A segmentation transfer method for articulated models

E. Elghoul and A. Verroust-Blondet

Inria Paris-Rocquencourt, France

Abstract

In this paper, we propose using a pre-segmented example model to perform semantic-oriented segmentation of non-rigid 3D models of the same class (human, octopus, quadrupeds, etc.). Using the fact that the same type of non-rigid models share the same global topological structure, we exploit coarse topological shape attributes in conjunction with a seed-based segmentation approach to transfer a meaningful and consistent segmentation from the example to the target models. Promising results of inter-shape segmentation transfer are shown and discussed for different classes of models.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—I.3.6 Methodology and Techniques

1. Introduction

Mesh segmentation consists in partitioning the surface into a set of patches that are uniform with respect to a given property (see [AKM*06, Sha08, CGF09, BVLD09] for comparative studies and surveys on segmentation techniques). Attene et al. [AKM*06] distinguish between the geometric and semantic-oriented approaches according to the properties held in the segmentation process. We are interested in retaining the semantic information during the segmentation. A particularly challenging task is then the automatic identification of semantically meaningful parts of a 3D model, which can be hard to achieve when only the shape geometry is considered. We propose letting the user provide the desired semantic information by segmenting one model into meaningful parts and then using this model as an example and reproduce a similar segmentation on each model of the same class. This corresponds to what Golovinskiy and Funkhouser [GF09] call "segmentation transfer".

Partitioning the surfaces of a family of models in a similar way [SSCO08, SNKS09, GF09, KHS10, SvKK*11] is slightly different, as the main goal in this case is only to obtain consistent mesh partitions. Moreover, in [SSCO08, KHS10, SvKK*11] the meshes are divided into parts that may be non connected (group the four legs of quadrupeds in one region of the surface, for example), which is not what we expect. Nevertheless, the co-segmentation method of [GF09] is extended to a technique that transfers a segmentation from an example to a set of models. They obtain good results with rigid models and less convincing segmentations for articulated models such as a giraffe and a horse.

In fact, if we focus our attention on articulated models, we can notice that the models belonging to the same class, for example quadrupeds, have a similar structure, which can be decomposed into a head, a torso, four legs and, sometimes, a tail. Thus, to perform a rough localization of the segments to be transferred from the example model to the target models, our main idea is to extract a graph representing this structure for each model of the same class. Reeb graphs seem to be suitable for this goal. These graphs have been used to segment 3D meshes into meaningful parts [TVD07, BDBP09] or to perform partial retrieval [TVD09]. As the expected segmentation depends on the exemplar one, we cannot ensure that the previous methods will succeed in transferring a segmentation from one model to another. However, the Reeb graph is useful to establish a correspondence between two meshes and a rough localization of the segments. A flexible segmentation process is then necessary to reproduce the desired segmentation on the target model. For this purpose, we use the random walk algorithm [LHMR09]. It is a fast seed-based segmentation method that outperformed the K-means approach [STK02] in [CGF09]. This choice enables us to transform the problem of segmentation transfer into a problem of selecting seed facets on the target model.

The following paragraphs provide a definition of the Reeb

DOI: 10.2312/conf/EG2013/short/017-020

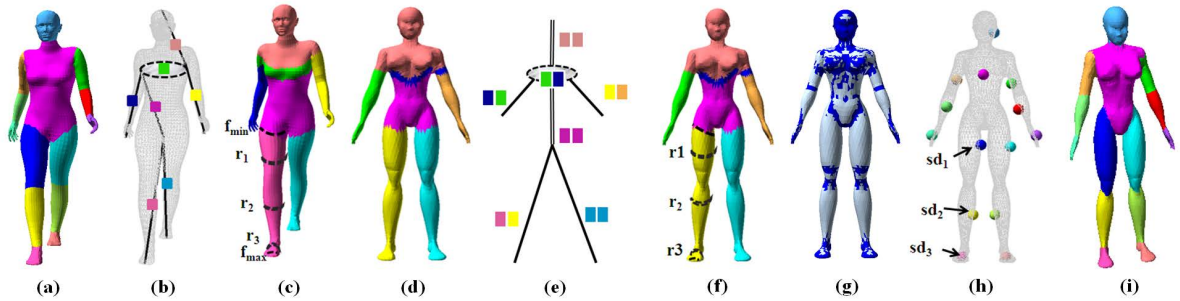

Figure 1: *Segmentation transfer process: (a) Segments of M_E, (b) $\mathcal{G}(M_E)$ with its associated labels, (c) $\mathcal{P}(M_E,f)$ and relative distances r_i of some segment centroids, (d) $\mathcal{P}(M_T,f)$ of the unsegmented target model M_T, (e) Illustration of $\mathcal{C}(M_E,M_T)$ using paired labels from $\mathcal{P}(M_E,f)$ and $\mathcal{P}(M_T,f)$, (f) detection of seed positions using (e), (g) convexity measure : unsuitable locations for seeds in dark blue, (h) seed positioning, (i) consistent segmentation of M_T using random walks.*

graph and a brief description of the random walk method that will be used in our approach, which is detailed in section 2.

Reeb graphs. Let $f : M \to \mathbb{R}$ be a function defined over a 2-manifold triangular mesh M. Level sets of f are the sets $f^{-1}(u) = \{x \in M, f(x) = u\}$. Each of these sets, if it exists, can be connected or not. Topological changes in the level sets of f can occur on critical points while following the evolution of f over M. The Reeb graph of f w.r.t. M, $R(M,f)$ is a graph whose nodes correspond to these critical points and which encodes the connectivity between them. It is the quotient space defined by the equivalence relation that identifies the points belonging to the same connected component of level sets of f (see [BGSF08] for a more formal definition and a detailed study of Reeb graphs).

Random walk segmentation method. The random walk mesh segmentation method (see [LHMR09] for more details) works on 2-manifold triangulated models M. It builds a segmentation of M in N segments from N facets of M given as seeds. The segmentation is computed by assigning a value k, $k = 1,...,N$ to each facet F of M, where k is the index of the seed facet which has the highest probability of being reached first by a random walk from F on the dual graph of M. Probabilities introduced by Lai et al. ensure that each segment is a contiguous region of the surface and lead to the creation of segment boundaries on edges of high negative curvature.

2. Our method

In what follows, M_E will denote the segmented model given as example, with $S_1..S_N$ its N segments, C its class of models and M_T any model of C. All the models are articulated triangulated 2-manifold meshes and are such that all the models belonging to one class are consistently aligned and are in an upright position, e.g. head on top and legs at the bottom for human and animal models. The algorithm is decomposed as follows: (i) a coarse graph $\mathcal{G}(M_E)$ is computed and a localization of the segments S_i of M_E w.r.t. $\mathcal{G}(M_E)$ is performed in a preprocessing step (cf. section 2.1), (ii) then, for any mesh M_T of C, a coarse graph $\mathcal{G}(M_T)$ is computed, (iii) a common correspondence graph $\mathcal{C}(M_E,M_T)$ is built from $\mathcal{G}(M_E)$ and $\mathcal{G}(M_T)$ (cf. section 2.2), (iv) the segmentation of M_T is performed (cf. section 2.3).

2.1. Preprocessing of M_E

This step is executed once for the segmented model M_E. The construction of the coarse graph $\mathcal{G}(M)$ is the same for all the meshes M of C.

Computation of $\mathcal{G}(M)$. A Reeb Graph $R(M,f)$ is built by taking as a function f upon M a distance on the mesh to a source point located on the top of M. f is computed with Dijkstra's algorithm on the mesh graph during the sweeping process which builds $R(M,f)$. As noticed by [BDBP09], the Reeb graph partitions the 3D mesh into connected parts corresponding to contiguous connected components of level sets between two critical points that define an edge of $R(M,f)$. During the sweeping process, each facet of M is labeled by its corresponding edge of $R(M,f)$.

[ZMT05, AHLD07] note that the Reeb graph $R(M,f)$ may contain multiple neighboring local extrema or saddle points in almost flat regions. Moreover, some internal edges, parallel to the level sets, may corrupt the topological symmetry of $R(M,f)$. Thus we use the graph filtering mechanism of Aujay et al [AHLD07] that eliminates topologically insignificant edges to obtain a coarse graph $\mathcal{G}(M)$. A weight equal to the normalized difference between the distance to the source vertex of two adjacent nodes of $R(M,f)$ is associated with the corresponding edge of $R(M,f)$. Edges of $R(M,f)$ are first ordered by increasing weight and two thresholds τ_1 and τ_2 are introduced. External edges with a weight lower than τ_1 and their nodes are removed and the labels of the facets of M associated with these edges are updated. Internal edges of a weight less than τ_2 are eliminated and their extremities are merged together into a node marked as "fusion". The labels of the facets of M associated with these edges are then associated with the fusion node. This leads to a low-level partitioning $\mathcal{P}(M,f)$ of the surface (Fig. 1 (c)), with one label

associated with each surface part. We then obtain a graph $\mathcal{G}(M)$ where both geometrical and morphological symmetries are recovered (Fig. 1 (e)).

Localization of the segments w.r.t. $\mathcal{G}(M_E)$. Each segment S_i of the segmentation associated with M_E is a connected sub-part of the mesh. As we consider that the segmentation corresponds to a decomposition of M_E in meaningful parts, it is strongly related to $\mathcal{G}(M_E)$ and its associated low level partitioning of the surface $\mathcal{P}(M_E, f)$: S_i is mainly composed of facets having the same label l_i and belonging to a surface part M' of $\mathcal{P}(M_E, f)$. This label l_i corresponds to an edge or a fusion node of $\mathcal{G}(M_E)$. Let f_i be the average of the distance values $f(x)$ of the vertices x belonging to a neighborhood of the centroid of all faces of S_i and $f_{min}(M')$ and $f_{max}(M')$ the extremal values of f on M'. We then associate the couple of values (l_i, r_i) to localize S_i w.r.t. $\mathcal{G}(M_E)$, r_i being equal to the relative position of f_i:

$$r_i = \frac{f_i - f_{min}(M')}{f_{max}(M') - f_{min}(M')}$$

2.2. Construction of the common graph $\mathcal{C}(M_E, M_T)$

In fact, a one-to-one mapping of parts of articulated models may be difficult to achieve as the number of parts may vary irregularly. However, due to the intra-class similarity of anatomical features, potential matches can be established between most of the edges of the coarse graphs $\mathcal{G}(M_E)$ and $\mathcal{G}(M_T)$, which is required for the computation of a common graph $\mathcal{C}(M_E, M_T)$. The process begins by detecting the minimum common symmetry axis between $\mathcal{G}(M_E)$ and $\mathcal{G}(M_T)$ and pairing together their respective edges as shown in Figure 1 (e). It uses symmetry axes computed while building $\mathcal{G}(M_E)$ and $\mathcal{G}(M_T)$ as in [AHLD07]. Then $\mathcal{C}(M_E, M_T)$ is built hierarchically by matching first the remaining edges of $\mathcal{G}(M_E)$ adjacent to the common symmetry axis with the corresponding edges of $\mathcal{G}(M_T)$. This process takes into account the angular ordering of the edges around the symmetry axis at junction levels. The resulting graph $\mathcal{C}(M_E, M_T)$ establishes a correspondence between the edges of $\mathcal{G}(M_E)$ and those of $\mathcal{G}(M_T)$.

For some models, extra edges from either $\mathcal{G}(M_E)$ or $\mathcal{G}(M_T)$ may not be matched with an element of $\mathcal{C}(M_E, M_T)$ because the corresponding parts are not present on the other model. This can be the case of the tail for quadruped models. As these edges may be needed later in the segmentation process, they are stored in two extra queues, $\mathcal{Q}_{ext}(M_E, M_T)$ and $\mathcal{Q}_{ext}(M_T, M_E)$ respectively.

2.3. Segmentation of M_T

With the correspondence graph $\mathcal{C}(M_E, M_T)$ and the localization of each S_i w.r.t. $\mathcal{G}(M_E)$, we will ensure that the adjacency graph of the segments of M_E and of the segments of M_T will be similar. As mentioned in the introduction, we use the random walk technique [LHMR09] to perform the segmentation of M_T. Thus, in order to transfer the segmentation of M_E on M_T, for each segment S_i of M_E we need to select a candidate seed facet s_i on M_T whose location is deduced from the values (l_i, r_i) computed in section 2.1. Note that the identifiers i indicate the correspondence between the segment patches of M_E and those of M_T.

If the label l_i of segment S_i belongs to the extra queue $\mathcal{Q}_{ext}(M_E, M_T)$ computed in section 2.2, S_i does not correspond to an element of $\mathcal{C}(M_E, M_T)$ and it is ignored. In fact, a segment S_i has a corresponding segment on M_T only when l_i corresponds to an element of $\mathcal{C}(M_E, M_T)$. As the random walk approach creates segment boundaries around edges of high negative curvature, we don't want to select seed facets in concave regions of the surface. Such zones are not suitable for a normal propagation of the random walk algorithm. The choice of the seed facet s_i is then performed in two steps. First, a rough localization of the region of M_T where s_i will be chosen is made and a set of candidate facets is computed. These facets belong to the region of $\mathcal{P}(M_T, f)$ whose label is associated with l_i via $\mathcal{C}(M_E, M_T)$ and are located on a set of contiguous level sets whose relative values are given by r_i. Second, a convexity measure equal to the average of the convexity values of the three edges composing the facet is associated with each facet of M_T. This value is given by the expression of the probability distribution introduced in the random walk algorithm [LHMR09]. The average convexity value C_T of all the facets of M_T is computed and used as a threshold. The candidate facets that have a convexity value lower than C_T are rejected. Then s_i is chosen among the remaining set of facets: it is the facet whose relative distance value is the nearest to r_i.

We also identify seed facets of the additional parts of M_T whose labels are stored in $\mathcal{Q}_{ext}(M_T, M_E)$, if they exist. These parts correspond to edges of $\mathcal{G}(M_T)$ that did not match with an element of $\mathcal{C}(M_E, M_T)$. For each of these parts, an extra segment is created: the centroid of its facets is computed and the closest facet to the part centroid is detected and stored as a seed with an identifier i greater than N.

Once all the identified seeds are computed, the random walk algorithm is run on M_T. Results of the described method of segmentation transfer are shown in Figure 2.

3. Results and Discussion

We used articulated models from the Princeton Segmentation Benchmark [CGF09] for all our tests. Like [GF09], we used a PCA algorithm followed by an ICP algorihm to align pairs of models. For each class, the values of the graph filtering thresholds τ_1 and τ_2 were deduced experimentally to obtain comparable graphs $\mathcal{G}(M)$ within the same class. Then our examples were manually segmented by a user.

Let us now comment on the results presented in Figure 2. Even if the alignment of the models is not perfect (a2), our method succeeded in consistently segmenting the giraffe and the bull w.r.t. the segmented horse. Moreover, our method

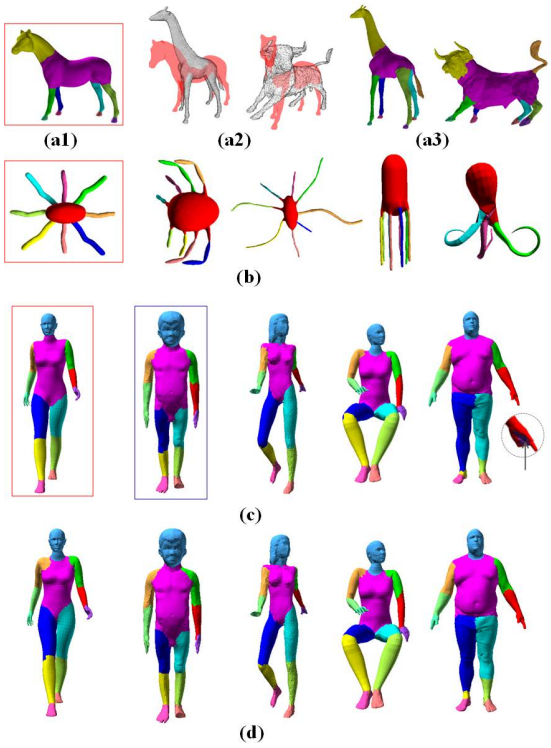

Figure 2: *Segmentation transfer results: (a1) M_E belongs to the quadruped class, (a2) the poor alignment of the subparts of M_E and of the target models does not degrade the quality of the resulting segmentation shown in (a3), (b) and (c) M_E is framed in red, (d) stability of the method on the human class: segmentation results taking the boy framed in blue in (c) as M_E.*

allows for extra segments, e.g. tails in (a3), and eliminates inconsistent segments when necessary, e.g. the rightmost octopus in (b). To test the stability of our approach, we used the segmented boy (which is the result of the segmentation transfer of the framed woman in (c)) to segment the human models in (d). We obtained similar results. This is accounted for by the robustness of the random walk algorithm w.r.t. seed shifts and its flexibility to reproduce different user-given segmentations. We noticed that the segment boundaries of most of the models follow the natural part contours which would have been difficult if they had been deduced from *constrictions* of a more sophisticated Reeb graph [TVD07].

The rightmost model in (c) shows the limitations of our algorithm. They are due to unsuitable seed positions. Although mesh facets were filtered for seed selection by a convexity threshold, some of them can be positioned inside a large concave region, like the seed in the palm of the hand. Moreover, the seeds corresponding to the thighs were positioned on the knees because the branching node corresponding to the crotch in $\mathcal{G}(M_T)$ is very low, which shortened the man's legs. To resolve this issue, we plan to extend the random walk algorithm using several seeds per segment when necessary. Further research includes the use of simplified target meshes to improve the graph computation and, consequently, the seed localization.

References

[AHLD07] AUJAY G., HÉTROY F., LAZARUS F., DEPRAZ C.: Harmonic skeleton for realistic character animation. In *Symposium on Computer Animation* (2007), pp. 151–160. 2, 3

[AKM*06] ATTENE M., KATZ S., MORTARA M., PATANÈ G., SPAGNUOLO M., TAL A.: Mesh segmentation - a comparative study. In *SMI* (2006), p. 7. 1

[BDBP09] BERRETTI S., DEL BIMBO A., PALA P.: 3D mesh decomposition using Reeb graphs. *Image Vision Comput. 27*, 10 (2009), 1540–1554. 1, 2

[BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theor. Comput. Sci. 392*, 1-3 (2008), 5–22. 2

[BVLD09] BENHABILES H., VANDEBORRE J.-P., LAVOUÉ G., DAOUDI M.: A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. In *SMI* (2009), pp. 36–43. 1

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T. A.: A benchmark for 3D mesh segmentation. *ACM Trans. Graph. 28*, 3 (2009). 1, 3

[GF09] GOLOVINSKIY A., FUNKHOUSER T. A.: Consistent segmentation of 3D models. *Computers & Graphics 33*, 3 (2009), 262–269. 1, 3

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics 29*, 3 (2010). 1

[LHMR09] LAI Y.-K., HU S.-M., MARTIN R. R., ROSIN P. L.: Rapid and effective segmentation of 3D models using random walks. *Computer Aided Geometric Design 26*, 6 (2009), 665–679. 1, 2, 3

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Comput. Graph. Forum 27*, 6 (2008), 1539–1556. 1

[SNKS09] SIMARI P. D., NOWROUZEZAHRAI D., KALOGERAKIS E., SINGH K.: Multi-objective shape segmentation and labeling. *Comput. Graph. Forum 28*, 5 (2009), 1415–1425. 1

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259. 1

[STK02] SHLAFMAN S., TAL A., KATZ S.: Metamorphosis of polyhedral surfaces using decomposition. *Comput. Graph. Forum 21*, 3 (2002), 219–228. 1

[SvKK*11] SIDI O., VAN KAICK O., KLEIMAN Y., ZHANG H., COHEN-OR D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph. 30*, 6 (2011), 126. 1

[TVD07] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Topology driven 3D mesh hierarchical segmentation. In *Shape Modeling International* (2007), pp. 215–220. 1, 4

[TVD09] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Partial 3D shape retrieval by reeb pattern unfolding. *Comput. Graph. Forum 28*, 1 (2009), 41–55. 1

[ZMT05] ZHANG E., MISCHAIKOW K., TURK G.: Feature-based surface parameterization and texture mapping. *ACM Trans. Graph. 24*, 1 (Jan. 2005), 1–27. 2