AUTOMATICALLY COMPUTED MARKERS FOR THE 3D WATERSHED SEGMENTATION

Sébastien Delest, Romuald Boné, Hubert Cardot

Université François Rabelais de Tours, Laboratoire Informatique
64 avenue Jean Portalis, 37200 Tours, France
{sebastien.delest, romuald.bone, hubert.cardot}@univ-tours.fr

ABSTRACT

This paper presents a new approach to the mesh segmentation based on watershed transformation. We propose an original method to compute markers from the topological information of the mesh. The skeleton of the mesh allows the interpretation of the meaningful parts and the watershed transformation builds the boundaries of theses parts. Our method, which combines the patch-type and the part-type segmentation approaches, is particularly well adapted to the problematic of meaningful part segmentation.

Index Terms— Mesh Segmentation, Watershed Transformation, Markers, Skeleton Graph

1. INTRODUCTION

Mesh segmentation has become an important problem and a necessary element in many applications in computer graphics like visualization and modeling, metamorphosis, compression, 3D shape retrieval, collision detection, texture mapping, reverse engineering, etc. The shape of models is important and can lead to different segmentation approaches depending on whether it is about natural shapes or mechanical parts for example. Mesh segmentation methods are mainly classified into two groups, the patch-type and the part-type, the former being related to surface segmentation of the model and the latter corresponding to shape and part segmentation. The patch-type methods segment the object into patches according to certain geometric properties such as planarity, uniformity, size or convexity and may be used as a pre-processing for the recognition of meaningful features. The part-type segmentation creates larger sub-meshes which can be associated to physical 3D parts of the object in the manner of semantics-oriented approaches to shape segmentation.

Several approaches to mesh segmentation into meaningful parts have been proposed in the past. Many methods are clustering oriented such as region growing, iterative clustering, spectral clustering, feature point based clustering and fuzzy clustering. Other techniques use skeleton based methods and snake based methods. Shamir offered a good survey of the mesh segmentation in [1].

Katz *et al.* [2] have recently proposed an original approach using feature point and core extraction. The mesh is transformed in a pose-invariant representation and its segmentation scheme allows decomposing the model in several levels of details. This method is discussed in the paper of Attene *et al.* [3] regarding the comparative study of the latest mesh segmentation methods; they defined several criteria in order to discuss the efficiency of the five methods they dealt with. The criteria correspond to the type of segmentation, the extracting of the correct segment, the boundaries, the hierarchical / multi-scale segmentation, the sensitivity to the pose, the asymptotic complexity and the control parameters.

In the following, we propose an original part-type segmentation based on watershed transformation using markers. This approach uses a connected vertices structure and the flooding is realised from the vertices curvature information. Vertex based approach [4] is commonly used but face based or edge based approaches can be dealt with also. The watershed transformation using markers is a powerful tool to segment an object. Markers are used to labelise obvious regions and the watershed transformation joins them in respect to the curvature. The automatic search of the best markers is a difficult problem and that represents the essential of our contribution.

2. WATERSHED AND MARKERS

Our method uses the watershed transformation with markers on a connected vertex structure. The markers are automatically determined from the skeleton of the model and represent the most meaningful parts. The watershed transformation entirely decomposes the mesh from marker information. 3D watershed transformation simulates water rising on the curvatures of the input mesh from the local minima or markers. The mesh can be seen as a map where vertices curvature is replaced by a height. The water rises in each basin and when the water from two basins meet, a watershed is created between them (Fig. 1). At the end of the process, all the basins are surrounded by watersheds. Our watershed algorithm derives inspiration from the unbiased implementation of the watershed transformation based on hierarchical queues [5] for computing the hierarchical queue watershed transformation on meshes.

1-4244-1437-7/07/$20.00 ©2007 IEEE

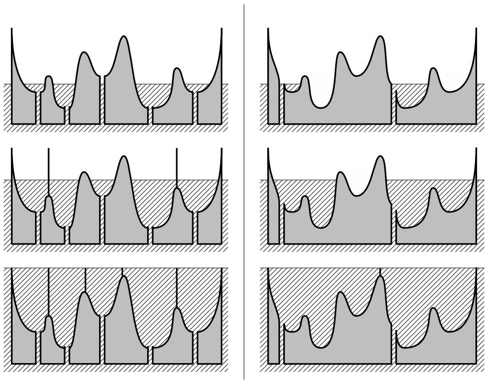

Fig. 1. A one dimensional example of watershed transformation. Three different water-levels are proposed for the watershed with minima at the left and watershed with markers at the right. The water-level stops rising as soon as all the basins are entirely filled. A watershed is built when the water from adjacent basins meet.

The curvature seems to be the best way to characterize the height and then the boundaries in a structure of connected vertices. We use the method proposed by Mangan and Whitaker [4] to calculate the curvature from the norm of the covariance matrix. This method stands on a statistical concept which consists in assessing vertex coordinate variance and covariance in the neighborhood of the considered vertex. The covariance matrix is given by the variance and covariance in all three directions:

$$\sigma_{uu}^2 = \frac{1}{N}\sum_{i=0}^{N}(u_i-\bar{u})^2 \tag{1}$$

$$\sigma_{uv}^2 = \frac{1}{N}\sum_{i=0}^{N}(u_i-\bar{u})(v_i-\bar{v}) \tag{2}$$

$$C = \|M\| \text{ with } M = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{3}$$

where σ_{uu} corresponds to the u coordinates standard deviation in the element neighborhood, and σ_{uv} corresponds to the square root of the covariance between u and v coordinates. N corresponds to the number of triangles associated with this element. The curvature C is defined by the norm of the covariance matrix M.

The simple process of the watershed transformation produces too many regions but this over-segmentation can be limited by a hierarchical process or markers. Hierarchical segmentation allows defining several levels of segmentation from the result of the watershed. The most famous method is the Waterfalls, developed by Beucher [6]. The markers can be used if the main parts of the objects are known and in this case, they offer the best way to reach the desired segmentation. Relevant markers are difficult to find. We explain in the next section how we compute theses markers from the skeleton of the model.

3. MARKERS COMPUTATION

The skeleton of the model gives an estimation of the parts of the objects. Our plan is to associate vertices of the mesh with their corresponding edge of the skeleton graph. Some vertices become markers because they are linked to only one edge. Vertices which belong to several edges could be labelised by the watershed. Our approach transforms the mesh in a voxel representation and involves several processes such as the voxelisation and the skeletonization based on the following methods.

Brunner and Brunnet [7] used an efficient process to store the voxels and to compress the 3D grid structure. A two dimensional array is sufficient to conserve all surface voxel information. Each element of this array can have several couples of voxels. Theses voxels are associated to the input and the output of rays traced from the array and which cross the object (Figure 2).

Karabassi *et al.* [8] proposed a very fast voxelisation algorithm based on six Z-buffers. Each Z-buffer is associated to a view and stores the coordinates of the closest voxels. This method does not consider internal or hidden parts of the object.

To compute the voxel skeleton, we developed a method based on the Palágyi algorithm [9]. Our approach removes successively voxels from six directions (up, bottom, north, south, east and west) in 3 sub-iterations. At each iteration of the process, all surface voxels are tested to be simple or not. A simple point (voxel) is a point which removal does not change the topology of the object. Such a point has the property that its removal does not create a hole, does not create a cavity and does not disconnect a connected component.

The voxels of the skeleton graph are associated to two categories from the number of their neighbours. A voxel belong to an edge if it has one or two neighbours. A voxel belong to a junction if it has more than two neighbours. Each edge

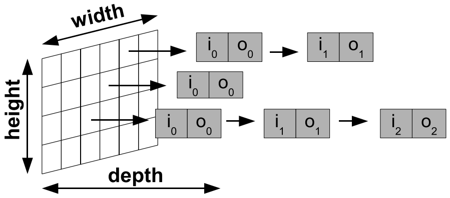

Fig. 2. A graphical view of the data structure holding the pair. Each pair contains the coordinates (input i and output o) to specify internal segments which belong to the object.

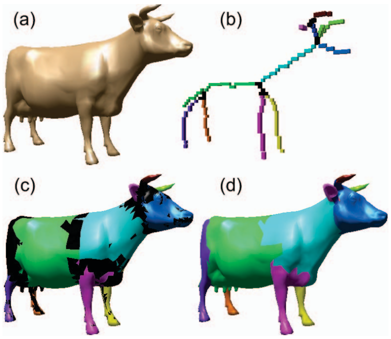

Fig. 3. The different steps of the segmentation process: (a) The Cow model, (b) its skeleton, (c) markers computed from edges of the skeleton graph and (d) the segmented object.

obtains a different label and their corresponding voxels are labelled from this label as shows in figure 3. During the voxelisation process, a link is created between the face of the mesh and their corresponding voxels. When a voxel is removed during the skeletonization process, its links to the faces are transferred to the voxels of the lower layer. Each voxel of the skeleton becomes linked to several faces of the mesh. These faces are considered as markers and obtain the label of the corresponding edge of the skeleton (Figure 4).

4. DISCUSSIONS AND RESULTS

The figure 5 illustrates the results of our segmentation algorithm on different models. In the following, we discuss the criteria defined in section 1 and compare our approach to those studied in [3].

Type of segmentation: models presented here often appear in the mesh segmentation literature and correspond to human or animals. Among the different kinds of segmentation, some methods like clustering are more appropriated to decompose CAD or face models, while skeleton based approaches are fitted to human body models and animals for example.

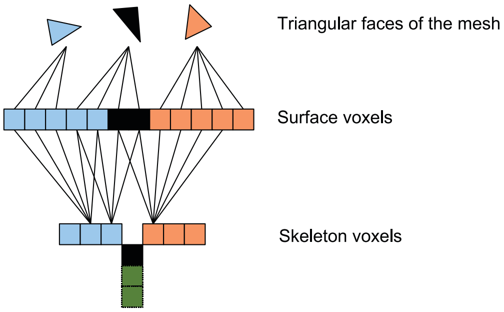

Fig. 4. Faces of the mesh are marked for the watershed transformation. Ambiguous voxels of the surface are not labelised and their corresponding faces are not considered as markers.

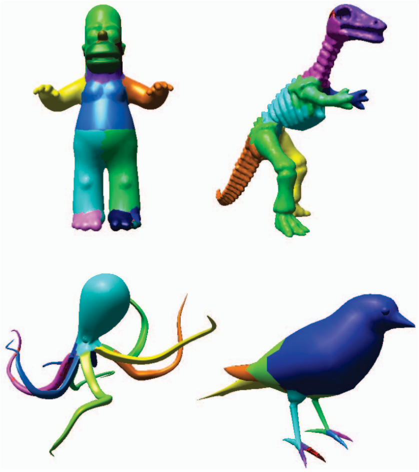

Fig. 5. Segmentation of animal and humanoid models.

Extracting the correct segments: the shape of the skeleton of the model depends on the resolution of the voxelisation. A high resolution voxelisation produces too many details and insignificant skeleton edges. A low resolution voxelisation allows a fast process computation and a simplified skeleton with meaningful edges only. Our experimentations show that a resolution of 100 for the largest dimension of the model is enough to produce a meaningful skeleton. The Homer model is segmented in the same manner as in [3] but additional parts like the nose and the mouth are segmented. The minimal length of edges can be configured in order to ignore this kind of details for example. The surface of the dinosaur is noised but the low resolution of the voxelisation prevents the creation of small edges.

Boundaries: the main parts are determined from the skeletonization process and the boundaries are created by the watershed transformation from markers. Areas which are not labelised correspond to junction areas and have more or less important curvature. The boundaries are made in respect to the scheme of the watershed process and regions meet at their highest area curvature. A minimal cut process [2] can be added to produce smooth boundaries.

Hierarchical segmentation: the number of details can be controlled from the minimal length of skeleton edges. Considering the most significant parts are the largest length of skeleton edge, the hierarchical process corresponds in our case to build a hierarchy of parts where each level corresponds to a minimal length of skeleton edges. The processes of vertex marking and watershed transformation have to be repeated for each level. Another approach considering the skeleton graph like a tree can be used. The hierarchical segmentation process decomposes the tree in levels and associates each level to a skeleton clustering scheme.

Sensitivity to the pose: the skeleton of the model allows a pose-invariant segmentation with some restrictions: the voxelisation and the skeletonization of the same model with different poses can lead to different levels of details because new skeleton edges can be created in some cases. Different parts which are too close can be merged during the voxelisation process; Brunner and Brunnet [7] explain how to prevent this problem with the consideration of local and global neighbourhood.

Calculation time: All performance measurements for the segmentation algorithm were made on a 2.8Ghz Intel Pentium IV system. The running time of the system has been decomposed into two steps. First step is the marking process, where the voxelisation, the skeletonization and the labelisation of marker faces are computed. Second step is the final labelisation with the watershed transformation initialised from markers. The computation time of the marking process and the watershed transformation depends on the number of voxels for the first step and the number of vertices for the second step. Our method allows a fast marking even for high resolution models as shows on table 1. The running time of the watershed process is directly linked to the number of vertices but it can be reduced considering only the markers not entirely surrounded by markers. For each model, the voxel resolution corresponds to 100 for the largest dimension of the 3D object.

Models	Number of vertices	Volume in voxels	Calculation time (s.)	
			Marking	Watershed
Bird	1129	42848	12.5	0.011
Cow	2903	54923	15.6	0.062
Homer	5103	43244	9.6	0.192
Octopus	16944	18250	4.2	5
Dinosaur	42146	21219	5.2	9.25

Table 1. Calculation times of the marking process and the watershed transformation.

Control parameter: two parameters are defined by the user. The first parameter controls the resolution of the 3D grid and the second controls the minimal length of skeleton edges allowed.

5. CONCLUSION

We have presented a new method to compute automatically the makers for the 3D watershed transformation. The markers allow to labelise important area; ambiguous areas, such as the junctions, are left to the watershed transformation. The markers are identified from the edges of the skeleton graph of the model and are associated to the faces of the 3D mesh. The segmentation approach extract the skeleton graph defined in a low resolution 3D grid, so the process is fast and only the main features of the object are determined, even if the mesh has a high number of vertices. The method has been evaluated from several criteria and offers a correct segmentation of the model parts. Future studies will focus on the skeleton graph of the object in the purpose of interpreting semantical information.

Acknowledgements

We are grateful to AIM@SHAPE (www.aimatshape.net) and Cyberware (www.cyberware.com) for making their models available online. We would like to thank Gregory Pageot for his help in this work.

6. REFERENCES

[1] Ariel Shamir, “A formulation of boundary mesh segmentation,” in *Proceedings of the second International Symposium on 3DPVT*, Greece, September 2004, pp. 82–89.

[2] Sagi Katz, George Leifman, and Ayellet Tal, “Mesh segmentation using feature point and core extraction.,” *The Visual Computer*, vol. 21, no. 8-10, pp. 649–658, 2005.

[3] Marco Attene, Sagi Kat, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal, “Mesh segmentation - a comparative study,” in *International Conference on Shape Modeling and Applications 2006*, Washington, 2006, pp. 14–25.

[4] Alan P. Mangan and Ross T. Whitaker, “Partitioning 3d surface meshes using watershed segmentation,” *IEEE Transactions On Visualization And Computer Graphics*, vol. 5, no. 4, pp. 308–321, October-December 1999.

[5] Serge Beucher, “Algorithmes sans biais de ligne de partage des eaux,” Tech. Rep., Centre de Morphologie Mathématique de l’École des Mines de Paris, Avril 2004.

[6] Serge Beucher, “Watershed, hierarchical segmentation and waterfall algorithm,” in *Mathematical morphology and its applications to image processing*, J. Serra and P. Soille, Eds., pp. 69–76. 1994.

[7] David Brunner and Guido Brunnett, “An extended concept of voxel neighborhoods for correct thinning in mesh segmentation,” *Proceedings of the 21st Spring Conference on Computer Graphics*, pp. 119 – 125, 2005.

[8] Evaggelia-Aggeliki Karabassi, Georgios Papaioannou, and Theoharis Theoharis, “A fast depth-buffer-based voxelization algorithm,” *J. Graph. Tools*, vol. 4, no. 4, pp. 5–10, 1999.

[9] Kálmán Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdhelyi, and K. Hausegger, “A sequential 3d thinning algorithm and its medical applications,” in *in Proc. 17th Int. Conf. Information Processing in Medical Imaging*, 2001.