Constant Curvature Region Decomposition of 3D-Meshes by a Mixed Approach Vertex-Triangle

Lavoué Guillaume
LIRIS FRE 2672 CNRS
43, Bd du 11 novembre
69622 Villeurbanne Cedex,
France
glavoue@liris.cnrs.fr

Dupont Florent
LIRIS FRE 2672 CNRS
43, Bd du 11 novembre
69622 Villeurbanne Cedex,
France
fdupont@liris.cnrs.fr

Baskurt Atilla
LIRIS FRE 2672 CNRS
43, Bd du 11 novembre
69622 Villeurbanne Cedex,
France
abaskurt@liris.cnrs.fr

ABSTRACT

We present a new and efficient algorithm for decomposition of arbitrary triangle meshes into connected subsets of meshes called regions. Our method, based on discrete curvature analysis decomposes the object into almost constant curvature surfaces and not only "cut" the object along its hard edges like traditional methods. This algorithm is an hybrid approach vertex-triangle, it is based on three major steps: vertices are first classified using their discrete curvature values, then connected triangle regions are extracted via a region growing process and finally similar regions are merged using a region adjacency graph in order to obtain final patches. Experiments were conducted on both CAD and natural models, results are satisfactory. Segmented patches can then be used instead of the complete complex model to facilitate computer graphic tasks such as smoothing, surface fitting or compression.

Keywords

Triangle mesh, Segmentation, Discrete curvature, Vertex classification, Triangle region, Region adjacency graph.

1. INTRODUCTION

Triangle mesh is by far the most popular model for 3D-objects/3D-surfaces representation. Reasons are its algebraic simplicity (linear algebra), which facilitates largely rendering algorithms, and its capacity to model any complex object of arbitrary topology. Moreover, the majority of triangle manipulations necessary in graphic applications are managed by the graphic hardware, which facilitates the diffusion and the preponderance of this model. Many tasks in computer graphic and computer vision are performed on 3D-meshes: smoothing, decimation, surface fitting and compression (due to the large amount of data carried by a triangle mesh). The difficulty of these algorithms increases when the object becomes complicated, for instance when it is composed of high complexity surfaces or numerous components. Therefore the object decomposition, into parts or patches, becomes attractive since it simplifies the complex problems of treating a complicated object in several sub-problems, each dealing with simpler subsets of the object. Within this framework, we present a curvature based triangle mesh segmentation method which decomposes a 3D-mesh into connected subsets of mesh, called regions. Our purpose is to be able to fit subdivision patches on these regions for an adaptive compression objective.

Section 2 presents the related work on the mesh segmentation subject, whereas the overview of our method is detailed in section 3. Sections 4, 5 and 6 deal with the three different steps of the method: vertex classification, region growing and region merging. Experiments and results are presented and discussed in section 7.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.
Copyright UNION Agency – Science Press

2. RELATED WORK

There has been a considerable research work relevant to the problem of 3d-object segmentation. However the majority of these methods concern range images [Hof87] [Bes88][Rom94][Leo97] or 3d point clouds [Che03][Cha00]. Only few studies concern triangle meshes which is nevertheless the most widespread representation for 3d-objects. Wu and Levine [Wu97] present a physics-based original method which uses

the idea of electrical charge but this approach is computationally expensive. The other approaches generally use discrete curvature analysis combined with the Watershed algorithm described by Serra [Ser82] in the 2D image segmentation field. Mangan and Whitaker [Man99] generalize the Watershed method to arbitrary meshes, using the Gaussian curvature or the norm of covariance of adjacent triangle normals at each mesh vertex as the *height field*. Sun et al. [Sun02] use the Watershed with a new curvature measure based on the eigen analysis of the surface normal vector field in a geodesic window. More recently, Razdan and Bae [Raz03] proposed an hybrid method which combines Watershed algorithm with the extraction of feature boundaries by the analysis of dihedral angle between polygon faces. Zhang et al. [Zha02] use the sign of the Gaussian curvature to mark boundaries, and process a part decomposition.

We have distinguished two major shortcomings in these existing methods. They are described below.

Firstly, many approaches are only vertex based [Sun02] [Zha02], each vertex has its region information, therefore triangles on boundaries have multi-regions information, it results that boundaries are fuzzy; they are not clearly identified in term of edges. Our method is an hybrid approach vertex-triangle, which combines a vertex classification with a triangle region growing and merging. Boundaries between regions are clearly distinguishable edges.

Secondly, most of the approaches discussed above, particularly those based on the Watershed algorithm, extract regions surrounded by high curvature boundaries [Sun02] [Zha02] [Raz03] (see Fig.9.b) but fails to distinguish simple curvature transition between vertices (see Fig.9.a) without curvature pick. The K-Means vertex classification that we use allows to detect these transitions.

3. METHOD OVERVIEW

We present a segmentation algorithm for surface decomposition of arbitrary triangle meshes, based on curvature information analysis. Our method decomposes the object into almost constant curvature triangle regions with precise edge boundaries. It does not only "cut" the object along its edges but detects every curvature transitions. Fig.1 shows the diagram for this approach.

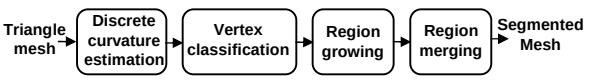


Figure 1. Diagram of our segmentation method.

Firstly, discrete curvature is calculated for each vertex according to the work of Meyer et al. [Mey02]. Then vertices are classified into clusters (see Section 4), according to their principal curvatures values *Kmin* and *Kmax*. A region growing algorithm is then processed (see Section 5) assembling triangles into connected labeled regions according to the vertex clusters. Holes between regions are filled taking into account boundary criteria. Finally a region adjacency graph is processed and reduced in order to merge similar regions (see Section 6) according to several criteria (curvature similarity, size, common perimeter).

4. VERTEX CLASSIFICATION

Vertices of the mesh are classified according to their discrete curvature.

Discrete curvature estimation

To estimate the curvature information of each vertex of the mesh we have implemented the work of Meyer et al. [Mey02], using averaging Voronoi cell and the mixed Finite-Element/Finite-Volume method. The Gaussian curvature, the mean curvature and the principal curvatures are estimated using the following equations (see Fig. 2):

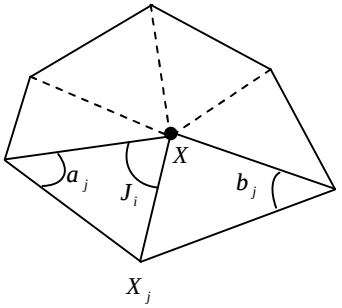


Figure 2. 1-ring neighbors ($N_1(X)$) around vertex X .

- Discrete Mean Curvature Normal of vertex X is defined by:

$$K(X) = \frac{1}{2A} \sum_{j \in N_1(X)} (\cot a_j + \cot b_j)(X_j - X) \quad (1)$$

Where A *is* the Voronoi area region (see equation 4).

- Discrete Gauss Curvature of vertex X is processed as follows:

$$Kg(X) = (2\Pi - \sum_{i=1}^{f} J_i) / A \quad (2)$$

f is the number of adjacent faces to the vertex X (6 in the example of Fig.2). The Gauss curvature can be negative (case of hyperbolic vertices).

- Discrete principal curvatures of vertex X are computed with the following equations:

$$Kmax(X) = Kh(X) + \sqrt{\Delta(X)}$$
$$Kmin(X) = Kh(X) - \sqrt{\Delta(X)} \qquad (3)$$

with $Kh(X) = \frac{1}{2}\|K(X)\|$ and $\Delta(X) = Kh^2(X) - Kg(X)$

A is the Voronoi area region, and is defined by:

$$A = \frac{1}{8}\sum_{j \in N_1(i)} (\cot a_j + \cot b_j)\|X_j - X\| \qquad (4)$$

$Kmax$ is always positive, whereas $Kmin$ follows the sign of Kg.

Results are presented in Fig.3. The color scale starts with the blue and ends with the red and depends on the considered curvature. When considering the interior of the torus, the mean curvature (a) is low and the gauss one is high in negative (blue color in b), just as the minimum curvature *Kmin*. In the interior, *Kmin* is high in absolute value (red color in c) whereas the maximum curvature *Kmax* is low (blue color in d), in this area *Kmin* and *Kmax* are almost equal (2,3 for *Kmin* and 2,7 for *Kmax*) .

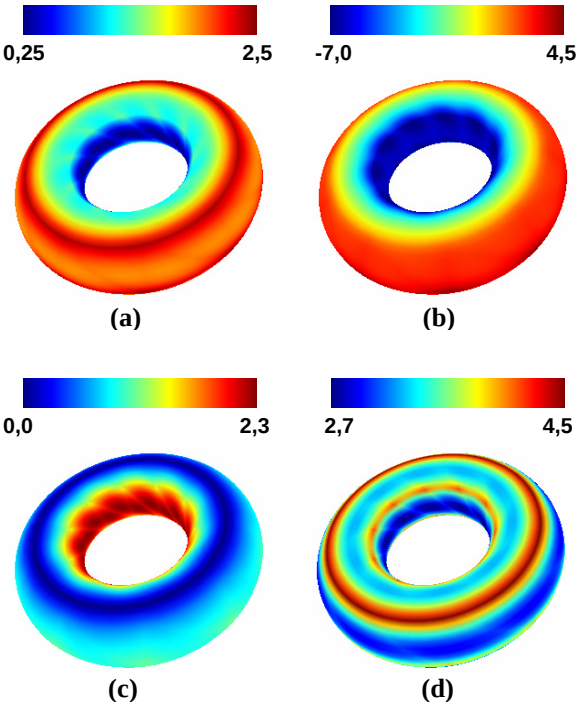


Figure 3. Curvature values of triangulated torus using pseudo-colors: (a) Mean, (b) Gaussian, (c) Minimum (absolute value), (d) Maximum.

Curvature classification

Vertices are classified according to the values of their principal curvatures *Kmin* and *Kmax* (see Fig.4). *Kmin* can be negative, we consider its absolute value, it is not necessary to differentiate positive and negative values in our classification. The clustering is done via a K-Means algorithm (an usual unsupervised fast classification method) [Ger92], completed by a cluster regularization (merging of small or similar clusters). At the end of the algorithm each vertex is associated with a Cluster C_i and a classified curvature value c_i (c_i is in fact a two scalars vector which contains the classified values for *Kmin* and *Kmax*). The number of clusters K is fixed by the user, but is not critical for the final segmentation result because of the region growing and merging steps.

Fig.4 shows the vertex classification process applied to the Fandisk object (6475 vertices). The number of clusters in the curvature space is fixed to 5, by the user (clusters colors are yellow, orange, blue, dark blue and light blue).

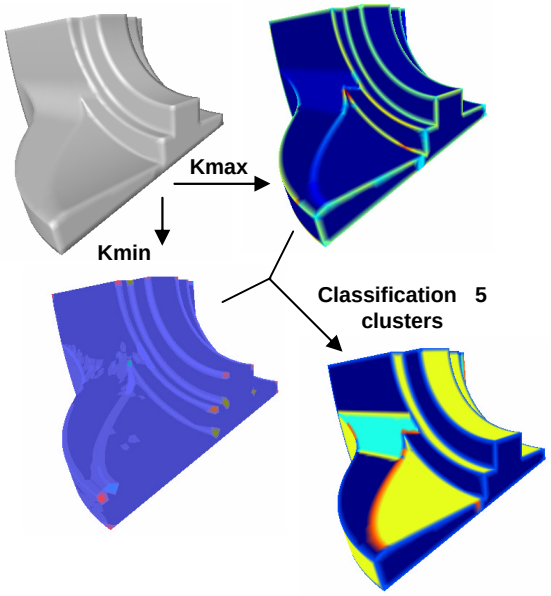


Figure 4. Vertex classification of the *Fandisk* mesh in 5 curvature clusters.

5. REGION GROWING

Once vertices have been classified, a triangle growing and labeling operation is performed as follows. Each triangle which has its three vertices belonging to the same cluster C_L (Corresponding to the classified curvature value c_L) is considered as a seed. Starting from a seed, a region with a new label L and the curvature value c_L is created containing this triangle.

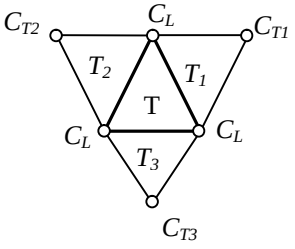
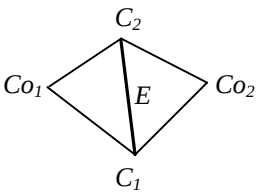


Figure 5. Seed triangle T with vertices from cluster C_L and its neighbor triangles.

Then a recursive process extends this region: for each triangle T belonging to the region *L*, we look its three neighbor triangles T_i (see Fig.5). For each T_i, if the cluster of its third vertex C_{Ti} is equal to C_L, thus T_i is aggregated to this region (Label *L*), else it is not labeled. This process is repeated for every other triangle marked as seed and not yet labeled.

Once all seeds have been treated, we have *n* connected regions with holes between them (in black Fig.7.b). Note that these Not-labeled triangles are potential candidates for defining the exact frontiers between labeled regions.

A crack filling process is performed. Not-labeled triangles are assigned to regions, according to labels of their neighbor triangles. The algorithm is iterative. For each non-labeled triangle *T*, its neighbors T_i are considered. For each T_i, we check if it is labeled or not, and if their common edge is considered as a *boundary* or not. If T_i has got a label and the common edge T-T_i is not a *boundary*, we memorize its label. Finally, if there is at least one memorized label, we assign *T* to the most represented one. We repeat the iterative process until every triangle is labeled.



E is a *boundary* if:

$Co_1 \neq C_1$ and $Co_1 \neq C_2$

and

$Co_2 \neq C_1$ and $Co_2 \neq C_2$

Figure 6. *Boundary* edge detection for the cracks filling algorithm.

An edge *E* is considered as a *boundary* if the triangles which share this edge have their third vertices Co_1 and Co_2 with different labels from the edge vertices ones (C_1 and C_2) (see Fig.6). This *boundary* notion is only used in this part of the algorithm in order to improve the crack filling method and does not correspond to the final boundaries at the end of the segmentation method.

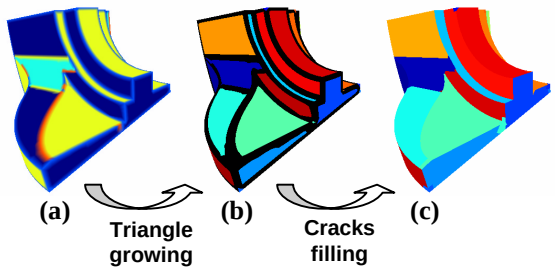


Figure 7. The region growing process for the Fandisk mesh: (a) 5 clusters curvature classification, (b) the connected labeled regions, (c) the result of the cracks filling post-process. (Regions colors are randomly chosen).

Fig.7 shows the region growing process for the Fandisk object, starting from a 5 clusters vertex classification. The region growing extracts 38 connected regions. Holes in black are then filled according to the crack filling post-process.

A fixed value of *K* for the K-Means classification algorithm can generate different sets of clusters because of the random choice of the *K* initial seeds. Thus for a given *K*, the merging step will give different results in term of number and localization of extracted regions. Because of this uncertainty, a region merging process was developed, in order to unify results.

6. REGION MERGING

According to the nature of the object to segment, and to its number of curvature clusters, the region growing step can extract numerous small regions, (see Fig.10). The purpose of the merging step is to reduce this over-segmentation, in order to extract significant sub-surfaces, corresponding to 'meaningful' parts of the object.

The region adjacency graph

The efficiency of an algorithm depends on the data scheme used. The purpose here is to merge adjacent similar regions. Thus a good representation to operate is a region adjacency graph (RAG), a data scheme used in image segmentation [Saa94]; it is an algebraic structure that contains a set of nodes and a set of edges. Each node represents a connected region (i.e. a connected subset of the mesh), and each edge represents an adjacency between two regions. Edges are evaluated by a curvature distance between the two corresponding regions.

General algorithm

Once connected regions have been extracted by the region growing algorithm, we have a region-map of the mesh, with the labels of the regions. Thanks to the region-map, the RAG is processed, and distances between adjacent regions are calculated. Then the reduction of the graph is processed: at each iteration the smallest edge of the graph is eliminated, thus the corresponding regions are merged; then the graph is updated.

When two regions are merged, the curvatures of the resulting region are merged proportionally to their areas. This graph reduction stops when the number of regions reaches a queried number chosen by the user, or when the weight of the smallest edge is larger than a given threshold.

Region distance measurement

The distance D_{ij} used in our method is equal to the curvature distance DC_{ij}, between the two

corresponding regions R_i and R_j weighted by two coefficients: N_{ij}, which measures the nesting between the two corresponding regions and S_{ij} of which aim is to eliminate the smallest regions.

$$D_{ij} = DC_{ij} \times N_{ij} \times S_{ij} \qquad (5)$$

Each coefficient is detailed in the following paragraphs.

The curvature distance DC_{ij} is processed using the curvature values C_i and C_j of the two corresponding regions and the curvature value C_{ij} of their boundary.

$$DC_{ij} = \left\| C_i - C_{ij} \right\| + \left\| C_j - C_{ij} \right\| \qquad (6)$$

C_i and C_j come from the region growing step. C_{ij} is the average of the vertices curvatures on the boundary between the two regions. Only vertices with two incident edges separating these regions (*real boundary vertices)* are taken into account (see Fig.8), in order to consider only the real boundary between them.

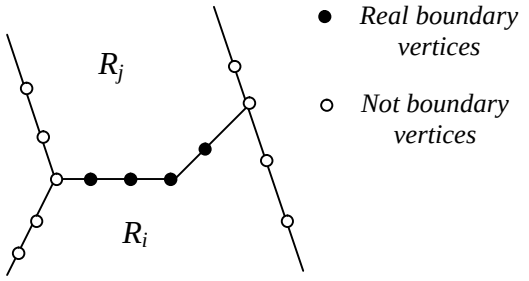


Figure 8. Representation of the vertices taken into account for the calculation of the average curvature of the boundary between R_i and R_j.

It is important for the calculation of the curvature distance between R_i and R_j to consider not only their respective curvatures C_j and C_j but also their boundary one C_{ij}, because two situations may exist between these regions. Either regions have different curvatures and no precise boundary (see Fig.9.a), or regions have almost the same curvature and a very different boundary curvature (see Fig.9.b).

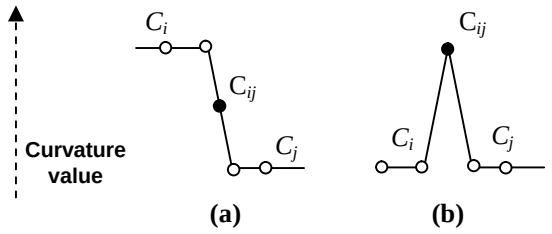


Figure 9. The two different situations between two adjacent regions. (a) no boundary but a curvature difference, (b) no curvature difference, but a significant boundary.

The N_{ij} **coefficient** measures the nesting between the two corresponding regions.

$$N_{ij} = \frac{\min(P_i, P_j)}{P_{ij}} \qquad (7)$$

with P_i the perimeter of the ith region and P_{ij} the size of the common border between ith and jth regions. This coefficient was introduced in image processing by Schettini [Sch93] for color image segmentation. The aim of the N_{ij} factor is to consider the spatial disposition of the regions in the merging decision. Regions with a large common border are more likely to belong to the same 'meaningful' part, thus their similarity distance is reduced.

The S_{ij} coefficient purpose is to accelerate the fusion of the smallest regions.

$$S_{ij} = \begin{cases} e & if\ (A_i < A_{\min}\ or\ A_j < A_{\min}) \\ 1 & else \end{cases} \qquad (8)$$

where A_i is the area of the ith region, A_{min} is a minimum area fixed by the user and e is a positive number near 0. The S_{ij} factor can be considerate as a filtering factor. When a region's area is smaller than A_{min}, it is considered too small, thus its distance with its adjacent regions is reduced by the S_{ij} coefficient, equal to e ; the considered region will be more easily merged with another. It is a method to eliminate the smallest regions. The value of A_{min} depends on the queried size (or number) of final regions. The value of e is fixed to 1e-5. This value accelerates the fusion of the smallest regions, while keeping the merging order.

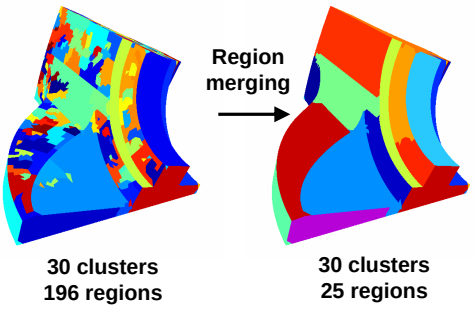


Figure 10. The region merging process for the Fandisk mesh.

Fig.10 shows the merging process. The initial pre-segmented object was obtained after the classification step in the curvature space (30 curvature clusters), and after the region growing step. It contains 196 connected spatial regions. After the merging process, the final region number is 25. For this example this number was fixed by the user.

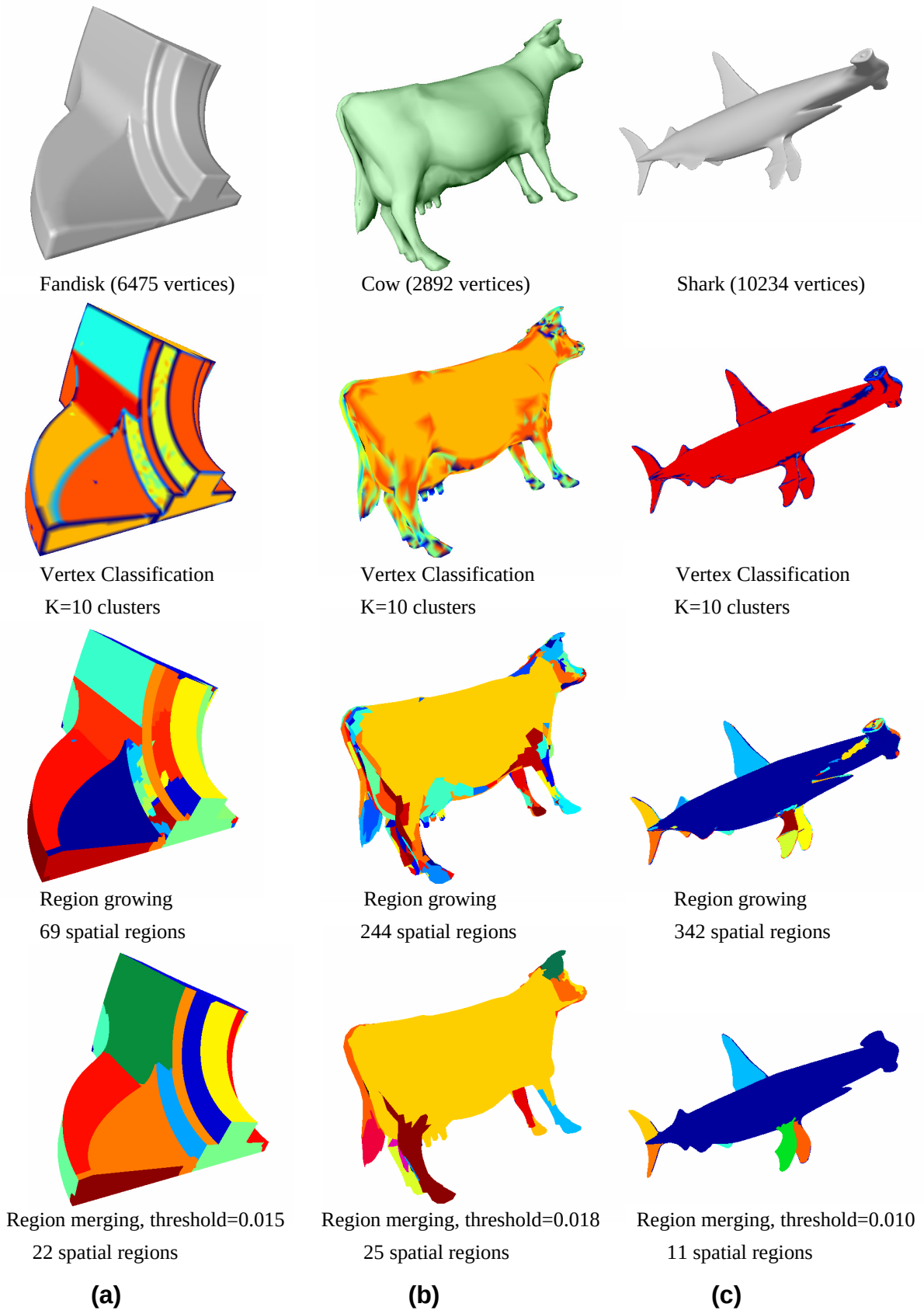


Figure 11. Segmentation of (a) Fandisk, (b) Cow and (c) Shark object. The number of curvature clusters is 10 for all. The region merging threshold is 0,015 for Fandisk, 0,018 for Cow and 0,010 for Shark.

7. EXPERIMENTS AND RESULTS

Our segmentation method was tested on several different objects. Examples are given for natural objects (Fig.11.b-c) and a mechanical part (Fig.11.a). For the classification step the cluster number *K* is fixed to 10 and A_{min} is fixed to 0.5% of the total mesh area. The region merging threshold is chosen heuristically according to the nature of the object and to its curvature distribution (0,015 for Fandisk, 0,018 for Cow and 0,010 for Shark).

For the natural objects, the aim of a segmentation method is generally the extraction of 'meaningful' parts for a possibly further analysis, that is to say, features which represent a semantic notion: the shark's fin, or the cow's paws for instance. We have good results with our method, although boundaries between regions are not very precise. The bull's paws, head and tail are well extracted (see Fig.11.b), just as the shark's fins, and tail (see Fig.11.c).

Concerning the Mechanical parts, the purpose of a segmentation algorithm is generally rather the extraction of simple sub-surfaces for a further surface fitting or adaptive smoothing for instance. The mechanical part Fandisk is correctly segmented (see Fig.11.a), the segmented surfaces are almost from constant curvature. On the other hand, we can observe that boundaries between segmented regions are not completely correct, even if the general shape is good. We plan to develop a boundary rectification method, in order to obtain clean and regular boundaries.

K	**Region Number after growing step**	**Region Number after merging step**
5	38	22
10	70	22
10	65	21
10	113	23
15	192	22
15	160	20
15	200	22

Table 1. Influence of the cluster number *K* of the classification algorithm, on the number of final regions for a given threshold.

The number of curvature clusters *K* which parameters the K-means algorithm during the vertex classification step is not a critical factor. Of course it influences the number of regions created after the growing step (besides, this number can vary for a same *K*, because of the random choice of the *K* initial seeds for the K-Means algorithm) but the final region number is regularized by the merging algorithm. We have conducted tests with several objects; results for Fandisk are shown in Table 1. The vertex classification was processed with different values for *K* and a unique threshold 0,02 was chosen for the region merging step, results show that *K* influence is negligible in the final region number, moreover resulting segmented regions are almost identical. Therefore *K* can be fixed (to 10 for instance) in the algorithm and does not have to be considered as a parameter for the method.

8. CONCLUSION

This paper presents an original method to decompose a 3D-object into connected sub-surfaces. This algorithm processes first a vertex classification and then extracts and merges triangle regions by using a region adjacency graph. This hybrid vertex-triangle approach dissociates itself from the traditional Watershed based methods which fail to extract hard boundaries [Raz03]. Moreover the proposed method permits the user to choose the final number of regions; if he does not have this information he can simply fix a precision threshold, we are investigating the possibility to process automatically this graph reduction stopping threshold.

This work is part of a larger compression process. The objective is to fit the segmented sub-surfaces with subdivision surfaces, in order to obtain the object in the form of a set of "light" subdivision patches, which will allows adaptive and scalable compression and transmission.

9. ACKNOWLEDGMENTS

This work is supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the Semantic-3D national project (http://www.semantic-3d.net).

10. REFERENCES

[Bes88] Besl, P. Surfaces in Range Image Understanding. New York, USA, Springer-Verlag, 1988.

[Cha00] Chaine, R., Bouakaz, S. Segmentation of 3-D Surface Trace Points, Using a Hierarchical Tree-Based Diffusion Scheme. Fourth Asian Conference on Computer Vision, Taiwan, Vol 2, 995-1002, 2000.

[Che03] Chevalier, L., Jaillet, F., and Baskurt, A. Segmentation and superquadric modelling of 3D objects. WSCG, Plzen - Bory, Czech Republic, Vol 11, No 2, 232-40, 2003.

[Ger92] Gersho, A., Gray , R. Vector Quantization and Signal Compression. Boston, USA, Kluwer Academic Publishers, 1992.

[Hof87] Hoffman, R., Jain, A.K. Segmentation and classification of range images. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 9, No 5, 608-20, 1987.

[Leo97] Leonardis, A., Jaklic, A., and Solina, F. Superquadrics for Segmenting and Modeling Range Data. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 19, No 11, 1289-95, 1997.

[Man99] Mangan , A., Whitaker, R. Partitioning 3D Surface Meshes Using Watershed Segmentation. IEEE Visualization and Computer Graphics, vol 5, No 4, 308-21, 1999.

[Mey02] Meyer, M., Desbrun, M., Schröder, P. et al. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. International Workshop on Visualization and Mathematics, Berlin, Germany, 2002.

[Raz03] Razdan , A., Bae, M. A hybrid approach to feature segmentation of triangle meshes. Computer-Aided Design, vol 35, No 9, 783-9, 2003.

[Rom94] Rom, H., Medioni, G. Part decomposition and description of 3d shapes. International Conference on Pattern Recognition, Jerusalem, Israel, Vol 1, 629-32, 1994.

[Saa94] Saarinen, K. Color image segmentation by a watershed algorithm and region adjacency graph processing. IEEE International Conference on Image Processing, Austin, TX, USA, Vol 3, 1021-5, 1994.

[Sch93] Schettini, R. A Segmentation Algorithm For Color Images. Pattern Recognition Letters, vol 14, 499-506, 1993.

[Ser82] Serra, J. Image Analysis and Mathematical Morphology. Academic Press, London, 1982.

[Sun02] Sun, Y., Page, D., PAIK, J. et al. Triangle Mesh-Based Edge Detection And Its Application To Surface Segmentation And Adaptive Surface Smoothing. IEEE International Conference on Image Processing, Rochester, NY, USA, Vol. 3, 825-28, 2002.

[Wu97] Wu, K., Levine, M. 3d part segmentation using simulated electrical charge distributions. IEEE Transaction on Pattern Analysis and Machine Intelligence., vol 19, 1223-35, 1997.

[Zha02] Zhang, Y., PAIK, J., Koschan, A. et al. A simple and efficient algorithm for part decomposition of 3D triangulated models based on curvature analysis. IEEE International Conference on Image Processing, Rochester, NY, USA, Vol 3, 273-76, 2002.