# 3D Mesh Segmentation Methodologies for CAD applications

Alexander Agathos[1], Ioannis Pratikakis[1], Stavros Perantonis[1], Nikolaos Sapidis[2] and Philip Azariadis[2]

[1]National Center for Scientific Research "Demokritos" {agalex, ipratika, sper}@iit.demokritos.gr
[2]University of the Aegean {sapidis, azar}@aegean.gr

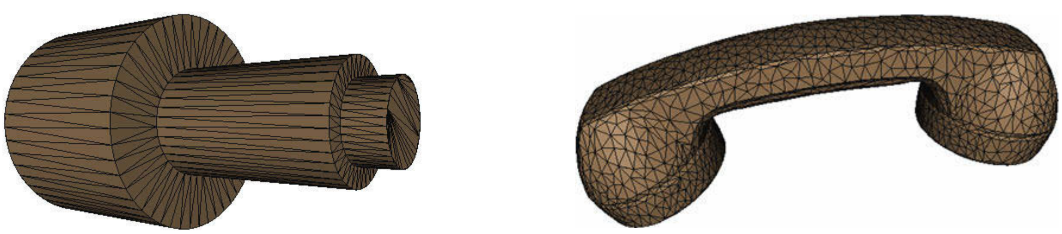
**ABSTRACT**

3D mesh segmentation is a fundamental process for Digital Shape Reconstruction in a variety of applications including Reverse Engineering, Medical Imaging, etc. It is used to provide a high level representation of the raw 3D data which is required for CAD, CAM and CAE. In this paper, we present an exhaustive overview of 3D mesh segmentation methodologies examining their suitability for CAD models. In particular, a classification of the various methods is given based on their corresponding underlying fundamental methodology concept as well as on the distinct criteria and features used in the segmentation process.

**Keywords:** 3D objects, mesh segmentation, shape analysis.

## 1. INTRODUCTION

3D Mesh Segmentation is a crucial step in the pipeline of Digital Shape Reconstruction, for which the exploitation of high level semantics in 3D raw data is addressed, enabling a semantic richness useful for CAD, CAM and CAE purposes. Usually a 3D physical object is sampled with a laser scanner and the output is a set of 3D points which represent the surface of the object. This set is triangulated acquiring by this way a topological structure of the surface which is called a 3D mesh. A 3D mesh (Fig. 1) contains unorganized information (triangles, edges, points) which should be further processed in order to be useful for CAD purposes. There are mainly three types of CAD models. The first type consists of those engineering objects which contain surfaces of known geometry (e.g. planes, spheres, cylinders, surfaces of revolution, etc). The second type consists of those engineering objects which contain free-form surfaces. The third type is a combination of the first and second type. In Fig. 1(a), an example is given that shows a 3D mesh that represents a CAD model of the first type while in Fig. 1(b) an example is given that shows the 3D mesh of a CAD model of the second type.



(a) (b)

Fig. 1: 3D meshes of CAD models. (a) Example of a model that consists of surfaces with known geometry; (b) Example of a model that consists of free-form surfaces.

Segmentation is the process which provides the necessary organization of the data points by partitioning them into connected regions or parts that can be approximated by standard CAD surfaces (e.g. planes, cylinders, etc.) or volumetric primitives (e.g. super-ellipsoids). There is a variety of algorithms for 3D mesh segmentation, which can be grouped in two basic categories:

- Surface-based: The 3D mesh is segmented into regions which represent distinct surfaces of the CAD model and can be approximated by various primitives like planes, cylinders, spheres, polynomials, etc.
- Part-based: The 3D mesh is segmented into volumetric parts which can be approximated by volumetric primitives (e.g. super-ellipsoids).

The quality of segmentation is a crucial issue that is directly related to the corresponding application which imposes particular requirements. For surface-based algorithms it is usually required that (i) the boundaries of the segmented regions should be smooth; (ii) the extracted regions should be able to be approximated by smooth surfaces; and (iii) the boundaries where the regions meet should allow certain types of continuity (like $C^1$ or $C^2$) to hold for the approximating surfaces. For part-based algorithms a variety of criteria can be used in order to be able to extract the meaningful parts of the object. The most used criterion is the *minima rule* introduced by Hoffman and Richards [12] which states that an object is segmented by human perception at areas of concavity. This criterion has been adopted by many researchers [17-18], [22-23], [27], [30-31], [46-47] to segment 3D meshes. The quality of segmentation is also directly dependent on the type of the CAD object that is being processed. Different algorithms work better for each CAD object type.

In this paper, an exhaustive overview for 3D mesh segmentation is presented. The aim of this paper is twofold: (i) to categorize and outline these algorithms and (ii) to pinpoint advantages and disadvantages.

The remainder of this paper is structured as follows: In Section 2, a formal definition of the segmentation process is introduced as well as other definitions that facilitate clarity of method's description. In Section 3, a classification of the various algorithms for mesh segmentation is presented along with their corresponding description. Finally, in Section 4, conclusions are drawn.

## 2. DEFINITIONS

In this section, we introduce a set of definitions that are appropriate for the understanding of the methods that will be described. Let $S$ be the set of either the vertices, edges or faces of the mesh. *Segmentation* of a mesh $M$ is called the partitioning of $S$ into $k$-disjoint connected sets, i.e:

$$\bigcup_{i=1}^{k} S_i = S, \quad S_i \subset S, \quad S_i \cap S_j = \varnothing, \quad i,j = 1...k, \ i \neq j \tag{0.1}$$

The segmentation of a mesh depends on criteria which dictate which elements of S belong to the same partition. These criteria are built upon the segmentation objective which in turn depends on the application / context. However, there seem to be common constituents in the general objectives of many segmentation algorithms, *see* Shamir [38, 39].

A widely used function introduced by Hilaga et. al. [15] represents the magnitude of a protrusion on the surface $M$. The protrusion function on a point $u$ on $M$ is the sum of its distances from all surface points, defined as:

$$m(u) = \int_{p \in \mathrm{M}} g(u,p) \mathrm{dM} \tag{0.2}$$

where $g(u,p)$ denotes the geodesic distance of the surface points $u, p$. Large values of $m$ denote that $u$ belongs to a protrusion while small values denote that $u$ is near the center of the surface, *see* Fig. 2.

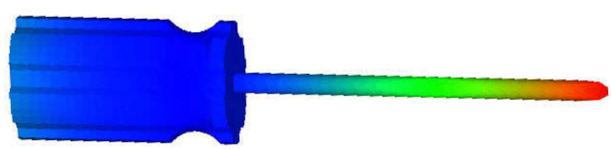


Fig. 2: Chromatic representation of the protrusion function, blue color denotes small values of m, red denotes large values and green denotes in between values.

The *dual graph* of a mesh is the graph for which nodes are the triangular faces and each node is connected with another if their representative triangles share a common edge. The triangles, which represent its nodes, will be referred as the *dual triangles*, *see* Fig. 3. For the sake of clarity, from now on, the parts that the object is segmented into will be called *segmentation areas*. The boundaries where segmentation areas meet will be called *segmentation boundaries.*

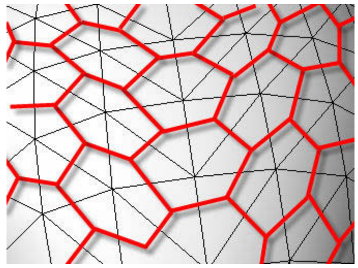


Fig. 3: The dual graph a mesh denoted by red lines.

## 3. CATEGORIZATION OF SEGMENTATION METHODS

In the literature, there are several algorithms for 3D mesh segmentation. These algorithms can be grouped into categories according to their core methodology. In our classification, we have identified the following categories: (i) Region growing; (ii) Watershed-based; (iii) Reeb graphs; (iv) Model-based; (v) Skeleton-based; (vi) Clustering; (vii) Spectral analysis; (viii) Explicit Boundary Extraction; (ix) Critical points-based; (x) Multiscale Shape Descriptors; (xi) Markov Random Fields and (xii) Direct segmentation. A detailed analysis of the aforementioned categories is given in the following.

### 3.1 Region Growing

In this technique, segmentation areas are generated with the expansion of distinct *seed* elements (points, triangles, regions). In region growing, a set of rules is applied which will either permit or stop growing. In the literature, this set of rules is called either termination or growing criteria.

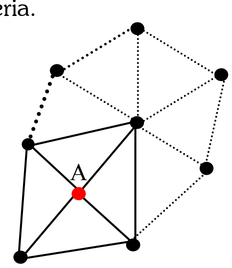



Fig. 4: Point A “grows” adding its neighbors in the segmentation area it belongs to.

For example, the segmentation area label of point A in Fig. 4 is attributed to its neighbors, thus expanding the segmentation area that it belongs to. This process is subsequently repeated until a termination criterion is satisfied. In this category, both surface-based and part-based algorithms can be found:

#### *3.1.1 Surface-based Algorithms*

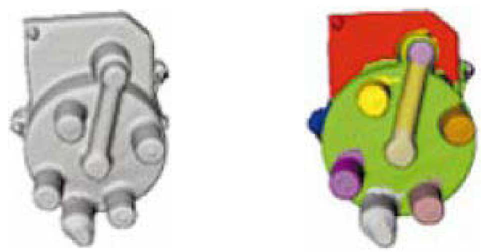
Besl and Jain [6] perform region growing in range images. They initially label the data points using the mean and Gaussian curvature. Using this labeling, they construct seed regions from which region growing will occur. Region growing is accomplished by fitting variable order bi-variate polynomials to the growing region and neighboring data points are added according to their compatibility with the approximating polynomial. Here the termination criterion is satisfied when the region can’t grow any further. Their algorithm has been later extended by Vieira and Shimada [42] to 3D meshes.

Sapidis and Besl [36] also perform region growing in range image data which are segmented in $N$ quasi-disjoint regions that can be approximated by $N$ polynomial functions in a way that (i) the distance between a point of the region and its associated polynomial is smaller than a threshold; (ii) the total number of data elements contained in each region is maximized and (iii) $C^1$ or $C^2$ continuity can be enforced between adjacent polynomials. Region growing starts with a seed region to which a polynomial is fitted. This region grows according to a distance and orientation criterion and the approximating polynomial of the expanded region is computed. The process continues until there can be no further expansion. Here the termination criterion is satisfied when there can be no further growth of the region within a certain polynomial fitting tolerance. A strong advantage of their algorithm is that the approximation surface of the segmented part and the part itself is found simultaneously while conforming to the constraints that the authors have set.

Lavoue et. al. [20] classify the vertices according to the absolute values of their principal curvatures into $k$-clusters using the $k$-means algorithm, where $k$ is user defined. They also detect sharp edges (edges for which the dihedral angle of the incident triangles exceeds a threshold) and sharp vertices (vertices belonging to a sharp edge). The growing criterion in this case adds to the region neighboring triangles which do not share a sharp edge and all of their vertices are either sharp or belong to the same cluster.

*3.1.2. Part-based Algorithms*

Zhang et. al. [46] label the points according to their Gaussian curvature. Following the minima rule, by thresholding, they label points of high negative Gaussian curvature as boundary points. Then, by selecting randomly seed points among the non-boundary points they proceed in region growing. The growing criterion here does not add neighboring points which belong to the boundary or those that are already added to the region. Results of this algorithm are shown in Fig. 5. An advantage of this method is that it is very simple to implement. One of the disadvantages is that the output depends heavily on the choice of the user defined threshold.



(a) (b)

Fig. 5: (a) A 3D mesh of a distributor cap; (b) its corresponding segmentation (taken from [46]).

Zuckerberger et al [48] extract the segmentation areas with a flooding of the dual graph. Specifically, their segmentation technique begins from a node traversing the graph using a Depth First or Breadth First Search. This traversal incrementally collects faces as long as they form a convex segmentation area. The termination criterion here is satisfied when convexity is violated. In the case of convexity violation, a new segmentation area is created from an unvisited node and the traversal is resumed. The main problem with this approach is over-segmentation which the authors propose to tackle by favoring merging of smaller segmentation areas to neighboring larger ones.

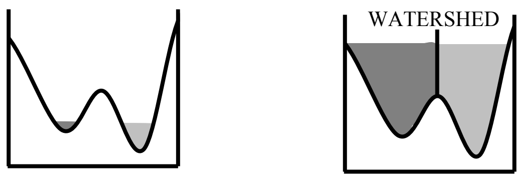
Leonardis et. al. [24] use the region growing method and the recover-and-select paradigm in order to recover volumetric parametric primitives (superquadrics) from range data. The segmentation process consists of two stages which are combined iteratively : (i) Model Recovery stage : Seed areas of points are selected from the range image to which Superquadrics are fitted. These elements are expanded adding neighbouring points and a new superquadric is fitted. The growing criterion for the model expansion is based on an average error-of-fit measure which uses the radial Euclidian distance metric. (ii) Model-selection which, as the name implies, selects those models that produce the simplest description, i.e. it selects the minimal number of superquadrics which describe the data while keeping low the deviations between the data points and the chosen superquadrics. The iteration between (i) and (ii) is repeated until the remaining superquadrics are completely recovered. It has to be noticed that the segmentation areas created in the region growing process are not mutually exclusive.

**3.2 Watershed-based**

In this category, the segmentation is achieved in analogy to the way that water fills a geographic surface (Fig. 6). As the water floods its basins there will be points where flood regions meet. These points construct the watershed lines that divide the surface into distinct regions. The flooded basins are called catchment basins and they represent segmentation areas.

Initially, the watershed transformation was used for 2D/3D Image segmentation (*see* for example [33], [43]). Mangan and Whitaker [28] were the first to extend it to 3D meshes. For segmentation purposes, the watershed algorithm is not applied to the original 3D Mesh but it is applied to a transformed version based on a watershed function $f : \Re^3 \rightarrow \Re$ that guides the watersheds to identify the crest lines of the mesh. There is a one-to-one correspondence between the minima of $f$ and the catchment basins.

From the algorithmic point of view, the extraction of catchment basins can be achieved in two ways: (i) *Steepest Path* Following, where a point on the surface follows its steepest path and slides to a local minimum of *f*. This point and the followed path belong to the segmentation area that corresponds to the minima of the catchment basin; (ii) *Flooding*, where the algorithm is flooding its topographic relief starting from its local minima. One of the main disadvantages of the watershed segmentation is that most of the time it leads to over-segmentation. One solution to deal with this is by merging areas based upon different criteria.




(a) (b)

Fig. 6: (a) Initial flooding of the geographic surface. (b) Watershed line emerged at a certain flooding level.

As in the previous category, both surface-based and part-based algorithms have been addressed:

*3.2.1 Surface-based Algorithms*

The majority of the algorithms use curvature as the watershed function: In [34] various types of curvature have been used like the Gaussian, the Mean, the Absolute, etc. In [28] the square root of the "deviation from flatness" has been used ( $D = \sqrt{k_1^2 + k_2^2}$ , $k_1$, $k_2$ principal curvatures). The basic drawback of the above approaches is that the segmentation boundaries are not smooth as well as that they can only segment at regions of high curvature.

Zuckerberger et. al. [48] use the edges of the triangles to flow to the minimum. By this method, a better segmentation is achieved in some cases where the vertices of the input mesh might not be dense enough to define the catchment basins. In this work, the dihedral angle of adjacent to the edges triangles is used as a watershed function. The main disadvantage of this method is that due to the small local support of the dihedral angle it is susceptible to noise, thus leading to oversegmentation.

(a) (b)

Fig. 7: (a) A "cup" model after thresholding and application of the morphological operators on the minimum curvature (the points above the threshold are shown with black color while points below the threshold are shown in white). (b) The segmentation of the cup is illustrated with colours denoting different regions (taken from [30]).

*3.2.2. Part-based Algorithms*

Page et. al. [30, 31] implemented a scheme that diminishes the problem of over-segmentation. In this scheme, the basins are filled until a certain point which is determined by thresholding and the application of 3D morphological operators [35] on the minimum curvature. Afterwards, the basins continue to be filled with a flooding watershed algorithm using as watershed function the normal curvature. The process is illustrated in Fig. 7. As can be seen in Fig. 7(b) the segmentation boundaries in this method may not be smooth.

### 3.3 Reeb Graphs

In this method, a graph structure called reeb graph is used for segmentation. Let $f : M \subseteq \Re^3 \rightarrow \Re$ , where $M$ is a three dimensional surface. The Reeb graph of the surface $M$ is the quotient space [1] of $f$ in $M \times \Re$ generated using the following equivalence relation:

$$(u_1, f(u_1)) \sim (u_2, f(u_2)) \Leftrightarrow \begin{cases} f(u_1) = f(u_2) \\ \& \\ u_1, u_2 \text{ lie in the same connected component of} \\ f^{-1}(f(u_1)) \end{cases} \tag{0.3}$$

If for example, $f$ is a height function which returns the $z$ coordinate of a point on a surface $M$ , then the equivalence classes of the Reeb Graph of a surface $M$ are contours perpendicular to the height direction, *see* Fig. 8.

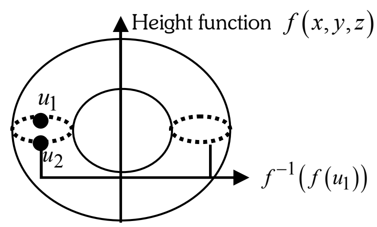



Fig. 8: A torus object associated with the height function $f(x,y,z) = z$ . The two points $u_1$ , $u_2$ on the mesh belong to the same connected component, thus $(u_1, f(u_1))$ , $(u_2, f(u_2))$ are equivalent.

Antini et. al. [2] quantized the protrusion function in order to do segmentation. For example the quantization (in 7 bins) of the protrusion function (see section 2) of a human model is illustrated in Fig. 9(a). Based on this quantization a discrete Reeb graph can be constructed. Their segmentation objective is to acquire perceptually meaningful components. Antini et. al. managed to cope with over-segmentation by simplifying the Reeb Graph.

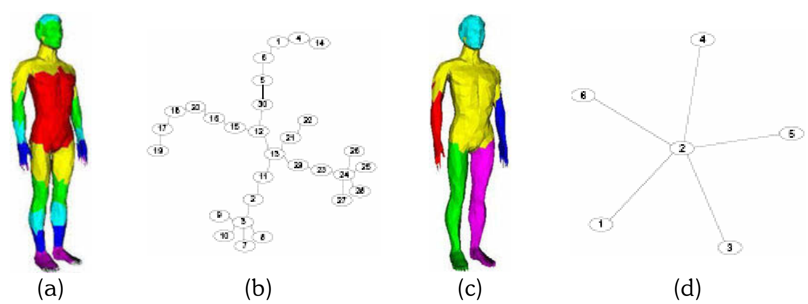



Fig. 9: (a) Quantization in 7 levels, (b) The Discrete Reeb Graph which corresponds to the quantization (c) The Segmentation areas after the simplification of the Reeb Graph (d) The Simplified Graph (taken from [2]).

While Antini et. al. manage to acquire perceptually meaningful components the boundaries between regions are not positioned at areas of concavities as expected by the minima rule. Although their algorithm is not used to segment CAD objects it could be considered for part-based CAD model segmentation due to their Reeb-graph simplification process which groups together parts with similar geometric characteristics.

### 3.4 Model-based

In this method, a surface model of the mesh is constructed in order to achieve segmentation. Wu and Levine [45] use the distribution of electrical charge across the surface in order to do segmentation. Specifically they used the property that the density charge is very low at areas of deep concavities of the surface and very high at areas of high convexities of the surface. According to the minima rule, segmentation areas will reside on areas of the mesh where minima in the density charge occur. Their algorithm trace contours on the object where minima in the local density charge exist. These contours are the segmentation boundaries.

One of the advantages of this method is that it uses a global characterization of the surface to find concavities while most other methods use local characterization like curvature. This makes it less sensitive to noise. Disadvantages of this method is that it can only trace segmentation boundaries surrounded by concavities and that it can only work on surfaces which don't have any other concave areas except from those that contain segmentation areas. Their segmentation algorithm belongs to part-based algorithms and can be used to extract parts which can be represented by volumetric primitives useful in CAD applications like object recognition.

### 3.5 Skeleton-based

In this method, the skeleton of the object guides the segmentation algorithm. In Li et. al. [25], the skeleton of the object is constructed by performing simplification of the surface using the edge contraction method [14]. The segmentation areas of the object are extracted by using a plane which sweeps the mesh along the skeleton edges. The intersection of the plane and the mesh consists of one or more polygon contours. The segmentation areas are extracted by examining the way that these contours alter as the sweep plane moves. For this purpose, a parametric geometric and topological function is defined on these contours. The contours which contain critical points of these functions signify segmentation boundaries.

A disadvantage of this method is that due to the application of smoothing filters on the parametric functions there can be cases where small scale structure disappear. This segmentation algorithm is part-based.

### 3.6 Clustering

In this category, segmentation is achieved either by an iterative clustering approach where a k-means algorithm is applied or by a hierarchy of cluster elements (triangles. points, etc.) approximated by a primitive (plane, sphere, etc.). Similar to previous approaches, surface-based and part-based algorithms can be found.

*3.6.1. Surface-based Algorithms*

Garland et al [10] propose a hierarchical clustering methodology suitable for applications that use 3D models with close to planar segmentation areas. Specifically, they do edge contraction on the dual graph of the mesh. The edge contraction cost is based on planarity, orientation bias and shape bias measures. The contraction sequence is settled by a priority queue which sorts the dual edges of the dual graph according to the contraction cost. This algorithm can be used for CAD model segmentation where planar regions should be extracted.

Inoue et al [16] use a clustering algorithm similar to Garland [10] which is used to simplify the CAD model structure. The edge contraction cost is based on the following criteria : (i) area size - The area of a region should be large enough with respect to a mesh element area; (ii) border smoothness - The boundary of a region should be as smooth as possible; (iii) region flatness - Each region should be as flat as possible.

Attene et al [3] use the same hierarchical clustering as in [10] to classify the mesh triangles into clusters that can be approximated by a plane or sphere or cylinder primitive. The contraction cost of a dual edge is the minimum of the errors generated after fitting the plane, sphere and cylinder primitives to the triangles of the merged regions. The primitive corresponding to the minimum fitting error approximates the cluster. Their algorithm can provide a hierarchy of regions approximating planes, spheres and cylinders. It can give a high-level structure to the model useful for semantic interpretation purposes. The only drawback is that it approximates only a narrow range of surfaces (planes, spheres, cylinders).

Gelfand and Guibas [11] base their hierarchical clustering technique on slippable motions. These are rigid motions which, when applied to the shape, slide their transformed version against the stationery version without forming any gaps. This happens when the instantaneous motion at each point is tangent to the mesh. A kinematic surface is a surface which has a slippable motion. A slippable component is a collection of vertices from the mesh which can be approximated by a kinematic surface. Their algorithm segments the input surface into a set of slippable components using a hierarchical clustering technique similar to [10] on the point set of the mesh. Their segmentation algorithm can be used in CAD model applications in order to extract planes, spheres, cylinders, linear extrusion surfaces, surfaces of revolution and helical surfaces.

*3.6.2. Part-based Algorithms*

Shlafman et al [41] use the $k$-means clustering algorithm for segmentation. Their segmentation scheme clusters triangles and proceeds as follows: (i) compute the distances between each triangle pair; (ii) compute the triangle cluster representatives; (iii) assign all other triangles to their nearest representative and (iv) update the representatives by minimizing the sum of distances between the triangles belonging to the cluster and the representative. The distance function between each pair of adjacent triangles is a weighted sum of the geodesic distance of the barycenters and the angle between the triangles.

Katz and Tal [17] use a hierarchical fuzzy $k$-means algorithm to segment the object. As in Shlafman et. al. [41], their algorithm clusters triangles with a distance function which is a weighted sum of the geodesic distance of the barycenters and the angle between the triangles. In the binary partitioning case, their algorithm partitions the mesh into two segmentation areas and a fuzzy area. The fuzzy area contains triangles that should be assigned to one of the two segmentation areas. This decision is made by a minimum cut algorithm, which cuts the fuzzy area into two regions separated by deep concavities. A segmentation example is shown in Fig. 10. The advantage of this method is that it segments the mesh into meaningful components and at deep concavities, which conforms to the minima rule (*see* Section 2). A disadvantage is that when the segmentation boundary is not surrounded by concavity, the minimal cut may not always find the optimal segmentation boundary.

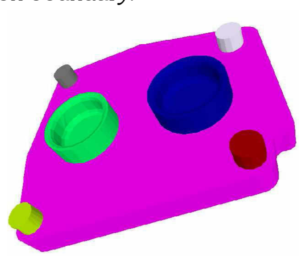


Fig. 10: Segmentation of a mechanical part (taken from [17]).

It can be noticed that the clustering and region growing methods are rather similar in nature. In the iterative clustering approach the main difference is that the representatives get refined using a minimization function while in the region growing method the seed elements are selected only at the beginning of the algorithm and do not change. Also, the criterion which assigns an element of the mesh to a segmentation area is based on its distance from the representative in the iterative clustering approach while in the region growing approach the criterion is independent of the seed element. In the hierarchical clustering approach all elements can be regarded as seed elements at the first level of the hierarchy and then are merged at the finer level according to a criterion while in the region growing approach a few elements are regarded as seed elements which then expand adding non seed elements according to a criterion.

### 3.7 Spectral Analysis

The spectral analysis method uses the eigenvalues of properly defined matrices based on the connectivity of the graph in order to partition a mesh.

Liu and Zhang [27] use spectral analysis on the dual graph of the mesh. They define an affinity matrix $\mathbf{W}$ with elements $0 \leq w_{ij} \leq 1$ denoting the likelihood that faces $i, j$ can be clustered into the same segmentation area. These elements are created using the distance function of Katz et. al. [17] and an exponential kernel. This type of matrix has been used successfully for clustering since it groups elements having high affinity, see for example [44]. Thus, using this matrix with distances defined by Katz et. al. [17], a meaningful segmentation can be achieved. Following the Polarization Theorem [7] and similar to [37], they construct a matrix $\mathbf{V}$ whose columns are the $k$ largest eigenvectors of the normalized affinity matrix $\mathbf{W}_n$ and normalize each of the rows. Then they apply the $k$-Means clustering on the row vectors of **V** using the standard Euclidian distance. Each of the above clustered vectors corresponds to the faces of the mesh, leading to segmentation. A segmentation example is shown in Fig. 11(a). One of the disadvantages of their algorithm is that they emphasize mostly in concavity. This means that they segment sometimes shallow concavities making the quality of segmentation of some areas low.

Zhang and Liu [47] use the affinity matrix as in Liu et. al. [27] to classify the mesh triangles into two segmentation areas in the same spirit as was done by Shi and Malik [40] in their Normalized Cut algorithm. In order to lower the complexity the affinity matrix is partially defined. Specifically it is constructed using the distances of two faces from all other faces. These two faces belong to perceptually different parts of the mesh. The two leading eigenvectors of this partial defined affinity matrix are extracted using Nyströms method [9] and are used in a line search algorithm to find the most salient cut which bipartitions the mesh. A segmentation example is shown in Fig. 11(b).

Both of the above algorithms are part-based and can be used for CAD applications.

(a) (b)

Fig. 11: (a) The segmentation of a toy plane using [27] (taken from [27]). (b) Segmentation of a mechanical part using [47] (taken from [47]).

### 3.8 Explicit Boundary Extraction

In this method, the contours of the segmentation boundaries are extracted. Lee et. al. [22-23] explicitly extract the segmentation boundaries of the mesh. Specifically, they find feature contours on the mesh based on the minimum negative curvature. Afterwards, they select some of these contours and convert them to closed contours which are the segmentation boundaries. The selection is based on the protrusion function and the contours are closed by finding the shortest path on the mesh using as edge cost a weighted combination of a distance, normal, centricity and feature function. The validity of each of the contours is tested against the saliency of the segmented part (see [13]), and is further refined using a geometric snake [21]. Fig. 12 illustrates the whole process.

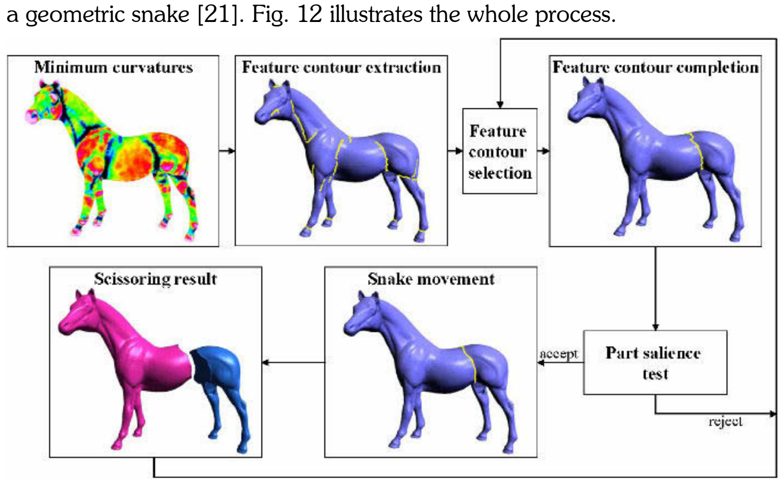



Fig. 12: Lee et al Boundary Extraction method (taken from [23]).

One of the main advantages of this method is that it produces smooth and closed segmentation boundaries which also pass a salience test. Sometimes though, the accepted contour may not cut the mesh into meaningful components with a consequence of the need of a user to cut the mesh manually. It has to be noted also that the 3D object should be sufficiently sampled at its concavities in order for the algorithm to work properly. Their algorithm is part-based and they have shown its ability to segment CAD objects when the feature lines are restricted to sharp concave creases.

### 3.9 Critical Points-Based

Algorithms that follow this method use critical points defined on the mesh to guide segmentation. These critical points are salient features of the 3D mesh and are used to aid the segmentation process to distinct between the different protrusions of the mesh.

Katz et. al. [18] presented a segmentation method based on critical points defined on the mesh. They first transform the mesh into a pose invariant representation using the multi-dimensional scaling method (MDS) [8]. Then they find on this representation critical points. These critical points are local maxima of the protrusion function that belong on the convex hull of the representation. Afterwards they create a surrounding sphere of the representation and use it to mirror its points so that vertices of the central part of the object become external points of the sphere. The central part of the object will now belong on the convex hull of this mirroring which they call it as core. The vertices of the core and their incident triangles are extracted, thus, isolating all the critical points (with the exception of those that are close to each other). At the end the segmentation is further refined by a graph cut method that cuts on the concavities of the object similarly to [17]. Their segmentation algorithm is part-based and they have demonstrated its ability to segment CAD objects in [4], see Fig. 13.

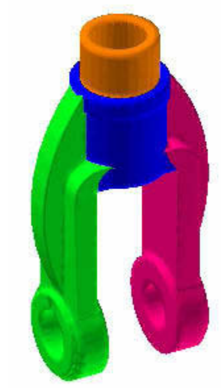


Fig. 13: Segmentation of a CAD object (taken from [4]).

Lin et al [26] define also critical points on the mesh. They call them salient representatives. These representatives are local maximums of the protrusion function defined on the dual graph of the mesh. Each representative creates geodesic zones which are called locales. These locales are used to define a border function which identifies those locales containing the border of the protrusion. After finding these locales the minimum cut algorithm of Katz and Tal [17] is used to separate the protrusion from the rest of the object. An advantage of their method is that it can identify the protrusions of a mesh even if they are very noisy. A disadvantage is that their algorithm works well only on meshes that have protrusions. Their segmentation algorithm is part-based and can be suitable to segment protrusions in CAD models.

### 3.10 Multiscale Shape Descriptors

In this category, multiscale shape descriptors are used to describe the mesh and extract components. Mortara et. al. [29] segment the mesh by describing descriptors of shape features at different scales. At each point they use a set of spheres of radii $R_i$ and intersect them with the mesh $M$ . The number of intersections characterizes the shape of the 3D neighborhood around a point on different scales. If there is one intersection then different features can be extracted, for example the point can be a tip or a pit or a mount or a dip. If there are two intersections then the feature can be a limb, a well, a joint, etc. If there are three or more intersections then the point is named a split. After labelling all the vertices with a shape feature, a flooding technique on the mesh is used in order to acquire segmentation areas corresponding to a shape feature at a given scale. This algorithm is surface-based and can be used for segmenting CAD models into feature regions at various scales.

### 3.11 Markov Random Fields

Pichler et. al. [32] used Markov Random Fields to acquire the segmentation of a range view scan. They first create a hierarchical Markov process in order to acquire noiseless shape index [19] values at the vertices. Afterwards they use Markov Random Fields for the extraction of contiguous convex areas (segmentation areas) which are partitioned by deep concavities. The minimization that this process requires uses the shape index and the curvature information. Their algorithm is part-based and can be used for CAD applications involving object recognition.

### 3.12 Direct Segmentation

Benko and Varady [5] introduced the Direct Segmentation Method in which the CAD model is segmented into smaller regions where the 3D points belonging on sharp or various sort of smooth edges are disregarded. These regions are later approximated by a hierarchy of surfaces. The hierarchy starts from simple surfaces like planes, spheres, cylinders, tori and move to more complicated ones like surfaces of linear extrusion and surfaces of revolution, ending with smooth multiple region surfaces (satisfying $G^1$ continuity) which are further subdivided until simpler surfaces of the hierarchy are found. Their algorithm is surface-based.

## 4. CONCLUSIONS

In this paper, we proposed a categorization of the existing 3D Mesh segmentation methods based upon their underlying fundamental methodology concept as well as on the distinct criteria and features used in the segmentation process. Furthermore, for each category we discuss the representations used for the segmented parts which can be either surface-based or part-based. The basic conclusions drawn in this paper will be discussed in the following for each representation, separately.

*Surface-based*: In the case of CAD models that consist of surfaces with known geometry the segmentation algorithms should be able to segment the surface into regions of known geometry (planes, spheres, cylinders etc.). Thus, the effectiveness of a segmentation algorithm for this type of models lies in its ability to discriminate between the different surfaces that exist in the model. The segmentation of this type of models gives also a direct high level representation to the object since it provides knowledge about the type of surfaces that the regions represent. The algorithms that can deal with this type of models are reported in [3, 5, 11]. In the case of CAD models which consist of free form surfaces there is a variety of ways that an algorithm can partition the mesh since the 3D object consists of surfaces of arbitrary form. The aim is to extract regions which can later be fitted by parametric surfaces, see for example [36]. In the case of CAD models which consist of both free form surfaces and surfaces of known geometry, one can adapt all methods related to the free form surface case. Concerning the proposed criteria set at the introduction, their importance depends on the particular application. There are not only cases in which it is desired to acquire smooth boundaries in order to achieve $C^1$ or $C^2$ continuity at the boundaries of the approximating primitives [36], but also there exist cases in which the focus is on the acquisition of points that belong to each region treating the extraction of boundaries as a post process [5].

*Part-based*: In part-based representation the main objective is to acquire volumetric parts. In this representation type the CAD model type is not important since the segmentation objective is not to extract the surfaces that exist on the surface but to acquire volumetric parts using various methods and criteria. From all algorithms, the most popular are those that use the minima-rule to segment the mesh. Therefore the measure of quality in this case lies in the algorithm effectiveness to utilize the minima rule [17-18], [22-23], [27], [30-31], [46-47]. The most crucial issue when dealing with the minima rule is noise. Most algorithms use dihedral angles or principal curvatures to detect concavities, which are susceptible to noise and can mislead them to detect false segmentation boundaries. Furthermore, in part-based representation a good partitioning is achieved when the segmentation boundaries are completely surrounded by deep concavities, otherwise, fuzzy segmentation boundaries can occur. An exception to this can be exemplified by the algorithm of Lee et. al. [22- 23].

For the sake of clarity, by means of Tab. 1, we present the criteria as well as the features used for each segmentation method which was analysed in Section 3. As it can be observed, there is a great diversity in the criteria and features used for segmentation. However, some trends towards segmentation can be discerned. It can be seen that the geodesic distance and dihedral angles are widely used by many algorithms. The protrusion function is becoming also very popular in acquiring meaningful components. Primitives like planes, spheres, cylinders, etc which approximate the segmentation areas are also very popular. Finally the curvature information is a basic foundation element of many segmentation algorithms.

From the overall analysis, it becomes evident that a successful partitioning depends on the application at hand which takes into account specific requirements. This means that a segmentation technique can produce meaningful results only if the corresponding features optimally support all required criteria suitable for the chosen application. Toward this end, the proposed structured overview on 3D mesh segmentation methodologies has the potential to provide the reader with meaningful hints for the selection of features and criteria given a particular CAD application.

<table>
<tr><th>METHOD</th><th>AUTHOR</th><th>FEATURES</th><th colspan="2">CRITERIA</th></tr>
<tr><td rowspan="6">Region Growing</td><td>Besl and Jain [6]</td><td>Variable-order approximating polynomials, ,mean and gaussian curvature</td><td colspan="2">Distance of points from the polynomial surface, derivatives estimates of points close to the polynomial surface derivatives</td></tr>
<tr><td>Sapidis and Besl [36]</td><td>Approximating polynomial surface, Normals of the point cloud</td><td colspan="2">Distance of points from the polynomial surface, Normal orientation of points compared with the direction of the Z axis</td></tr>
<tr><td>Lavoue et. al. [20]</td><td>Principal Curvatures</td><td colspan="2">A triangle is added to a region based on the clustering of the Principal Curvatures</td></tr>
<tr><td>Zhang et. al. [46]</td><td>Gaussian Curvature</td><td colspan="2">Gaussian Curvature value above a user defined threshold</td></tr>
<tr><td>Zuckerberger et. al. [48]</td><td>Dihedral angle of adjacent triangles</td><td colspan="2">Convexity validation based on the dihedral angles</td></tr>
<tr><td>Leonardis et. al. [24]</td><td>Superquadrics</td><td colspan="2">Average error-of-fit between surfaces points and Superquadric</td></tr>
<tr><td rowspan="4">Watershed based</td><td>Mangan and Whitaker [28]</td><td>Deviation from Flatness</td><td rowspan="4">Watershed Function f</td><td>Points belong to the catchment basins that the function f create</td></tr>
<tr><td>Pulla et. al. [34]</td><td>Gaussian curvature, Mean curvature, Root mean square curvature, Absolute curvature</td><td>Points belong to the catchment basins that the function f create</td></tr>
<tr><td>Page et. al.[30, 31]</td><td>Minimum curvature, Normal Curvature</td><td>Points belong to a catchmnet basin if their minimum curvature are above a threshold, additionally points are added to the segments based on their normal curvature</td></tr>
<tr><td>Zuckerberger et. al. [48]</td><td>Dihedral Angles</td><td>Edges belong to the catchment basins that the function f create</td></tr>
<tr><td>Reeb Graphs</td><td>Antini et. al. [2]</td><td>Mean Curvature, Protrusion Function</td><td colspan="2">Discrete Reeb Graph Connectivity, Average Mean Curvature</td></tr>
<tr><td>Model based</td><td>Wu and Levine [45]</td><td>Electrical charge density distribution</td><td colspan="2">Boundary points are selected based on the minima of the electrical charge density distribution</td></tr>
<tr><td>Skeleton based</td><td>Li et. al. [25]</td><td>Geometric and parametric function</td><td colspan="2">Critical points of the geometric and parametric function define segmentation boundaries</td></tr>
<tr><td rowspan="3">Clustering</td><td>Garland et. al. [10]</td><td>Plane, surface normals, irregularity measure</td><td colspan="2">Planarity, Orientation Bias and Compact Shape Bias</td></tr>
<tr><td>Inoue et. al. [16]</td><td>Area and perimeter of regions, normals of surface</td><td colspan="2">Area size, Boundary Smoothness, Region Flatness</td></tr>
<tr><td>Attene et. al. [3]</td><td>Plane, sphere and Cylinder primitives</td><td colspan="2">Points are clustered based on their best fit to the plane, sphere and cylinder primitive</td></tr>
</table>

| | | | |
|---|---|---|---|
| | Gelfand and Guibas [11] | Slippable motions | The number and compatibility of slippable motions define segmentation areas |
| | Shlafman et. al. [41] | Geodesic distance and dihedral angles | Triangles are clustered based on a distance function defined by a weighted sum of geodesic distances and dihedral angles, the weighting is set manually |
| | Katz and Tal [17] | Geodesic distance and dihedral angles | Triangles are clustered based on a distance function defined by a weighted sum of geodesic distances and dihedral angles, the weighting is set manually |
| **Spectral Analysis** | Liu and Zhang [27] | Eigenvalues of the affinity matrix | Eigenvalue clustering of the affinity matrix define the labeling of the triangles into segmentation areas |
| | Zhang and Liu [47] | Eigenvalues of the affinity matrix | The two leading eigenvectors of the affinity matrix define a bipartition of the triangle set |
| **Explicit Boundary Extraction** | Lee et. al. [22, 23] | Minimum Curvature | Centricity of the contour, Salience test |
| **Critical Points based** | Katz et. al. [18] | Critical Points, Dihedral angles | Centrality of the points of the mesh |
| | Lin et. al. [26] | Critical Points, Geodesic Distance, Protrusion Function, | Geodesic and Angular distances from critical points of the mesh define zones where possible segmentation boundaries belong |
| **Multiscale Shape Descriptors** | Mortara et. al. [29] | Integral Gaussian curvature, Contours generated by the intersection of sphere with the mesh | Number of contours generated by the intersection of sphere and surface |
| **Markov Random Fields** | Pichler et al. [32] | Shape Index, Curvedness | Convexity |
| **Direct Segmentation** | Benko and Varady [5] | Normal vectors, various types of filters that distinguishes between the surfaces type | Points of the mesh are tested upon a sequence of surface hypothesis in order to determine to which type of surface they belong |

Tab. 1: Features / criteria in 3D mesh segmentation algorithms.

## 5. ACKNOWLEDGMENTS

This research was supported by the Greek Secretariat of Research and Technology (PENED "3D Graphics search and retrieval" 03 E 520).

## 6. REFERENCES

[1] Andreadakis, S.; Linear Algebra, Symmetria Editions, 1989.
[2] Antini, G.; Berretti S.; Bimbo, A.Del; Pala, P.: 3D Mesh Partitioning for Retrieval by Parts Applications, Proc. IEEE International Conference on Multimedia & Expo (ICME'05), The Netherlands, 2005, 1210- 1213.
[3] Attene, M.; Falcidieno, B.; Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives, The Visual Computer, 22(3), 2006, 181-193.
[4] Attene, M.; Katz S.; Mortara, M.; Patane, G.; Spagnuolo, M.; Tal, A.: Mesh segmentation – A comparative study, International Conference on Shape Modeling and Applications (SMI06), Matsushima (JAPAN), 2006.
[5] Benko, P.; Varady, T.: Direct segmentation of smooth, multiple point regions, Geometric Modeling and Processing, Wako, Saitama (JAPAN), 2002, 169-178.
[6] Besl, P. J.; Jain, R.: Segmentation through Variable-Order Surface Fitting, IEEE PAMI, 10(2), 1988, 167-192.

[7] Brand, M.; Huang, K.: A Unifying Theorem for Spectral Embedding and Clustering, Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, 2003.
[8] Cox, M.; Cox, T.: Multidimensional Scaling, Chapman and Hall, London, 1994.
[9] Fowlkes, C.; Belongie, S.; Chung, F.; Malik, J.: Spectral Grouping Using the Nystrom Method, IEEE Pattern Analysis and Machine Intelligence, 26(2), 2004, 214-225.
[10] Garland, M.; Willmott, A.; Heckbert, P. S.: Hierarchical face clustering on polygonal surfaces, Proc. Symposium on Interactive 3D Graphics, 2001, 49-58.
[11] Gelfand, N.; Guibas, L. J.: Shape Segmentation Using Local Slippage Analysis, Eurographics Symposium on Geometric Processing (SGP), Nice (FRANCE), 2004, 214-223.
[12] Hoffman, D.; Richards, W.: Parts of recognition, Cognition 18, 1984, 65–96.
[13] Hoffman, D.; Singh, M.: Salience of visual parts, Cognition 63, 1997, 29–78.
[14] Hoppe, H.: Progressive Meshes, ACM SIGGRAPH, 1996, 99-108.
[15] Hilaga, M.; Shinagawa, Y.; Komura, T.; Kunii, T. L.: Topology Matching for Full Automatic Similarity Estimation of 3D, Proceedings of SIGGRAPH, Computer Graphics Proceedings, 2001, 203-212.
[16] Inoue, K.; Takayuki, I.; Atsushi Y.; Tomotake, F.; Kenji, S.: Face clustering of a large-scale cad model for surface mesh generation, Computer-Aided Design, 33, 2001, 251-261.
[17] Katz, S.; Tal, A.: Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts, ACM Trans. on Graphics, 22(3), 2003, 954-961.
[18] Katz, S.; Leifman, G.; Tal, A.: Mesh Segmentation using Feature Point and Core Extraction, The Visual Computer, 21(8-10), 2005, 649-658.
[19] Koenderink, J. J.; van Doorn, A. J.: Surface shape and curvature scales, Image and Vision Computing, 10, 1992, 557-565.
[20] Lavoué, G.; Dupont, F.; Baskurt, A.: Curvature tensor Based Triangle Mesh Segmentation with Boundary Rectification, IEEE Computer Graphics International (CGI'2004), Crete, Greece, 2004, 10-17.
[21] Lee, Y.; Lee, S.: Geometric snakes for triangular meshes, Computer Graphics Forum (Eurographics 2002), 21(3), 2002, 229-238.
[22] Lee, Y.; Lee, S.; Shamir, A.: Cohen-Or, D., Seidel, H.P.: Intelligent mesh scissoring using 3D snakes, Pacific Conference on Computer Graphics and Applications, 2004, 279–287.
[23] Lee, Y.; Lee, S.; Shamir, A.; Cohen-Or, D.; Seidel H.-P., Mesh Scissoring with Minima Rule and Part Salience, Computer-Aided Geometric Design, 22(5), 2005, 444-465.
[24] Leonardis, A.; Jaklic, A.; Solina, F.: Superquadrics for Segmenting and Modeling Range Data, IEEE PAMI, 19(11), 1997, 1289-1295.
[25] Li, X.; Toon, T. W.; Huang, Z.: Decomposing polygon meshes for interactive applications, SI3D, 2001, 35-42.
[26] Lin, H. S.: Liao, H. M.: Lin, J.: Visual Salience-Guided Mesh Decomposition, IEEE Int. Workshop on Multimedia Signal Processing, Siena (ITALY), 2004, 331-334.
[27] Liu, R.; Zhang, H.: Segmentation of 3D meshes through spectral clustering, Pacific Conference on Computer Graphics and Applications, 2004, 298–305.
[28] Mangan, A. P.; Whitaker, R. T.: Partitioning 3D surface meshes using watershed segmentation, IEEE Transactions on Visualization and Computer Graphics, 5(4), 1999, 308-321.
[29] Mortara, M.; Patane, G.; Spagnuolo, M.; Falcidieno, B.; Rossignac, J.: Blowing Bubbles for Multi-Scale Analysis and Decomposition of Triangle Meshes, Algorithmica, 38(2), 2003, 227-248.
[30] Page, D. L.; Koschan, A.; Abidi, M.: Perception-based 3D triangle mesh segmentation using fast marching watersheds, In Proc. of Computer Vision and Pattern Recognition, 2003, 27-32.
[31] Page, D.; Abidi, M.; Koschan, A.; Zhang, Y.: Object representation using the minima rule and superquadrics for under vehicle inspection, In Proc. of 1st IEEE Latin American Conf. on Robotics and Automation, 2003, 91-97.
[32] Pichler, A.; Fisher, R. B.; Vincze, M.: Decomposition of range images using Markov random fields, ICIP, 2004, 1205-1208.
[33] Pratikakis, I.: Watershed-driven image segmentation, PhD Thesis, Vrije Universiteit Brussel (VUB), 1998.
[34] Pulla, S.; Razdan, A.; Farin, G.: Improved curvature estimation for watershed segmentation of 3-dimensional meshes, manuscript (available online).
[35] Rossl, C.; Kobbelt, L.; Seidel, H.-P.: Extraction of feature lines on tri-angulated surfaces using morphological operators, In Proc. of the AAAI Symposium on Smart Graphics, 2000, 71-75.
[36] Sapidis, N.; Besl, P.: Direct Construction of Polynomial Surfaces from Dense Range Images through Region Growing, ACM Trans. on Graphics, 14(2), 1995, 171-200.

[37] Scott, G. L.; Longuet-Higgins, H. C.: Feature grouping by relocalisation of eigenvectors of the proxmity matrix, BMVC, 1990, 103-108.

[38] Shamir, A.: A formulation of boundary mesh segmentation, In Proc. of 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 2004, 51-56.

[39] Shamir, A.: Segmentation and Shape Extraction of 3D Boundary Meshes, Eurographics 2006 - State of the Art Reports, 2006, 137-149.

[40] Shi, J.; Malik, J.: Normalized Cuts and Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 2000, 888-905.

[41] Shlafman, S.; Tal, A.; Katz, S.: Metamorphosis of polyhedral surfaces using decomposition, Eurographics, 2002, 219-228.

[42] Vieira, M.; Shimada, K.: Surface mesh segmentation and smooth surface extraction through region growing, Computer-Aided Geometric Design, 22(8), 2005, 771-792.

[43] Vincent, L.; Soille, P.: Watershed in digital spaces, an efficient algorithm based on immersion simulations, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(6), 1991, 583-598.

[44] Weiss, Y.: Segmentation Using Eigenvectors: A Unifying View, Proceedings IEEE International Conference on Computer Vision, 1999, 975–982.

[45] Wu, K.; Levine, M. D.: 3D Part Segmentation Using Simulated Electrical Charge Distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 1223-1235.

[46] Zhang, Y.; Paik, J.; Koschan, A.; Abidi, M. A.: A simple and efficient algorithm for part decomposition of 3D triangulated models based on curvature analysis, Proc. Intl. Conf. on Image Processing, 2002, 273-276.

[47] Zhang, H.; Liu, R.: Mesh Segmentation via Recursive and Visually Salient Spectral Cuts, Vision, Modelling, and Visualization (VMV), Erlangen (GERMANY), 2005, 429-436.

[48] Zuckerberger, E.; Tal, A., Shlafman, S.: Polyhedral surface decomposition with applications, Computers Graphics, 26(5), 2002, 733-743.