Salient Point SUSAN 3D operator for triangles meshes

N. Walter, O. Laligant, O. Aubreton

Le2i Laboratory, 12 rue de la fonderie, Le Creusot, FRANCE,
{Nicolas.Walter | Olivier.Aubreton | o.laligant}@u-bourgogne.fr

ABSTRACT

This paper will deals with edge detection and point characterization of 3D objects. Nowadays, with developments of 3D acquisition methods for geometry measurements, salient point detection operators become a crucial part of quality control on 3D triangles meshes. First, we present a new operator that characterize and make difference between a salient, flat or cavity point of each point of the inspected object. This new operator require several steps before applying that we will describe. We finally show results on an icosahedron to improve the operator, a flint artifacts that archaeologists needs accurate geometry measurements for and the well known Stanford Dragon. The new method describes in this paper is efficient on arbitrary or regular meshes but compulsorily oriented and closed objects to avoid false detections. We propound an improvement of the detection of salient, flat or cavity points that can be used for 3D segmentation.

Keywords : 3D characterization, edge detection, salient point detection, arbitrary or regular meshes.

1. INTRODUCTION

Recent years, lots of technological improvements were realized in the field of computer graphics. 3D data analysis is a huge domain, which starting with 3D acquisition, visualization, passing by triangles meshes or volumetric description, simplification and regularization of meshes, denoising, filtering, characterization and segmentation of objects. One of the most important part in these last types analysis is the edge detection or salient point detection in which research take different ways. Current techniques require several conditions and computations to efficiently define and characterize cloud points or meshes of objects. Nowadays, most characterization techniques are based on curvature analysis, underlines Page & al.[9]. In the aim of simplify or parameterize the 3D domain, Taubin & al.[11] has developed a matrix closely related to the matrix representation of the tensor of curvature that eigenvalues and eigenvectors give principal curvatures and directions. Desbrun & al.[3, 6] define a discrete operator that is able to characterize the curvature in the 1Ring neighborhood of regular meshes with Voronoi finite-volume area and extend it to arbitrary meshes with barycenter finite-volume area and preserve edges by providing isotropic and anisotropic smoothing techniques. They achieve good results, but this technique is sensible to noise. Chen & al.[1] evaluate the Gaussian curvature in the 1Ring neighborhood and extend the 1Ring to NRing, to obtain results on objects that have a large number of faces. They also order and simplify the NRing of each point and set up a watershed scheme in the neighborhood to achieve the detection of boundaries of 3D objects. Although, this technique give quite good results, it appears that it is difficult to define boundaries of objects and require several computation and condition to segment the inspected object correctly. A similar approach has been developed by Pulla & al[10]. These kinds of process are efficient and achieve good results on different types of meshes. Fournier & al.[7] has developed a technique to filter the output of a scanner similar to 2D processing that can be applied on meshes. Loriot & al.[2] use a slicing technique to detect boundaries of the object. This technique gives quite good results, but requires conditions to slice the 3D object and is not optimal on the beginning and on the end of the object because of the use of slicing method. Mao & al.[5] define a Susan-like method for denoising. They consider each point of the object as pixels and the angle between the normals of two points on the surface as color. Thus, they keep the definition of the 2D Susan operator and use it as a surfaced operator in 3D. They obtain a 3D non linear operator which achieves good denoising results.

Based on this different works, we describe a new Susan method to detect boundary of 3D objects. As defined by Smith & al.[4], to apply Susan operator in discrete 2D domain, a circular pixellized operator is defined. In the under influence area of the operator, the nucleus is defined as the center of the filter and defines the pixel under analysis. The color of the nucleus is picked and compared to other pixel colors in the filter area. If the pixel tested as a color close to the nucleus color (except for a threshold), a counter is incremented. The final Susan value is determined by the ratio area of near pixel over the total number of pixels in the area of the filter. Inspired by Monga & al.[8] that extend a classical 2D operator to 3D, we directly extend the Susan 2D method to 3D data. A voxellized sphere is defined that had a nucleus under the surface of the object and compute the ratio of under surface voxels over the total number of voxels of the sphere. This technique appear very simple but require some conditions on the object inspected. The object must be closed, have triangles meshes with normals oriented on the exterior and do not contain holes. The meshes can be regular or arbitrary, with a number of faces around each point differing or not. We are interested in point characterization and had not take into account the possible noise that appear during the acquisition of the 3D objects.

We will first define some requirements of the computation and describe the characterization operation. Secondly, we show results on a virtual 3D icosahedron, a real 3D flint artifact and the Stanford dragon. We finally discuss performances of our operator and conclude.

2. DESCRIPTION OF THE OPERATOR

2.1. Used 3D data format

In this part, we will describe the two types of 3D data used to improve our algorithm. It corresponds to "*.stl*" files that permit a knowledge of the cloud points and normals. It gets triangles meshes with associated points and normals. So, it's easy to create the point vector and the normal vector corresponding to the object. All followed processing are based on these two vectors that completely define the 3D object. After characterization, we create a "*.fcs*" file that permit color visualization of extracted characteristics and validation of our operator.

2.2. Principle

This part is composed of three sections : First we describe the NRing surfacing process (XMR exposed Chen & al.[1]), that refine a point definition with his surround direct or extended triangles meshes as an ensemble of triangle faces, normals and areas. The second describe the developed spherical operator that characterize a point as salient, flat or cavity. Finally, we describe the established method to compute the intersection of the NRing surface and our operator.

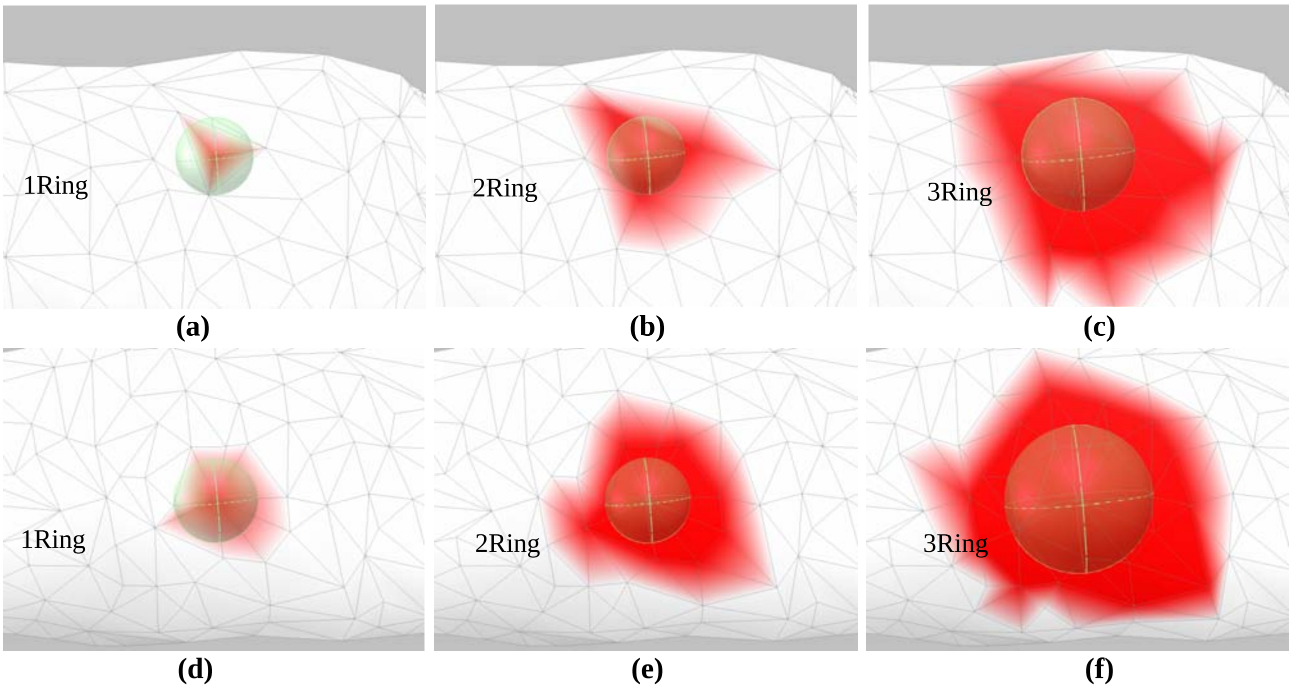
First, a copy of the 3D object is reproduced. This include the copy of point vector and normal vector. From this first definition, we named the direct local structure the 1Ring and the extended the 2Ring, ..., NRing[1]. In addition, we implement a spherical operator, which take the position of each point and precisely determine the intersection of the local NRing surface of the object and a sphere. Previously, to simplify the computation problem of the intersection, a voxel decomposition is executed on the based sphere of the operator. Indeed, Intersection of triangles meshes with continuous definition of a sphere is difficult to understand and require heavy calculation. Next, we simply move the sphere point to point to characterize them. The intersection volume between the oriented triangles meshes and the voxellized sphere, conclude the characterization of the tested point, by classifying and counting the number of voxels under the NRing surfaces. Finally, the resulting characteristic is obtained by establishing the ratio of the intersection volume over the total volume of the sphere. The ratio defines directly if the inspected point is salient, flat or cavity.

2.3. NRing surfacing process

Based on the point and normal vectors definitions, we extract for each point, the surround triangles meshes and points of the direct neighborhood and register them as the 1Ring neighborhood. Then, we extend it to NRing to obtain a knowledge of the local structure of each inspected point of the object. This method presented by Chen & al.[1], is time consuming, that's why we don't order faces or points. Indeed, the more extended the neighborhood, the more wasted the time.

2.4. Spherical operator and characterization

The operator correspond to a voxellized sphere. From this point, we need a ray to create the sphere. This ray is computed on the NRing$_{-1}$ neighborhood to avoid non-closed intersection. The following figures represent the evolution of the NRing for two different points :



(a) **(b)** **(c)**

(d) **(e)** **(f)**

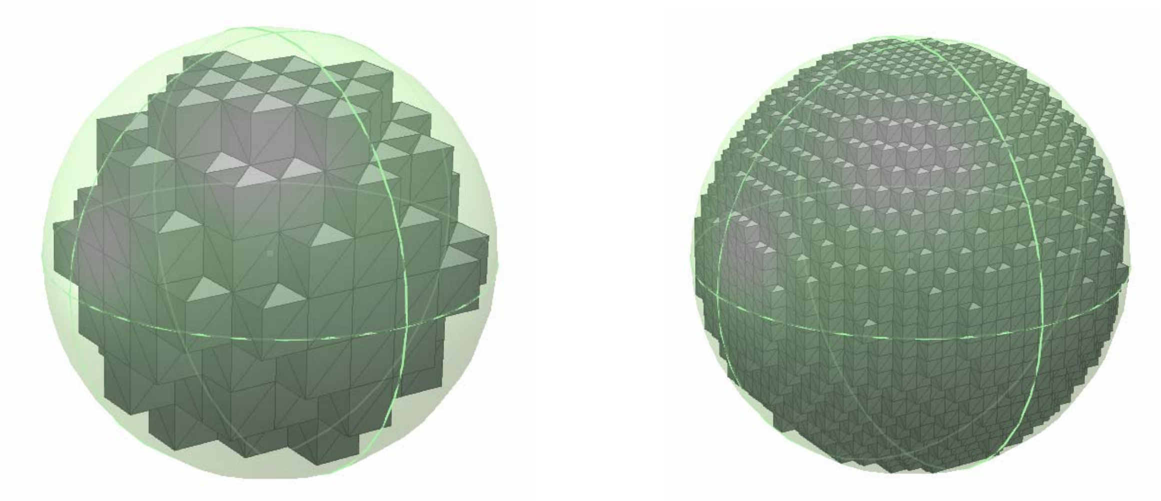
Figure 1. (a) and (d) represent non-closed intersections in a 1Ring neighborhood, (b) and (e) represent closed intersections in a 2Ring, (c) and (f) closed intersections in a 3Ring

So, we inspect all points and compute the minimum of the minimal Euclidean distances between the inspected point and all points of his NRing$_{-1}$ to determine the ray of the sphere for the object :

$$Ray.Sph.Obj = \min_{i=0}^{N}\left[\min_{k=0}^{M}\sqrt{\left[PtNRing_{-1}(k) - PtMes(i)\right]^2}\right], \qquad (1)$$

In (1), N corresponds to the number of points of the object and M the number of points of the NRing$_{-1}$ of the ith point.

Now, the sphere can be created, with the radius computed and voxellized it. First, a big voxel such as his half edge correspond to the ray of the sphere is created and his gravity center correspond to the sphere center. One that is defined, the voxel is broken into 27 equal parts which correspond to the new voxels. Voxels are labeled as In, Out or OnEdge and a level decomposition number is given to each voxel. Then, only voxels labeled as OnEdge are decomposed to minimize the computation time. When the volume of all the In voxels is judged sufficient to approximate the sphere, here 83% on the 3rd level, the decomposition is stopped and the last decomposition level is saved. Finally, a refinement of the decomposition of In voxels of levels differing from the last decomposition level is done to obtain a perfect voxellization of the sphere. In the next figures, we present the results of voxellized sphere at 2nd and 3rd level decomposition, from left to right. Only the In voxels for the 2nd and 3rd level decomposition are displayed :



(a) **(b)**

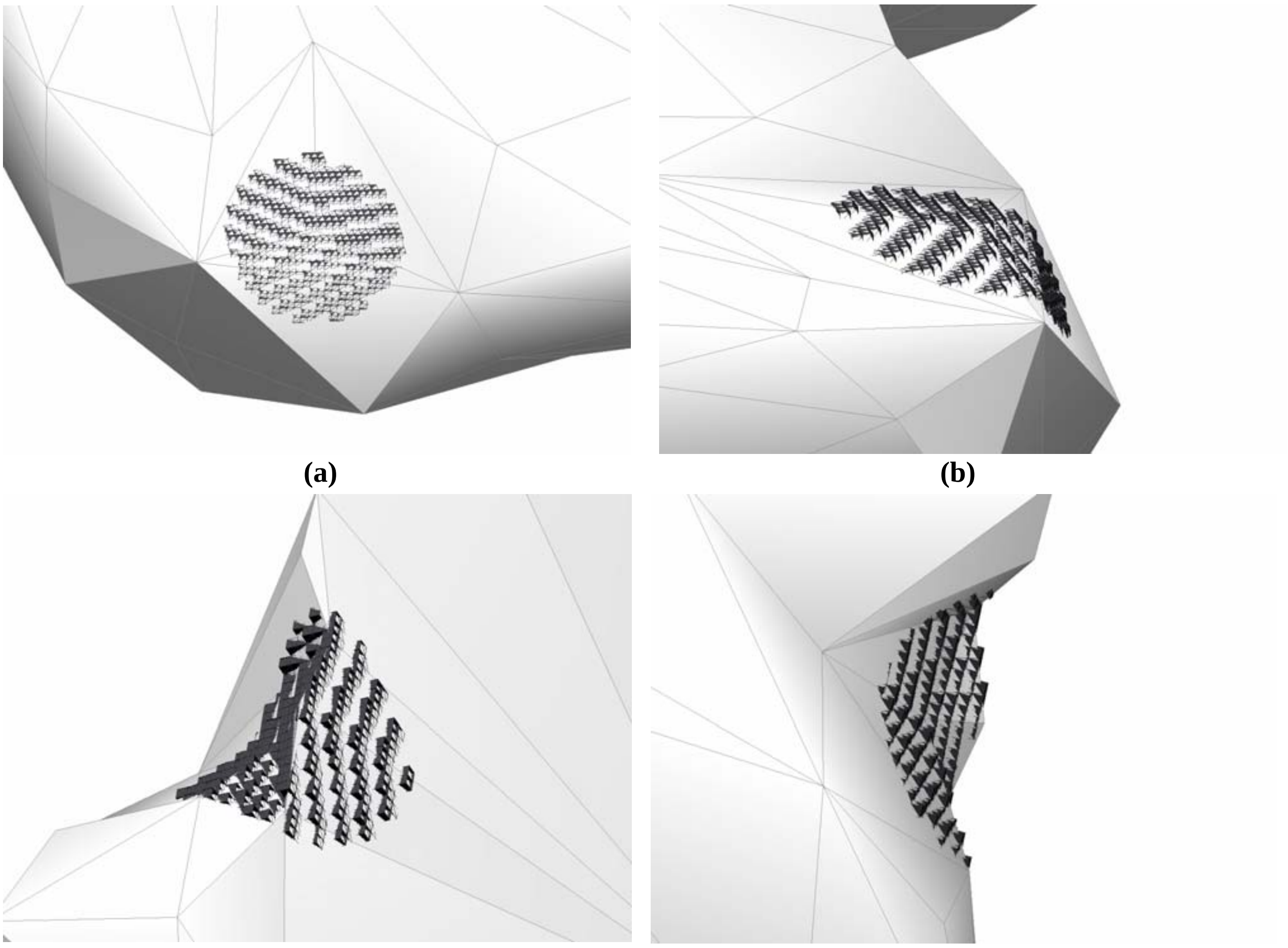
Figure 2. (a) represent the 2nd level and (b) the 3rd level decomposition of the voxellized sphere.

The 27tree versus the Octree decomposition is defined to make correspondence with 2D operators. All 2D linear operators have a number of rows and columns odd (3*3, 5*5,...) like Sobel, Prewitt, DOG, LOG,Susan or other operators, to define an apply filtering center. Moreover, as the 2D non linear Susan operator is adding pixels of near nucleus (filter center) gray levels, we adding a voxel if his gravity center is under the surface. This accelerate the computation time and simplify the application of our operator. The central voxel of the voxellized sphere, here defined as the 3D nucleus, is always count as a voxel under the surface. The characterization is deduce from the volumetric ratio of the under voxels over total volume of the voxellized sphere. We named this result the saliency degree. Whatever the object or curvatures inspected, the saliency degree has a response value limited at the interval]0,1[. First, a double threshold on the 3D Susan characteristics has been tested but not give quite precise representation of the 3D saliency degree extracted. Then, a color map of 1020 colors is created for visually classify points as salient, flat or cavity. Because of the geometry of objects that can contain no cavity points (icosahedron for example), the color map is differing, depending on the object inspected.

2.5. Intersection volume computation

The neighborhood and the voxellized sphere for each point of the object are defined. The computation of the intersection volume at each point is needed. To compute it, the 1Ring average of the normals at each point is computed. The average vector obtained is considered as the projection vector of all gravity centers of voxels on the NRing faces. Then, all gravity centers of voxels are projected one by one on all the NRing faces. To determine which face is valid for the inspected voxel, a control of the projected points along the average normal is done to verify that projected points are inside or on the border of the tested face, else we reject this face as a valid face for characterization of the tested voxel. Moreover, the Euclidean distance between the gravity center of the tested voxel and the projected point inside the tested face is saved if the face is valid.

If several faces are valid, the face with the minimum Euclidean distance between the voxel and the face is selected as valid. If none face is valid, the voxel is totally rejected. This last case is only present when the operator inspect a border point (salient) of the object, where the operator has a minimal response. Therefore, it is not disturbing the characterization (false responses) and visible results are good on points were operator is bad classify more than a half of the voxellized sphere. Next figures show results for two points of the flint artifact, that algorithm characterize well as salient points :



(a) (b)

(c) (d)

Figures 3. (a) and (b) represent two different views of a volume intersection at a simple point, (c) and (d) represent two different views of a volume intersection at a more complex point.

3. RESULTS

We made different tests on objects. The color at a point defines the extracted characteristic type of the point. The first object, an icosahedron, have a number of points equal to 642 with a pseudo regular triangles meshes and a number of faces constant around each point (6 faces), except for the 12 points defining the icosahedron (5 faces). The following figures show characterization results with a 3Ring and the following color bar :

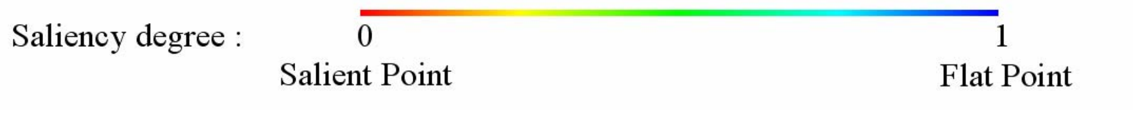


Figure 4. Color bar with a precision of 1020 colors

(a) (b)

Figure 5. (a) icosahedron with a 2nd decomposition level, (b) icosahedron with a 3rd decomposition level.

The second object, a flint artifact, has a number of points equal to 1560 with an irregular meshes and a number of faces varying around each point. The following figures present results for a 3Ring neighborhood with the following color map :

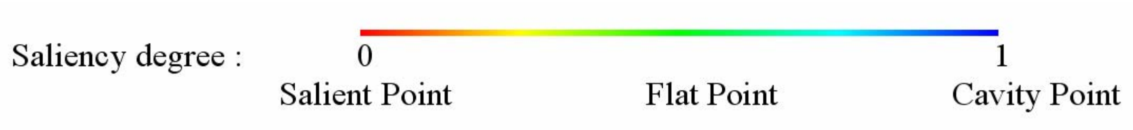


Figure 6. Color bar with a precision of 1020 colors

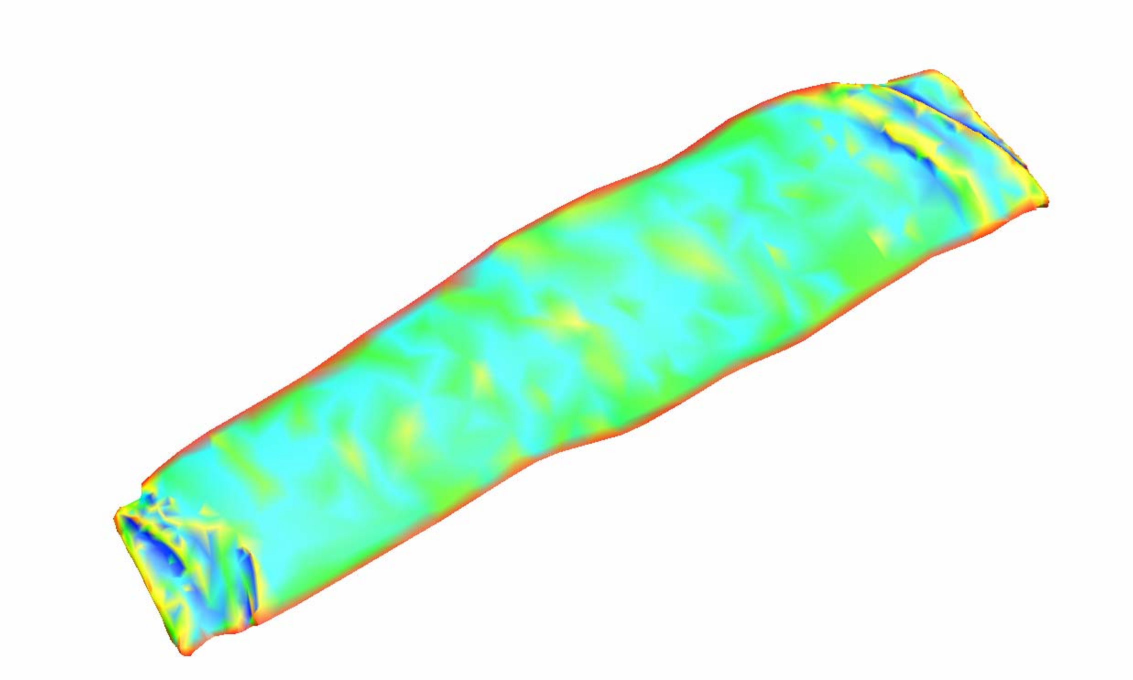


Figure 7. Sight of lower part of the characterized flint artifact with a 3Ring neighborhood and a 3rd decomposition level for the sphere.

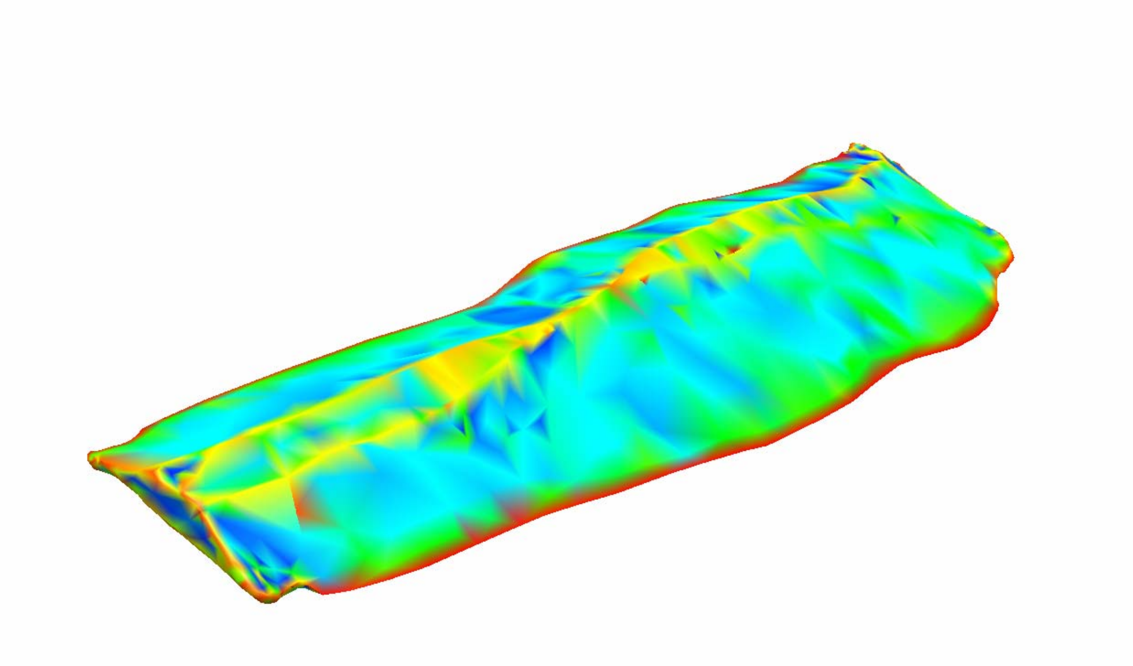


Figure 8. Sight of top part of the characterized flint artifact with a 3Ring neighborhood and a 3rd decomposition level for the sphere.

The last object is the well known Stanford dragon. The number of points after simplification is 12499 with an irregular triangles meshes and a number of faces varying around each point. The following figures present results for a 3Ring neighborhood :

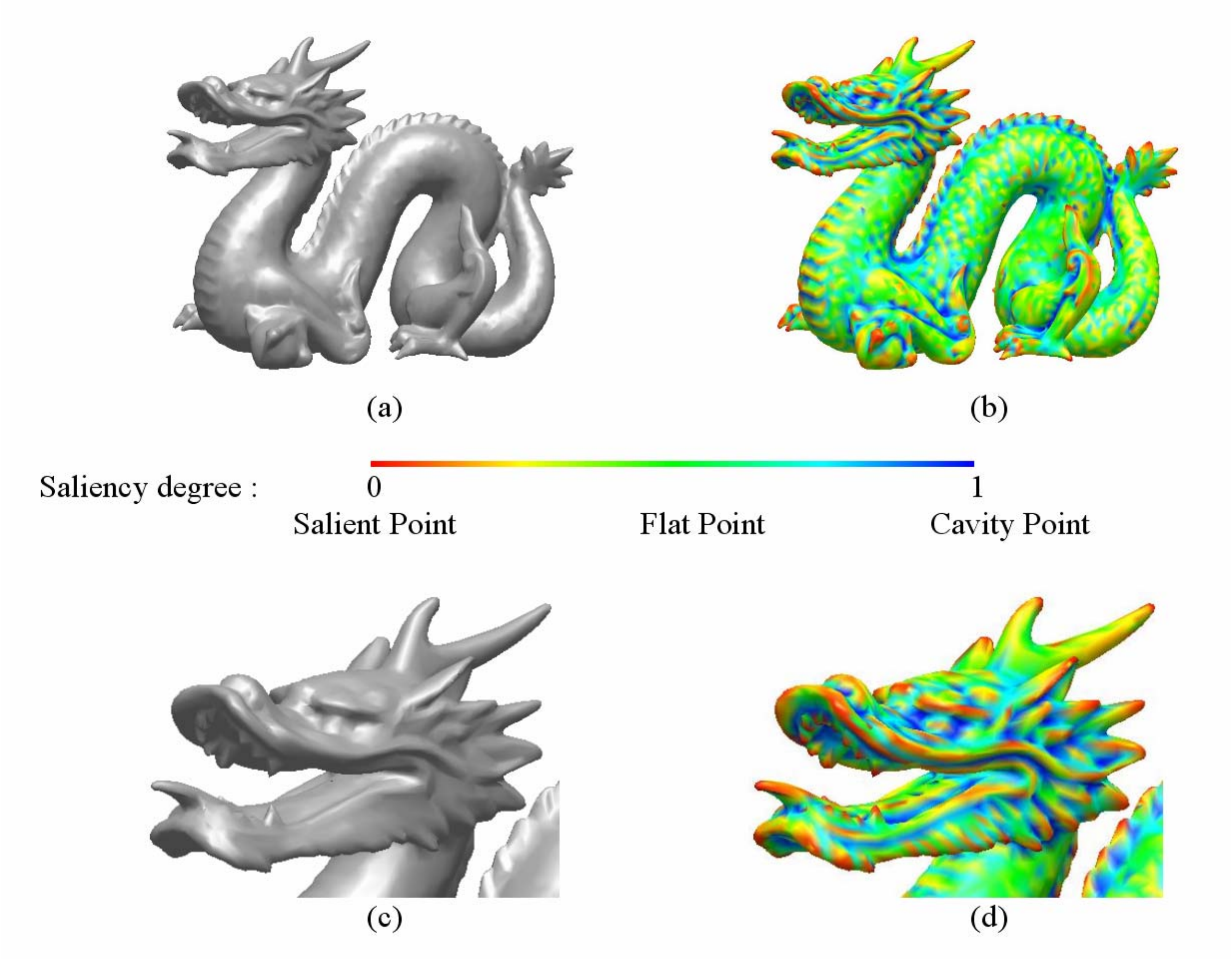


Figure 9. (a) General view of the Stanford dragon, (b) Characterized Stanford dragon with a 2nd decomposition level, (c) Face view of the Stanford dragon and (d) Characterized Stanford dragon face with a 2nd decomposition level.

4. DISCUSSION & CONCLUSION

In this article, we have presented a new approach to point characterization of 3D objects and introduced a new measure, the saliency degree that is able to classify points directly as salient, flat or cavity. A new 3D Susan operator is defined, from his 2D definition, and applied to differents objects. We have defined an operator to be applied on “ perfect ” scanned objects, to characterized them and why not in the future segmenting or simplifying them. Future works will be oriented in the study of noise influence on the saliency degree and the application of our operator for denoising with a multi-scale architecture. Moreover, a research on the principals saliency degree and directions is in progress to make difference between a cavity and a valley or a salient and a crest point.

References

1. L. Chen, N. D. Georganas, “An efficient and robust algorithm for 3D mesh segmentation,” Multimed Tools Appl, © Springer Science + Business Media (2006).
2. B. Loriot, Y. Fougerolle, C. Sestier, R. Seulin, “3D acquisition and modeling for flint artefacts analysis,“ SPIE Optical Metrology – Optics for AAA (2007).
3. M. Desbrun, M. Meyer, P. Schröder, A. H. Barr, "Discrete differencial-geometry operators for triangulated 2-manifolds,". *VisMath* (2002).
4. S.M. Smith, J.M. Brady, “Susan – A new approach to low level image processing,“ in International Journal of Computer Vision, vol. 23 (1), pp. 45-78 (1997).
5. Z. Mao, L. Ma, M. Zhao, X. Xiao, “SUSAN structure preserving filtering for mesh denoising,“ The Visual Computer vol. 22, pp. 276-284 (2006).
6. M. Desbrun, M. Meyer, P. Schröder, A.H. Barr, “Implicit fairing of irregular meshes using diffusion and curvature flow,“ Computer Graphics Proceedings (SIGGRAPH'99), pp. 317-324 (1999).
7. M. Fournier, J-M. Dischler, D. Bechmann, "Filtrage adaptatif des données acquises par un scanner 3D et représentées par une transformée en distance volumétrique," *AFIG day*, (2006).
8. O. Monga, R. Deriche, "3D edge detection using recursive filtering : Application to scanner images," *INRIA Research report* number 930, Program 6 (1988).
9. D.L. Page, Y. Sun, A.F. Koschan, J. Paik, M.A. Abidi, "Normal vector voting : Crease detection and curvature estimation on large noisy meshes," *Graphical Models vol.* 64 (3-4), pp. 199-229 (2002).
10. S. Pulla, A. Razdan, G. Farin, "Improved curvature estimation for watershed segmentation of 3D meshes,". Manuscript (2001)
11. G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," *Proceedings of the Fifth International Conference on Computer Vision* pp. 902-907 (1995).