Evolutionary 3D-Shape Segmentation Using Satellite Seeds

Kai Engel and Heinrich Müller

Technische Universität Dortmund, Informatik VII (Graphische Systeme),
Otto-Hahn-Straße 16, 44221 Dortmund, Germany
{kai.engel,heinrich.mueller}@tu-dortmund.de

Abstract. The aim of 3D-shape segmentation is to divide the surface of an object into meaningful parts. We present a novel version of seed-point-based segmentation including an evolutionary optimization to obtain better segments. At first, some initial seeds are defined. Each of them generates several so-called satellite seeds which enable a more detailed control of the segment boundaries. The locations and weights of the seeds are optimized with an Evolution Strategy. The objective function takes the object's curvature at the segments' boundaries into account as well as the length of these boundaries. An extensive evaluation and comparison with important existing segmentation approaches demonstrates the great potential of our approach.

Keywords: Mesh Segmentation, Satellite Seeding, Evolution Strategy.

1 Introduction

3D-shape segmentation has become an important research topic in the field of three-dimensional computer graphics. It is used in several domains like e.g. 3D modeling, texture mapping, and collision detection [1]. The aim of 3D-shape segmentation is to create a decomposition of a 3D-model like the bunny in Fig. 1 into disjoint segments according to some criteria. We focus on models where the surface is approximated by a triangle mesh as shown in Fig. 2. In this context, shape segmentation is also known as *mesh segmentation*.

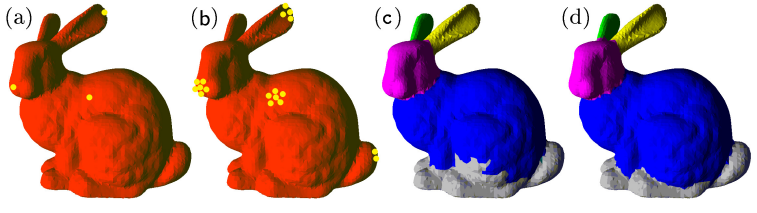

Fig. 1. Overview of our approach: normal seeds, satellite seeds, the resulting segmentation, and an optimized segmentation. The seeds are indicated by yellow circles.

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 438–447, 2012.

This paper is concerned with part-type segmentation [10], i.e. the desired segments correspond to meaningful parts of the original mesh. A horse model, for example, should be divided into head, body, legs, and so on. For the purpose of part-type segmentation, human perception has to be transferred into criteria which can be treated algorithmically. Important and often used criteria are that patch boundaries are typically located in concave surface regions and that the patch boundaries generally have locally minimal length. These are also the major criteria which will be considered within the optimization process introduced in this paper.

Several mesh segmentation approaches use seed points. The basic idea is to determine a region around each of a finite number of properly chosen points on the surface. The regions together induce the decomposition of the surface into patches. The seed points may serve as control points for automatic optimization using an Evolutionary Algorithm. The novel concept of satellite seeds introduced in this paper extends the potential of optimizing the borders between patches without increasing the dimension of the parameter space too much.

This paper is organized as follows. It starts with a short survey on related work in Section 2. In Section 3, an overview of our approach is given. The approach is described in detail in Sections 4 and 5, where the evolutionary optimization is explained in Section 5. In Section 6, we present results including a comparison with the results of eight state-of-the-art techniques. Section 7 concludes the paper and discusses future work.

2 Related Work

Over the last decade, a large number of automatic mesh segmentation approaches has been proposed. Most of them code the objective of segmentation implicitly in an algorithm. However, some other approaches explicitly define an objective function over a set of possible partitions, which may be optimized by an adequate general-purpose solver, cf. e.g. [11]. Extensive surveys on mesh segmentation approaches can be found in [1,10,3].

Most of the solutions up to now are based on fixed heuristics with the aim of an as complete as possible characterization of segmentations. In contrast, Kalogerakis et al. [6] present a data-driven approach which learns an objective function over a feature space from a collection of labeled training meshes, independent from a concrete mesh. It offers a flexible way of learning different types of segmentations for different tasks, without requiring manual parameter tuning. A still existing disadvantage is that always adequate training models are required.

There are several possibilities of deriving a partition into segments from seed points. The potential of satellite seeding will be demonstrated on partitions induced in a way known from weighted Voronoi diagrams on the surface of the mesh. Simari et al. [11,12] have used weighted Voronoi diagrams, too, but in 3D-space using an embedding of the original mesh obtained by multi-dimensional scaling (MDS).

To our knowledge, there is only one mesh segmentation approach containing an evolutionary optimization: Simari and Singh optimize the center initialization, i.e. the positions of weighted partition centers [11]. After the initialization, they use a generalized pattern search for segmentation optimization. In contrast to their approach, we apply an Evolution Strategy for the whole process of optimization. In addition, while Simari and Singh have taken the desired forms of the parts into account by labeling the partition centers, our intention is to obtain good segments without explicit labels.

The contributions are as follows. First, we extend the seed-point-based mesh segmentation approach by the concept of satellite seeds. Second, we introduce an evolutionary optimization to obtain more natural patches. Third, we present an extensive evaluation including an automatic seeding as well as a simulated manual seeding based on ground truth segmentations by Chen et al. [3].

3 Problem Statement and Overview of Our Approach

Given a triangle mesh defining the surface of a geometric object, a decomposition into patches is desired where the boundaries are smooth and optimally located in concave regions. The first aspect is based on the assumption that the boundaries between parts of an object are generally not jagged. The latter one is based on the minima rule [5]. Our solution adopts the seed point approach and includes the following main components:

1. *Seed definition:* The seeds are defined in two steps. The first step chooses a finite set of adequate initial seeds. In the second step these seeds automatically spawn satellite seeds to get a better control of the segment boundaries (see Fig. 1(a) and (b)).
2. *Patch calculation:* Given a finite set of seeds, a decomposition of the object surface into *patches* is calculated (see Fig. 1(c)).
3. *Optimization:* Since patches usually do not match with meaningful parts, we perform a patch optimization using an Evolution Strategy. The optimized patches are often identical with meaningful parts (see Fig. 1(d)). In this article, we denote meaningful parts of an object as *components* and calculated parts, which can be regarded as being meaningful, as *segments*.

Seed definition and patch calculation are described in Section 4, the optimization process is introduced in Section 5.

4 Seed Definition and Patch Calculation

At first, some *initial seeds* are placed on the object's surface in either a manual or an automatic way. Useful heuristics of *manual seeding* are to locate one initial seed in every expected component, and to place it as central as possible within the component. One possible heuristic of automatic seeding also used in this paper is to arrange the first seed far away from the object's centroid and all

following seeds one after another in such a way that their geodesic distance from each other is as large as possible. In contrast to other approaches, the seeds are presented by mesh triangles, which we call *seed triangles*.

Every initial seed defines a *patch*, which, roughly speaking, consists of all mesh triangles that are closer to it than to all other initial seeds according to an adequate distance function. The distance is calculated on the dual graph of the triangle mesh. As shown in Fig. 2, each mesh triangle corresponds to a node in the dual graph. Nodes belonging to neighbor triangles are connected with an edge.

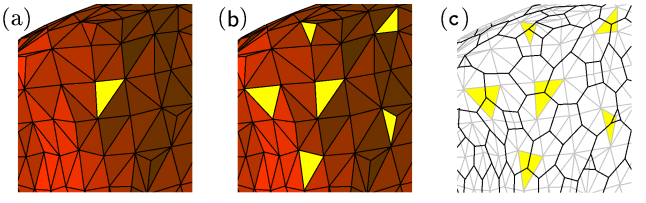

Fig. 2. Triangle mesh with a seed triangle (a), satellite seeds (b), and the dual graph (c)

The segments' boundaries should usually be surrounded by concave object areas. This condition is taken into account by using a *feature based distance function* which is defined by

$$d^{\zeta,\eta,\nu}_{feature}(t_1,t_2) := d_{geo}(t_1,t_2) + \zeta \cdot d_{ang}(t_1,t_2) + \nu \cdot \eta \cdot d_{shape}(t_1,t_2), \qquad (1)$$

where t_1 and t_2 are *adjacent triangles*. It combines the geodesic distance, an angular distance and a special distance which takes into account the object's shape in a local region. The influence of the different partial distances is controlled by $\zeta, \eta \in \mathbb{R}^+_0$, while $\nu \in \{0,1\}$ decides whether or not there is a need to use d_{shape} at all. d_{geo} denotes the usual *geodesic distance* between the centers of t_1 and t_2. The distance between two neighbor nodes of the dual graph is defined as the feature based distance of the corresponding mesh triangles.

By assigning weights to the seeds, the influence of a seed on the boundary of its patch becomes adaptable. Having γ_{seed} as the weight of the seed triangle t_{seed}, the *weighted-feature-based distance* of a triangle t' to t_{seed} is defined by

$$d_{wfeature}(t', t_{seed}) := \frac{1}{\gamma_{seed}} \cdot d_{dual}(t', t_{seed}), \qquad (2)$$

where d_{dual} is the length of a shortest path in the dual graph between the nodes corresponding to t' and t_{seed}. For a given seed, the weighted-feature-based distance between adjacent triangles can be estimated canonically on the dual graph by dividing the feature based distance of the dual graph edge by the seed weight. The assignment of mesh triangles to a patch is realized using a modified form of Dijkstra's shortest path algorithm on the dual graph: the seeds are processed in increasing weight order. If a dual graph node is reached which has already a lower weighted-feature-based distance to another seed, this node will not be

taken into account anymore in the current run of Dijkstra's algorithm. This modification of Dijkstra's algorithm is necessary, because without the modification non-connected patches may occur when different seed weights are chosen.

The reason for combining the geodesic, the angular and our special shape-based distance is that geodesic distances are insensitive to curvature and the part boundaries. Angular distances are insensitive to part boundaries over flat regions. If the edge between two mesh triangles is concave, the angular distance is high. Thus, concave edges should be crossed less often inside a patch than edges in convex regions. The shape-based distance can be seen as an additional quality factor, which for example penalizes the transition from a cylindric region to a concave one.

The shapes of the patches can be influenced by moving seeds or changing their weights initially set on 1.0. Such a variation usually has an effect on the whole patch boundary, so that good boundary parts could be changed to the worse while worse parts are optimized. To avoid such a deterioration of good boundaries, we have developed *satellite seeding* as an extension of initial seeding. The motivation is to divide each boundary into several smaller parts. This is automatically done by substituting each patch by several smaller ones, which we call *subpatches*. To obtain subpatches of a single patch, new seeds are arranged around the initial seed like satellites, close to the initial one (see Fig. 2).

5 Optimization

In this section, we present our patch optimization approach using an enhanced *Evolution Strategy* (ES) that can also handle integer parameters and nominal-discrete ones (cf. [4]). The object variable vector consists of the seed positions in form of triangle indices (assuming that each mesh triangle can be addressed by an index), the weights of the initial and satellite seeds, the influences ζ and η, and the flag ν. All individuals of the initial population are directly derived from the given segmentation. In the majority of cases the optimization produces more natural boundaries. After optimization, every patch is regarded as a segment.

5.1 Reproduction and Selection

Recombination: An extensive survey on well-known recombination strategies is given in [2]. For our optimization problem, we have chosen different recombination strategies for the object variables. The recombination of ζ and η is realized by an intermediate recombination operator, while for the binary valued ν a discrete recombination operator is chosen. Since the seed positions are saved as indices, they cannot be recombined with an intermediate operator, so a discrete operator is used. However, a patch must not become decomposed after applying the recombination. Therefore, for each patch all seeds are randomly chosen completely from one of the individuals selected for recombination.

Mutation: All real-value parameters are mutated by adding a normally distributed random number $\chi \in \mathbb{R}$ with mean 0 and standard deviation σ. Negative values are set to a small positive value. χ is recalculated for each parameter. We have used Rechenberg's 1/5-Rule [8] to adapt the standard deviations. The mutation of seed positions has to be carried out in another way. The seeds are moved over the triangle mesh in such a way that small movements occur more often than larger ones, by using additional information stored in the mesh data structure for mutating the seed indices. The mutation operator has to ensure that each triangle contains at most one seed.

Selection: There are different kinds of Evolution Strategies like the $(\mu + \lambda)$-ES and the (μ, λ)-ES [2]. They differ from each other in the selection mechanism. We have chosen a (μ, κ, λ)-ES [9] which contains the $(\mu + \lambda)$-ES as well as the (μ, λ)-ES as special cases. The (μ, κ, λ)-ES can consider all individuals with an age smaller than the life-span κ.

5.2 Fitness Evaluation

We formulate the optimization as a minimization problem, where the fitness value corresponds to the segmentation quality. According to the knowledge about human perception presented in [5], we want to obtain segments which are mostly surrounded by concave areas. On the other hand, the component boundaries perceived by humans are usually not "jagged". Thus, a segmentation Γ with short segment boundaries is desired. In our *fitness function*

$$f(\Gamma) := \begin{cases} (1-\Lambda) \cdot f_{concave}(\Gamma) + \Lambda \cdot f_{length}(\Gamma); \Lambda \in [0,1] & \text{if } \Gamma \text{ is valid,} \\ \infty & \text{otherwise,} \end{cases} \tag{3}$$

which is to be minimized, the first assumption is taken into account by $f_{concave}$, the latter one by f_{length}. A segmentation is called *valid* iff each patch is connected, i.e. if no patch is decomposed into several parts. Segment boundaries are polylines, all of which have edges of the triangle mesh as line segments. The function $f_{concave}$ is defined in such a way that long line segments have more influence than shorter ones:

$$f_{concave}(\Gamma) := \frac{1}{\sum_{i=1}^{|E|} \frac{1}{l(e_i)}} \cdot \sum\nolimits_{j=1}^{|E|} \left(\frac{1}{1+\max(\alpha_j, 0)} \cdot \frac{1}{l(e_j)}\right). \tag{4}$$

E is a list of all edges belonging to the segment boundaries of Γ. The ith boundary edge is denoted by e_i and its length by $l(e_i)$. α_i denotes the signed angle between the normal vectors of the two mesh triangles adjacent to e_i. This angle is positive, if the object is concave at e_i. The first factor in equation (4) is used for scaling; it ensures that $f_{concave}(\Gamma) \in [0,1]$. The more concave the segment boundaries of Γ are, the lower is the function value of $f_{concave}$.

The function

$$f_{length}(\Gamma) := \frac{1}{\sum_{i=1}^{|E_{mesh}|} l(e'_i)} \cdot \sum\nolimits_{j=1}^{|E|} l(e_j) \tag{5}$$

evaluates how "jagged" the segment boundaries are, which is related with the length of the boundary. The smoother they are, the smaller is the function value. Once again, the list of all boundary edges is denoted by E; the list of all edges belonging to the triangle mesh is denoted by E_{mesh}. While e_j is the jth element of E, e'_i denotes the ith edge of E_{mesh}. Since composition of $f_{concave}$ and f_{length} in (3) is realized as a convex combination, the range of f is a subset of $[0, 1]$.

6 Experiments and Evaluation

6.1 Behavior of the Optimization

A drawback of state-of-the-art techniques based on seeds is that a suboptimal seeding can cause wrong segments. The optimization of seed positions and weights included in our approach compensates for this drawback. In Fig. 3, an example for the positive effects of the evolutionary optimization can be seen. The initial segmentation on the left side is taken as the source for two optimizations: one using a (μ, κ, λ)-ES and one using a $(\mu + \lambda)$-ES. Both have been applied for 150 generations. Figure 3 also shows the corresponding curves of the best fitness values per generation. In both cases, the resulting segmentations are much better. In contrast, before starting the optimization, the green and the yellow segment are not quite good; the yellow one describing the middle finger was running down the other side of the hand up to the little finger. Please also note that even optimal segmentations yield fitness values considerably larger than 0.

Other segmentations calculated by our prototype are shown in Fig. 4. The results essentially correspond to human expectations. In particular, technical models like the bearing object are almost perfectly segmented. But also the segmentations of natural models are quite good in most cases. If still necessary at all, jagged boundaries may be fixed by postprocessing [7].

6.2 Evaluation

Since we have demonstrated some results of our approach so far, we now measure the quality of our segmentation method by taking segmentations manually created by human test persons into account, that can be seen as being "optimal". Such segmentations are known as *ground truth segmentations*. We have used the *Rand Index* (RI) to evaluate the discrepancy between a calculated segmentation and a ground truth segmentation [3]. It is the relative number of all pairs of mesh triangles which either belong to the same segment in both segmentations or which belong to two different segments in both segmentations. With this information, the similarity of the segmentations is calculated. Originally, the Rand Index RI_{orig} is defined to be 1 if both segmentations are identical, and it's range is $[0, 1]$. According to [3], we use $RI = 1 - RI_{orig}$ as Rand Index in order to compare our results to those of established approaches. Thus, this Rand Index grows with increasing discrepancy. A more detailed description can be found in [3]. Since the Rand Index can also be evaluated for segmentations of other approaches, an objective comparison of our results with other ones is feasible.

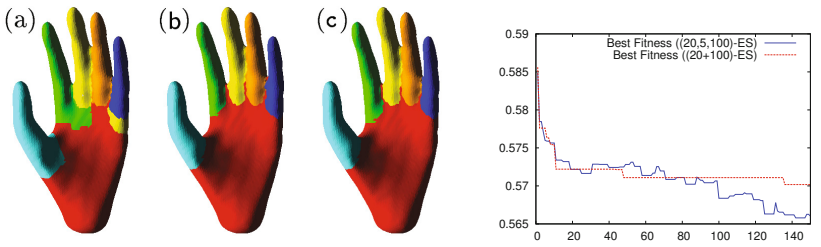

Fig. 3. The initial segmentation (a) was optimized by a (μ, κ, λ)-ES (b) and a $(\mu + \lambda)$-ES (c). The best individuals' fitness values are shown for 150 generations.

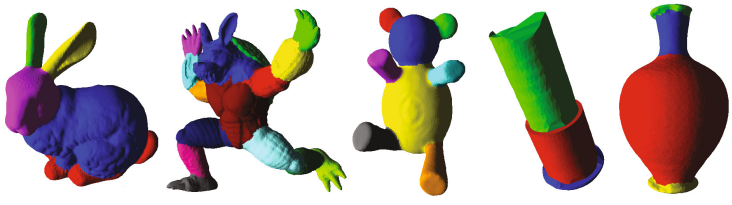

Fig. 4. Segmentations achieved by our prototype

An extensive benchmark containing 380 models as well as eleven different ground truth segmentations in the average for each model was published by Chen et al. [3]. The models are divided into 19 categories. We have used this benchmark to evaluate our segmentation approach.

In order to perform a large number of experiments, we have chosen two variants of defining a reasonable number of seeds individually for every model in an automatic way. *Variant 1* is given by our automatic seeding, where the number of initial seeds is determined by taking the average segment number of all ground truth data belonging to the current model. *Variant 2* is based on the ground truth data. The initial seeds are placed near the centroids of randomly chosen ground truth segments. This can be seen as a simulation of a manual seeding, which enables to evaluate the possible advantage of an expected optimal manual seeding against an automatic seeding. In this sense, variant 2 can also be considered as a lower bound (with respect to the Rand Index) for all automatic approaches.

We have calculated one segmentation per seeding variant for each of the 380 benchmark models mentioned above with $\mu = 3$, $\lambda = 15$, $\kappa = 5$ and an experimentally determined $\Lambda = 0.3$. The optimization was stopped already after 30 generations, which turned out to be sufficient. On the left side of Fig. 5, the influence of the optimization is shown on the basis of the Rand Index. The Rand Index values averaged over the 380 models are shown for the situations before and after optimization. *Sat1* and *Sat2* stand for our segmentation approach using seeding variant 1 and 2, respectively. Please remember that the fitness function is defined completely independent from the Rand Index and that even

the best possible RI value will be positive. The latter aspect is reasoned by different subjective perceptions of a model and its components, which results in different ground truth data. For example, the wings of a high-wing plane can be interpreted as one single component or as two independent components. According to this, in [3], a RI of 0.1 was observed for manual segmentations that are regarded as being optimal. For Sat2, the optimization yields a similar value, which confirms the capabilities associated with satellite seeding.

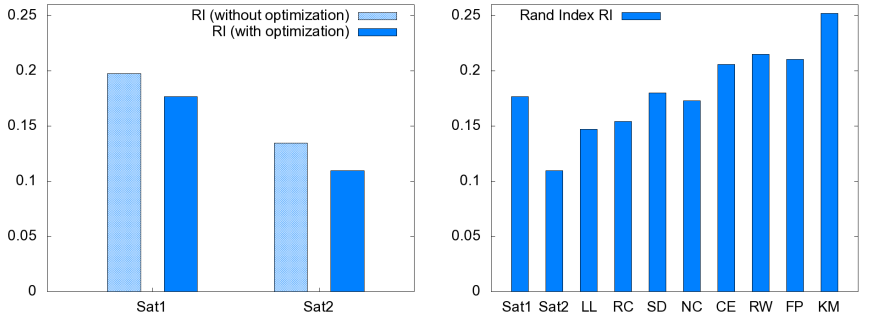

Fig. 5. Effect of the optimization (left) and averaged RI values (right)

In the right diagram of Fig. 5, the averaged RI values of our optimized segmentations and of established techniques are shown (cf. [3,6]). The established techniques are based on Labeling and Learning (LL) [6] regarding three training meshes for every evaluated segmentation, Randomized Cuts (RC), Shape Diameter Function (SD), Normalized Cuts (NC), Core Extraction (CE), Random Walks (RW), Fitting Primitives (FP), and K-Means Clustering (KM). A survey, except for the LL approach, is given in [3]. Variant 1 is superior to 5 of the 8 established techniques. A great theoretical potential of the proposed approach is demonstrated by the RI values of variant 2 which are significantly better than those of all other techniques. A trouble of our automatic seeding is that seeds may be placed nearby boundaries of meaningful components. This may be unfavorable even for satellite seeding in its current version. A further question of future research is whether the existing techniques also have a potential of improvement which can be quantitatively estimated analogously to variant 2.

We have also investigated the influence of satellite seeding on the optimization by taking only the initial seeds of Sat2 without satellite seeds. In this case, the RI after optimization is 0.146, which is clearly worse than the RI in the case of satellite seeds. Furthermore, it is also worse than in the case of Sat2 without optimization shown in Fig. 5. This behavior is caused by the fact, that the optimization algorithm tends to move "lonely" initial seeds onto parts bounded by concave areas, which often results in patches that are too small for the desired segmentation granularity. Therefore, satellite seeding as an extension of normal seeding is obviously effective for evolutionary optimization.

7 Conclusion and Future Work

We have presented a novel approach to mesh segmentation suitable for generating optimized segmentations using an Evolution Strategy. In many cases, it yields better results than important well-known techniques. Furthermore, our approach is especially well suited for semi-automatic segmentation, i.e. a manual seeding which can be done with minimal effort followed by automatic patch calculation and optimization. A major challenge of future work is to improve the automatic seeding to reduce the gap between the segmentation qualities of Sat1 and Sat2 shown in Fig. 5.

Further, the reliability of Sat2 as a simulation of a manual seeding could be investigated. First tests confirm that in most cases segmentations calculated from seedings by humans are very similar to the ones presented in this paper.

Finally, alternative fitness functions could be studied. For example, the fitness function might also force corresponding satellite seeds to stay close to each other. This could reduce the occurrence of invalid individuals in the parent population.

Acknowledgements. The bunny model is from the Stanford 3D Scanning Repository [13], all others are from the mentioned benchmark and available at [14].

References

1. Attene, M., Katz, S., Mortara, M., Patane, G., Spagnuolo, M., Tal, A.: Mesh Segmentation - A Comparative Study. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications, SMI 2006 (2006)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford (1996)
3. Chen, X., Golovinskiy, A., Funkhouser, T.: A Benchmark for 3D Mesh Segmentation. ACM Transactions on Graphics (Proc. SIGGRAPH) 28(3) (August 2009)
4. Emmerich, M., Grotzner, M., Schütz, M., Groß, B.: Mixed-Integer Evolution Strategy for Chemical Plant Optimization with Simulators. In: Parmee, I. (ed.) Evolutionary Design and Manufacture, ACDM (2000)
5. Hoffman, D.D., Richards, W.A.: Parts Of Recognition. Cognition 18, 65–96 (1984)
6. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3D Mesh Segmentation and Labeling. ACM Transactions on Graphics 29(3) (2010)
7. Kaplansky, L., Tal, A.: Mesh Segmentation Refinement. Comput. Graph. Forum 28(7), 1995–2003 (2009)
8. Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart (1973)
9. Schwefel, H.-P.: Evolution and Optimum Seeking. Sixth-Generation Computer Technology. Wiley Interscience, New York (1995)
10. Shamir, A.: A survey on Mesh Segmentation Techniques. Comput. Graph. Forum 27(6), 1539–1556 (2008)
11. Simari, P., Singh, K.: Multi-objective shape segmentation. Tech. rep., Departement of Computer Science, University of Toronto (2008)
12. Simari, P.D., Nowrouzezahrai, D., Kalogerakis, E., Singh, K.: Multi-objective shape segmentation and labeling. Comput. Graph. Forum 28(5), 1415–1425 (2009)
13. Stanford 3D Scanning Rep., `http://www.graphics.stanford.edu/data/3Dscanrep/`
14. Princeton Mesh Segmentation Benchmark, `http://segeval.cs.princeton.edu`