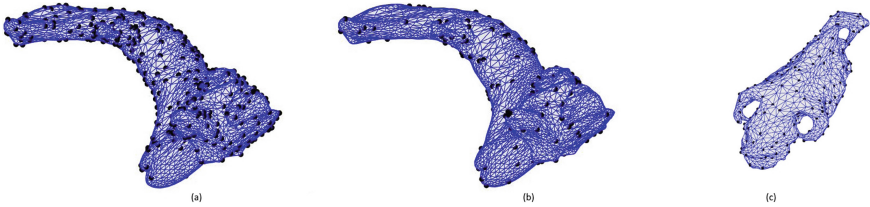


Fig. 2. Pipeline of spin images matching

spin images corresponding to every oriented point defined on each vertex  $v_i$  of the 3D mesh. Thus all vertices on the mesh are exploited to represent the object by a set of spin images with cardinality  $L = |V| = |v_i|$  equal to the number of object's vertices. In the other side, during the matching algorithm, 20% of vertices on surface mesh of the scene are randomly picked, to elaborate then a comparison between spin images of the model and those of the scene. Hence, vertices on the scene might be located sometimes in an irrelevant way, which affect the performance of the algorithm. Thereby, for the model, instead of utilizing all vertices, we propose to detect only salient ones. To do so, we use DoG detector proposed by [2]. Then each salient vertex  $v_i$  is considered as an oriented point, from which a spin image is constructed based on [6]. This modification has a direct effect on the complexity of the algorithm by reducing the number of spin images extracted from the model. For objects in our database, we notice that the number is decreased to only 10% of the number of vertices. Thus, just for the descriptor extraction phase, the complexity of computation changes from  $O(L^2)$  to  $O(L)$ . Furthermore, compared to the algorithm proposed

by Johnson and Hebert [6], during scene spin image extraction, salient vertices are always localized in the same place and covers always the surface of the object to recognize in the scene. Besides, also for the scene surface, the number of candidate vertices is reduced by around 90%. In Fig. 3, we present spreading of vertices on the surface of the scene for both spin images and SSI. As a result a huge number of correct correspondences to spin images of the model are found on the scene, which increases the chance of getting the correct transformations to align the object correctly, and accordingly the performance of the algorithm.



**Fig. 3.** Candidate vertices on some 3D surfaces for spin image extraction. (a) Salient vertices on a scene surface of four objects using DoG detector. (b) Randomly selected vertices on the same scene. (c) Salient vertices on skull model

### 3 Experimental Results

In this section we aim to evaluate experimentally the performance of the proposed approach. Therefore, we conduct a wide range of tests on both spin image algorithm [6] and our contribution, salient spin images, using models from Stanford 3D scanning repository<sup>2</sup> and our database, ArcheoZoo3D, of bones of a horse. First of all, in Sect. 3.1 we present briefly our database. Then, a description of the environment of the implementation is given in Sect. 3.2. Next, Sect. 3.3 reveals experimentation performed. Finally, we measure the precision and recall for both methods to quantify their performance, and we report the results in Sect. 3.4.

#### 3.1 Dataset

Our database was designed particularly for an archeozoology project between two laboratories: STIC laboratory (LE2I and iCUBE) and SHS laboratory (ARTEHIS). Its purpose is to meet the concrete needs of archaeozoologists who are interested in deciphering rites practiced in ancient societies, from the analysis of bone deposits: often skeletons of animals in pits. It contains 89 scans of horse's bones. For more details, readers can refer to [1] and ArcheoZoo3D<sup>3</sup>

<sup>2</sup> <http://graphics.stanford.edu/data/3Dscanrep/>.

<sup>3</sup> <https://archzoo3d.weebly.com/preacutesentation-du-projet.html>.

### 3.2 Implementation

We implemented all phases of spin image algorithm [6] in Matlab, based on the description giving in their thesis work [7]. We used the “Toolbox Graph” of Peyre<sup>4</sup> to process and display meshes. The software Meshlab and blender were used to create scenes, and to process meshes also. To compute transformations, in order to align objects, we used Horn’s et al. algorithm [4], and the implementation in Matlab proposed in<sup>5</sup>. Concerning our approach, to detect salient vertices, the DoG invariant to density proposed by [2] is used. Our experiments were carried out on a computer with 2.50 GHz Intel i7 processor, and 16 GB of memory.

### 3.3 Experimentation

To evaluate the performance of our contribution, we measure the precision and recall for both our contribution and spin image proposed by Johnson and Hebert [6]. To achieve reliable results, we need to conduct a wide range of tests, and to take into account different cases of transformations, occlusions and clutters. For this we constructed 60 scenes from four objects of ArchoeZoo3D database: caudal, ribs, femur and tarsal (see Fig. 4), and 60 scenes from 3D objects of Stanford dataset: bunny, armadillo and dragon (see Fig. 5).

We move objects randomly to get scenes with different transformations and to cover as much as possible different cases and percentages of occlusions and clutters. This process ensures a robust evaluation of the performance of the algorithm. For each object we run recognition on each of the 60 scenes. This results in 240 recognition trials for spin image algorithm and 240 recognition trials for salient spin images for each dataset separately.

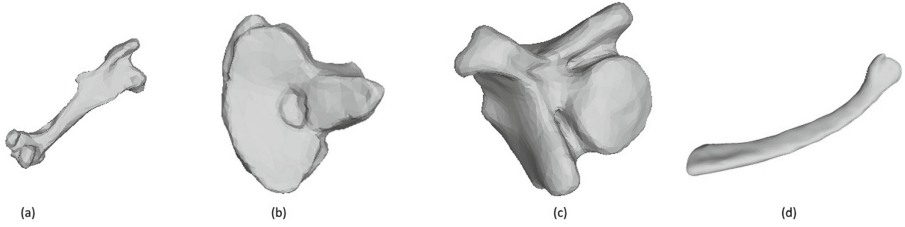
### 3.4 Evaluation

We evaluate the performance of the algorithm using precision and recall, known to be the most important measures used in the information retrieval domain. Studiously, we need first to compute true positives which means the model we are seeking to recognize exists in the scene and correctly recognized. Then, also false positives are calculated, to refer to number of times an object that does not exist in the scene, but despite that, it is recognized. Finally, we compute false negatives, when the object exist in the scene but not recognized. For false positives we used two 3D objects: Stanford bunny and Skull from our dataset (see Fig. 6).

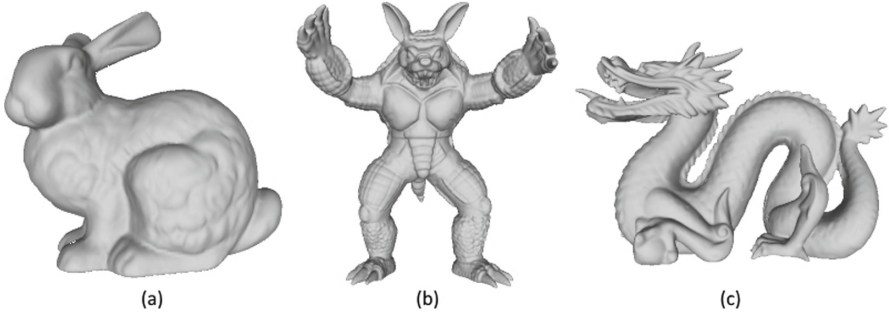
The spin image algorithm is mainly affected by occlusions and clutters. For a percentage of occlusion higher than 70% and clutters more than 60%, the recognition rate decreases, but for SSI as shown in Fig. 7, the recognition rate remains high until occlusion of around 80% (Table 1).

<sup>4</sup> <http://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph>.

<sup>5</sup> <https://fr.mathworks.com/matlabcentral/fileexchange/26186-absolute-orientation-horn-s-method>.



**Fig. 4.** Bone models: (a) Femur model. (b) tarsal model. (c) Caudal. (d) Ribs model.



**Fig. 5.** Stanford models: (a) Bunny. (b) Armadillo model. (c) Dragon.

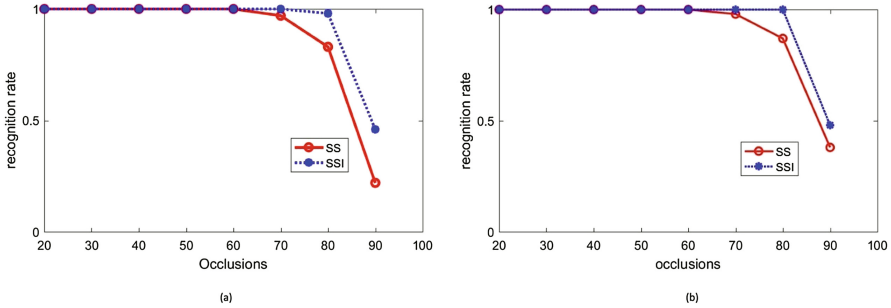
To quantify this performance we compute precision and recall for both algorithms. Table below shows that our contribution has a higher performance compared to spin image for the two data-sets Stanford and ArcheoZoo3D.

The rise in precision and recall is explained by the fact that salient vertices extracted using DoG detector are always localized in relevant places, resulting thus in significant scene spin images. Plus exploiting only salient vertices on both model and scene, helps at removing insignificant spin images and reducing the number of scene spin images that might not correspond to any model spin image (Table 2).

When it comes to the complexity, our contribution shows also better results. For example to create model spin images, instead of a complexity range of  $O(L^2)$ , using our contribution, it decreases to  $O(L)$ . This is due to the number of vertices used to create spin images. With our contribution, only salient ones are considered to be oriented points. Speaking in term of running time, using a computer with 2.50 GHz Intel i7 processor, and 16 GB of memory, for the caudal object with number of vertices equal to 1812, and a scene with 5823, and taking into account 20% of vertices to create scene spin images, we present in the table below some results.



**Fig. 6.** Skull model used to compute false positives



**Fig. 7.** Recognition rate under occlusions for spin images in red and for SSI in blue. (a) Recognition rate for Stanford dataset. (b) Recognition rate for ArcheoZoo3D. (Color figure online)

**Table 1.** Performance comparison of spin images and SSI using recall and precision.

Dataset	Method	Precision	Recall
ArcheoZoo3D	<b>Spin image</b>	0.95	0.83
	<b>SSI</b>	0.99	0.94
Stanford	<b>Spin image</b>	0.90	0.81
	<b>SSI</b>	0.97	0.90

**Table 2.** Running time comparison between spin images and SSI in seconds.

	Total running time (s)	Model spin image	Scene spin image
Spin images	686.14	10.64	23.80
SSI	77.42	2.40	14.23

## 4 Conclusion

In this work, we presented an improved version of spin images descriptor. Spin image descriptor is known to be robust to rotation, translation, occlusions under 70%, and clutters under 60%. However, it is time consuming, sensitive to resolution of the mesh and to scaling. An other problem with this approach, is it requires to know some parameters beforehand, such as the resolution of the



object. Our contribution improves the complexity by choosing only salient vertices using DoG for Difference of Gaussians. Our work has decreased significantly the complexity of the algorithm. Besides, through the relevant localization of salient vertices on the scene, the performance of the algorithm becomes better, and shows more robustness to occlusions. That being said, the uses of DoG doesn't make the spin image algorithm invariant to scale or density of the mesh, due to the number of vertices projected, which makes pixels of the images different. In our future research we intend to concentrate on making spin images multiresolution, scale invariant and also automating it, so it wouldn't require to know the resolution in advance.

## References

1. Barros, J.M.D., Habed, A., Demonceaux, C., Mansouri, A.: Computer vision-based approach for rite decryption in old societies. In: 2015 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 451–454. IEEE (2015)
2. Darom, T., Keller, Y.: Scale-invariant features for 3-D mesh models. *IEEE Trans. Image Process.* **21**(5), 2758–2769 (2012)
3. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361. IEEE (2012)
4. Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. *JOSA A* **4**(4), 629–642 (1987)
5. H'roua, J., Bekkari, A., Mammass, D., Bouzit, A., Mansouri, A., Roy, M., Le Goïc, G.: 3D objects descriptors methods: overview and trends. In: 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1–9. IEEE (2017)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (1999)
7. Johnson, A.E.: Spin-images: a representation for 3-D surface matching. Ph.D. thesis, Carnegie Mellon University (1997)
8. Loncomilla, P., Ruiz-del Solar, J., Martínez, L.: Object recognition using local invariant features for robotic applications: a survey. *Pattern Recognit.* **60**, 499–514 (2016)
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
10. Maes, C., Fabry, T., Keustermans, J., Smeets, D., Suetens, P., Vandermeulen, D.: Feature detection on 3D face surfaces for pose normalisation and recognition. In: 2010 Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), pp. 1–6. IEEE (2010)
11. Nouri, A., Charrier, C., Lézoray, O.: Multi-scale saliency of 3D colored meshes. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 2820–2824. IEEE (2015)
12. Shah, S.A.A., Bennamoun, M., Boussaid, F.: Keypoints-based surface representation for 3D modeling and 3D object recognition. *Pattern Recognit.* **64**, 29–38 (2017)

13. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Data-driven 3D voxel patterns for object category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1903–1911 (2015)
14. Zhao, S., Yao, H., Zhang, Y., Wang, Y., Liu, S.: View-based 3D object retrieval via multi-modal graph learning. *Signal Process.* **112**, 110–118 (2015)