Multi-objective shape segmentation

Technical report CSRG-575, Dynamic Graphics Project, Department of Computer Science, University of Toronto, ©June 2008

Patricio Simari and Karan Singh

Abstract

The users of shape segmentation algorithms possess a wealth of knowledge about the objects they wish to segment. Current automatic segmentation approaches, however, apply a fixed objective uniformly to all parts and limit their input to the number of segments desired and a small set of parameter values. In this paper, we propose the concept of multi-objective shape segmentation. This model allows for the incorporation of domain specific knowledge by means of competing objectives that can selectively refer to one or more segmentation labels or to the segmentation as a whole. The segmentation problem is thus cast as the optimization of an aggregate objective function which is a combination of these heterogeneous competing objectives. We introduce the use of multiplicatively weighted Voronoi partitioning as a means to parameterize segmentations and present algorithms for coarse center placement, segmentation labeling as a function of objectives, and center refinement. We then show how our approach can accommodate symmetry constraints, which ensure desired segmentation properties and effectively reduce the dimensionality of the optimization domain when prior knowledge of the shape is available. Finally, we show how even shapes under complicated articulation can be handled by our approach by using multi-dimensional scaling.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: [I.3.5 Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems. Hierarchy and geometric transformations.

1. Introduction

Digital representations of 3D models, such as meshes and point clouds, are rarely appropriate in their raw format for the range of applications that benefit from their use, including computer graphics, computer animation, engineering design, and medical diagnostics, to name a few. Such applications often require the preprocessing of this raw data into simpler constituting parts. This is commonly referred to as shape segmentation, and its automation remains a challenging area within computer science.

The users of segmentation algorithms, such as biologists, doctors, mechanical engineers or digital character modelers and animators, possess a wealth of knowledge about the objects they are analyzing and working with. However, current algorithms for shape processing do not allow for the incorporation of detailed domain knowledge and usually limit their input to the number of segments desired and a handful of threshold values. If a user wishes to automate the segmentation of models according to his domain-specific knowledge, the current approach is to implement a new segmentation algorithm from scratch. This is often out of the realm of most users. Even when it is not, such a ground-up implementation requires a considerable investment of time and resources.

In the following, we will present a framework for shape segmentation that allows for the incorporation of domain-specific knowledge through heterogenous objectives each of which refers to one or more segmentation labels. These objectives can be unary, asserting properties of an individual part associated with a given label (*e.g.* that it should be narrow, compact, flat, symmetric) or they can by n-ary, referring to part interrelations, (*e.g.* a set of parts should be parallel to each other, or have the same proportions, or two parts should be perpendicular to each other.) We thus cast the segmentation problem as an optimization minimizing an aggregate objective function which combines all objectives as a weighted sum. We summarize our contributions as follows.

Contributions: We introduce the notion of multi-objective shape segmentation and the application of weighted Voronoi space partitioning as an approach to segmentation parameterization. We propose seeding approaches to initialize the Voronoi centers, including a novel general-purpose evolu-

Figure 1: *Influence of multiple objectives. On the* ***left*** *we see the result of optimizing for minimum squared difference of convex hull volume to mesh volume. In the* ***center*** *is the result of optimizing only for the narrowness of the handle. On the* ***right****, we see the result of optimizing a combination of these two objectives and perpendicularity between head and handle. (See Section 10 for details.)*

tionary approach. We present strategies for segmentation labeling to maximize objectives, including an efficient optimal solution for unary objectives. We show how our approach can accommodate symmetry constraints which effectively reduce the dimensionality of the optimization domain when prior knowledge of the shape is available. Finally, we show how even shapes under complicated articulation can be handled by our approach by using multi-dimensional scaling.

2. Related work

Shape segmentation refers to a partitioning of the set of surface elements (such as mesh faces or point cloud points), the partition being determined by a labeling of these faces that optimizes some objective function. Historically, shape segmentation algorithms can be broadly classified according to the nature of this function and optimization strategy into the following broad groups.

Affinity: In affinity-based approaches, it is assumed that for every pair of mesh faces one can assign a confidence to their being in the same segment, or conversely, a dissimilarity metric that determines how likely they are to be in different segments. In such approaches the goal is to maximize intra-segment similarity and inter-segment dissimilarity. Examples include graph cut [KT03, KLT05, LZ07] and watershed approaches [MW99, ZTS02].

Model fitting: In model fitting approaches, it is assumed that every segment of the mesh was independently generated by a different parameterized model. In such a scenario, given a mesh segmentation, it is possible to determine the model types (if more than one is available) and associated parameters that best fit the observations. Conversely, given an arrangement of models, it is possible to classify the mesh faces accordingly and thus determine a segmentation. Within this type of segmentation, known approaches for optimizing the overall segmentation include region growing [LMM], variational [SS05, JKS05] and hierarchical [AFS06] approaches.

Property-based: This category includes segmentations based on some surface property of interest held to some degree locally by the segments but not by the overall surface. This property is often not possible to indicatively measure at the atomic level of faces, thus precluding affinity-based approaches, and difficult or impossible to capture by a generative model approach. Such properties include symmetry [SKS06, MGP06, PSG*06, TW05], convexity [CDST97, LA05, KS06, KJS07], tubular shape [MPS*04] and texture [LMLR06, LMLR07].

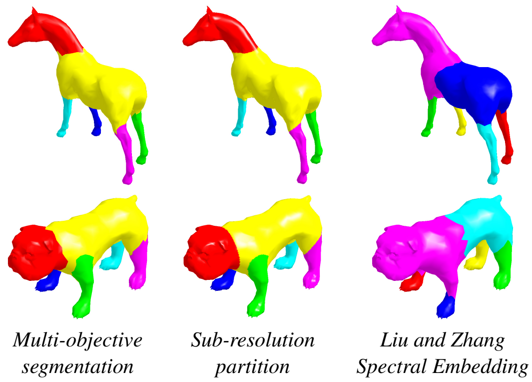
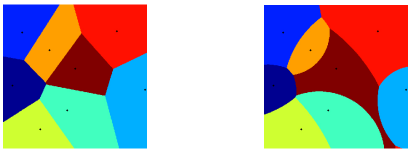


Figure 2: ***Left:*** *Multi-objective segmentation of quadrupeds. Unary objectives are symmetric head and body, ellipsoidal body, and narrow legs. We also have the n-ary objective asserting similar leg proportions and overall objective favoring convex parts. Bulldog model also incorporates symmetry constraints. (See Section 10 and Figure 9 for details.)* ***Center:*** *Given the nature of our approach, we are able to clip boundaries below mesh resolution.* ***Right:*** *Compare our results with those obtained using Liu and Zhang's spectral embedding approach [LZ07] using the same amount of segments.*

The methods presented thus far have natural intrinsic limitations corresponding to their category. Affinity based methods require that it be possible to evaluate the function in question at the face level for pairs of faces. This may not be possible for functions that require a larger context to sensibly evaluate, such as developability, tubularity, *etc.* Moreover, this approach is not amenable to optimizing multiple segment-dependant functions (*e.g.* if one part should be convex and another should not). Approaches based on generative models have the advantage of a continuous domain of optimization and possibility of gradient computation. Unfortunately they are also not applicable to some intuitive segmentation objectives that do not lend themselves to generative model formulations, such as convexity, symmetry, *etc.* Finally, the optimization approaches that address property-based segmentations are naturally tailored for, and tied to, the specific property they are segmenting with regards to. Furthermore this property is uniformly attributed to all parts in the segmentation.

3. Multi-objective shape segmentation

The users of segmentation algorithms possess a wealth of knowledge about the objects they are analyzing and working with. However, the implementation of an especially tailored



Standard Voronoi partition *Weighted Voronoi partition*

Figure 3: *Comparison of a standard Voronoi partition in 2D with randomly generated points (**left**) to a multiplicatively weighted Voronoi partition using the same centers and randomly generated weights. Notice how the weighted partition is able to generate circular boundaries.*

optimization algorithm is out of the realm of most users. To this end, we propose the decoupling of the objective function from the segmentation algorithm that optimizes it.

Additionally, we would like to accommodate for the use of heterogenous objectives. This is to say, each part of a segmentation is uniquely identified by a label and we would like to be able to assign varying objectives to different labels or groups of labels. Typical objectives a user might wish to attach to a label include that the associated part be narrow, flat, symmetric (reflective, n-fold rotational, axial), convex, compact, *etc.* Notice also however, that objectives need not be limited to referring to single labels, but can also refer to groups of labels and describe characteristics that the associated parts have with respect to each other. Intuitive examples include parallel, perpendicular, coaxial, similar (in proportions, for instance), *etc.*.

As an example, consider a hammer. This object is defined by Merriam-Webster as "a hand tool consisting of a solid head set crosswise on a handle and used for pounding." This naturally suggests two segmentation labels: *handle* and *head*. Moreover, it suggests the segmentation objective that the part labeled *head* be perpendicular to the part labeled *handle*. Our knowledge might further suggest that the parts should be approximately convex and that the part labeled *handle* should be narrow. The effects of incorporating multiple objectives into the segmentation of a hammer shape can be seen in Figure 1.

We thus cast the segmentation problem as an optimization minimizing an aggregate objective function which combines all given heterogenous objectives for the the segments of the shape at hand, specifically in our case, as a weighted sum. By formulating the problem in this manner, we address the previously mentioned motivations:

- By allowing different labels to be assigned different objective functions, we avoid the drawback of standard segmentation approaches which assume a uniform segmentation criterion for all parts.
- Since we assume the objective functions will be evaluated on a fully instantiated segmentation, we do not require that it be necessarily sensible to evaluate objectives on single face pairs as is the case in affinity-based approaches.

Algorithm 1 Connected partitioning

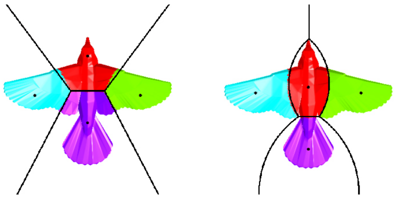
Let $\mathbf{C}$ be the given set of weighted Voronoi centers.
for all $f \in$ surface, $i \in [1,n]$ **do**
    $\mathbf{D}[f,i] \leftarrow$ weighted Voronoi distance from f to $\mathbf{C}[i]$
Initialize an empty queue $\mathbf{Q}$
Initialize segmentation indices **Seg** to 0
for $i = 1$ to n **do**
    $f \leftarrow \operatorname{argmin}_{\mathbf{f}} \mathbf{D}[f,i]$
    $\mathbf{Q}.push(f, priority = \mathbf{D}[f,i], centerid = i)$
while $\neg\mathbf{Q}.isempty()$ **do**
    $(f, priority, i) \leftarrow \mathbf{Q}.pop()$
    if $\mathbf{Seg}[f] \neq 0$ **then**
        $\mathbf{Seg}[f] \leftarrow i$
        for all f' adjacent to f **do**
            $\mathbf{Q}.push(f', priority = \mathbf{D}[f',i], centerid = i)$
return Seg

- As long as objective function evaluation is possible, a generative model is not strictly necessary (though certainly possible.)
- By means of n-ary objectives, the framework easily allows for assertions not just about parts but of inter-part relations as well.
- Objective functions can be reused. A non-expert user wishing to automate a segmentation can select from a set of already implemented, common functions (see Section 10 for examples.) Conversely, expert users who implement new objective functions can then make them available for reuse.

4. Parameterizing segmentations

In order to optimize a segmentation with respect to an aggregate objective function as described above, we need to be able to describe segmentations by means of some set of parameters. To this effect, we propose the use of a Voronoi space partition which is naturally parameterized by the spatial coordinates of a set of points, each of which corresponds to a segmentation label. This spatial partition naturally induces a segmentation on the shape by classifying each surface element according to the region of the partition that it occupies. Dealing with polygonal faces, we choose to classify them by their centroid. This scheme is simple and works well in practice but other alternatives are certainly possible.

A natural limitation of this approach would be that it is only capable of describing planar boundaries. It becomes possible, however, to describe curved boundaries (circular arcs in 2D) and non-convex regions by augmenting the centers with a distance scaling weight. Formally, a point p's distance to a center c with weight w is given by $||p-c||/w$. The result is known as a multiplicatively weighted Voronoi partitioning [OBSC00]. Figure 3 compares the standard Voronoi partition induced by a random set of points and the multiplicatively weighted Voronoi partition induced by points with the same locations but varying associated weights.



K-means seeding *Optimized result*

Figure 4: *Segmentation resulting from simultaneously optimizing for flatness of wings and tail, narrowness of body and compactness of tail, while incorporating the constraint information that the body and tail lie on the plane of symmetry and that the wings are symmetric to each other. (See Section 10 and Figure 9 for details.)* ***Left:*** *Result of k-means approach to initialization with optimal labeling.* ***Right:*** *Optimized result.*

Ensuring segment connectedness: This parameterized spatial partitioning can easily induce segmentations on shapes represented as point clouds, polygon soups, or any other representation whose elements' Voronoi partition membership can be evaluated. When shapes are represented using manifold meshes, voxels, or adjacency information is otherwise given, it may be desirable to ensure that the induced segments are connected. To achieve this we use a priority queue flooding scheme as introduced by Cohen-Steiner *et al.* [CSAD04] and successfully used in at least one other segmentation approach [SS05]. This process is detailed in Algorithm 1.

5. Initial center placement

The first step in our framework is to automatically find a coarse initial placement of partition centers which will serve as the initial guess for the optimization. In this section, we describe two such initialization approaches. Note that during initialization we assume Voronoi weights of 1 and optimize these weights in the later stage.

5.1. *K*-means center initialization

This approach has the advantages of simplicity and the fact that it produces a Voronoi partition by construction, given that it is based on distance to cluster centers. It also naturally produces compact, similarly sized segments.

The centers are initialized using furthers point initialization: choose the farthest pair of surface points and then, while centers remain to be chosen, select the point with maximum closest distance to the points chosen thus far. Then, surface elements are classified according to the closest center. Centers are now updated with the centroid of each segment and the process iterates till convergence.

The procedure is detailed in Algorithm 2. If adjacency information is available and connectivity of segments is desired, replace lines 7 and 8 with the queue approach described in Algorithm 1 for classification. Should a segment become empty, we select as its new center the surface element furthest from the current non-empty centers. Results of this approach are illustrated in Figures 4 (left), 5 (left), and 7 (center.)

Algorithm 2 *K*-means center initialization

1: // Choose n centers using furthest point initialization
2: $\mathbf{C}[1:2] \leftarrow$ two farthest surface points as initial centers
3: **for** $i = 3$ to n **do**
4: $\quad \mathbf{C}[i] \leftarrow \operatorname{argmax}_{\mathbf{x} \in \text{surface}} \min_i ||\mathbf{x} - \mathbf{C}[i]||^2$
5: **repeat**
6: $\quad$ // Classify surface elements
7: $\quad$ **for all** $f \in$ surface **do**
8: $\qquad \mathbf{S}[f] \leftarrow \operatorname{argmin}_{\mathbf{i}} ||f - \mathbf{C}[i]||^2$
9: $\quad$ // Update partition centers
10: $\quad$ **for** $i = 1$ to n **do**
11: $\qquad \mathbf{C}[i] \leftarrow \mathit{centroid}(\{f : \mathbf{S}[f] == i\})$
12: **until** no change in classification occurs
13: **return C**

5.2. Evolutionary center initialization

There may be cases for which the *k*-means alternative is not suitable but automation is still desired. Figure 5 (left), for example, shows such a case. To address this, we propose an evolutionary approach to initial center placement.

The space of all possible center locations in 3D seems prohibitively large, so our approach will choose initial center locations from among the set of surface elements. (Note that during this initialization, we will not optimize center weights.)

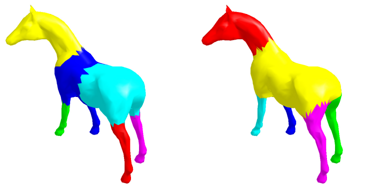
An individual in our population is naturally represented as an integer set $\mathbf{x}$ of size n where $i \in \mathbf{x}$ means surface point i is chosen as a center location.

Fitness of an individual will be determined by the the user-specified aggregate objective function. Given the heterogenous nature of the objectives however, the centers must be assigned labels prior to evaluation. In particular, we use a labeling procedure which is optimal with respect to unary objectives. This method is described in the next section.

As our selection strategy, we use tournament selection [Mic98], in which a small random subset of the previous population is considered and the individual with highest fitness is selected.

An individual can be mutated by randomly selecting one of its elements and replacing it by a new random center index which was not already present.

Finally, given two individuals $\mathbf{x}_1$ and $\mathbf{x}_2$, we can produce a crossover child $\mathbf{y}$ as follows. We first align the elements of



K-means seeding *Evolutionary seeding*

Figure 5: *There are cases in which a k-means approach to initialization (**left**) proves unsatisfactory. For such cases, we provide an evolutionary algorithm to choose initial center locations (**right**).*

these two sets to minimize the square distance of the corresponding centers. This can be done by computing all pairs distances between the centers from the first individual to those of the second individual and then efficiently solving for the optimal matching (for example, using the Hungarian algorithm [Kuh55, Mun57].) Once aligned, we apply a uniform crossover [Mic98]. In particular, we generate a uniformly random bit vector $\mathbf{b}$ and let $\mathbf{y}(i) := \mathbf{x}_1(i)$ if $\mathbf{b}(i) = 0$ and $\mathbf{y}(i) := \mathbf{x}_2(i)$ if $\mathbf{b}(i) = 1$.

This evolutionary seeding approach is specified in algorithm 3. In particular we use a population size of 50, a crossover fraction of 80%, a tournament size of 4, and allow for 20 iterations. Figure 5 shows the result of using this initialization strategy as compared to k-means and Figure 6 (top) shows the evolution of the aggregate objective function value during the run of the algorithm on this model.

6. Assigning labels to segments

A set of partition centers induces a shape segmentation, but due to the heterogenous nature of the objectives, labels must be assigned to each center prior to objective evaluation, as different labelings would produce different objective function values. This labeling step must be carried out after k-means initialization or at each step of the evolutionary approach to initialization prior to determining fitness (see Algorithm 3, line 7.)

When there are relatively few labels and the objective is not computationally expensive, an exhaustive approach of all permutations is feasible. As the number of segments grows, this approach quickly becomes intractable.

An alternative is to use a greedy approach. For some ordering of labels (which can be indicated by the user) we iterate over the label set and assign the current label to the segment which minimizes the increase in the aggregate objective function when evaluated on all labeled segments. While this is certainly an efficient and scalable alternative, there are no optimality guarantees.

If, however, for the purposes of addressing this assignment problem, we consider only the set of unary objectives

Algorithm 3 Evolutionary center initialization

1: // Initialize population $\mathbf{P}$
2: **for** $i = 1$ to *populationsize* **do**
3: $\quad \mathbf{P}[i] \leftarrow$ random subset of n indices into surface elements
4: **loop**
5: $\quad$ // Evaluate fitness
6: $\quad$ **for** $i = 1$ to *populationsize* **do**
7: $\quad\quad$ Optimal label assignment of $\mathbf{P}[i]$ based on *unary* objectives // See Section 6
8: $\quad\quad$ $\mathbf{F}[i] \leftarrow$ aggregate objective function value based on *all* objectives
9: $\quad\quad$ Remember best fitness and individual so far
10: $\quad$ $i \leftarrow 1$
11: $\quad$ // Generate crossover individuals
12: $\quad$ **while** $i \leq \mathit{crossoverfraction} * \mathit{populationsize}$ **do**
13: $\quad\quad$ $\mathbf{x}_1 \leftarrow \mathit{tournamentselection}(\mathbf{P}, \mathbf{F})$
14: $\quad\quad$ $\mathbf{x}_2 \leftarrow \mathit{tournamentselection}(\mathbf{P}, \mathbf{F})$
15: $\quad\quad$ $(\mathbf{x}_1, \mathbf{x}_2) \leftarrow \mathit{align}(\mathbf{x}_1, \mathbf{x}_2)$
16: $\quad\quad$ $\mathbf{P}'[i{+}{+}] \leftarrow \mathit{uniformcrossover}(\mathbf{x}_1, \mathbf{x}_2)$
17: $\quad$ // Generate mutation individuals
18: $\quad$ **while** $i \leq \mathit{populationsize}$ **do**
19: $\quad\quad$ $\mathbf{x} \leftarrow \mathit{tournamentselection}(\mathbf{P}, \mathbf{F})$
20: $\quad\quad$ $\mathbf{P}'[i{+}{+}] \leftarrow \mathit{mutate}(\mathbf{x})$
21: $\quad$ $\mathbf{P} \leftarrow \mathbf{P}'$
22: **return** individual with best fitness observed

for each segment, it is possible, to efficiently find an optimal matching.

Efficient optimal unary objective labeling

Assume that for each label j we have a set of k unary objective functions $\mu_{j,1}, \mu_{j,2}, \dots, \mu_{j,k}$. We can now build an $n \times n$ cost matrix C such that

$$C_{i,j} = \sum_k \mu_{j,k}(i)$$

which represents the cost of assigning label j to part associated with center i for all i, j pairs.

Given this cost matrix, we can now cast part labeling as an assignment problem which can be optimally solved in $O(n^3)$ by the Hungarian algorithm [Kuh55, Mun57]. Given C the algorithm will return an assignment vector $\mathbf{a}$ such that $\mathbf{a}_i = j$ indicates the part associated with segment i should receive label j to minimize the sum cost.

It should be noted that while only the unary objectives are used to solve the assignment problem in this approach, once the assignment is obtained, all objectives are evaluated to determine fitness during the evolutionary approach initialization (see Algorithm 3 line 8) as well as during the optimization described below.

7. Optimization

Now that we have addressed the matters of center initialization and label assignment we now focus on segmentation optimization.

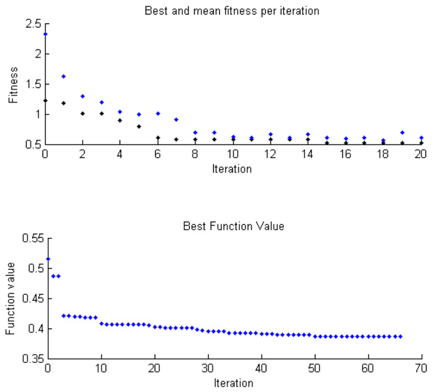


Figure 6: ***Top:*** *Mean (blue) and best (black) aggregate objective function value in population per iteration during evolutionary seeding of horse model.* ***Bottom:*** *Function value per iteration during pattern search optimization.*

If we wish to obtain a segmentation consisting of n segments, as previously described, we can consider the segmentation parameterized by a real vector $\mathbf{x}$ of dimensionality $m = 4n$ of the form

$$\mathbf{x} = (x_1, y_1, z_1, w_1, x_2, y_2, z_2, w_2, \dots, x_n, y_n, z_n, w_n)$$

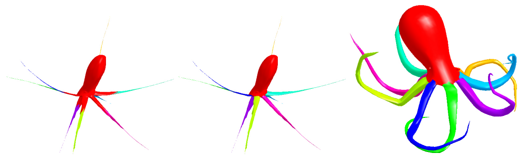
where (x_i, y_i, z_i) are the 3D coordinates of the i-th Voronoi center and w_i is its associated weight. The task is now to search for a value of $\mathbf{x}$ that minimizes the aggregate objective function when evaluated on the segmentation induced by $\mathbf{x}$.

Gradient-descent type approaches are not suitable given the discrete nature of the representation: a center must be altered sufficiently to induce a change in the labeling of at least one surface primitive. We instead choose generalized pattern search (GPS), which is a derivative-free, direct search method [Tor97, AJ03].

Given a pattern size Δ, the method works by, at each iteration, evaluating the aggregate objective function at all $2m$ neighbors formed by adding and subtracting Δ to each coordinate of the current $\mathbf{x}$. If any such neighbor produces a lower objective value, the iteration is considered successful, $\mathbf{x}$ is updated and Δ is multiplied by an expansion factor. Otherwise, the iteration is considered unsuccessful, no update of $\mathbf{x}$ occurs, and Δ is multiplied by a contraction factor. The algorithm terminates when Δ falls below a given threshold. In particular, we use an initial Δ of .2 times the shape's bounding box diagonal, an expansion factor of 2, a contraction factor of .5 and a Δ threshold of 10^{-6} times the bounding box diagonal. Figure 6 (bottom) shows the evolution of the aggregate objective function value during the run of the algorithm, for the final horse model segmentation of Figure 2.

8. Symmetry constraints

Recent methods allow for the robust automatic detection of symmetries in 3D shape [TW05, SKS06, MGP06, PSG*06]



Initial seg. in MDS | *Optimized seg. in MDS* | *Corresponding seg. in initial shape*

Figure 7: *Our approach can deal with shapes with convoluted articulations through the use of multi-dimensional scaling. The segmentation induced by the partition in MDS space is then easily mapped through correspondence with the original shape. Here we simultaneously optimize for narrow arms with similar proportions and an ellipsoidal head. See Section 10 and Figure 9 for details.*

and, whenever possible, shape processing algorithms should leverage this redundancy.

An advantageous property of our framework for parameterizing mesh segmentations is that it is easy to apply symmetry constraints to optimization parameters. For instance, if one segment is known to be symmetric to another, then the latter's Voronoi center position and weight can be generated by symmetry from the first. If a segment is known to lie on a plane of global symmetry, then its Voronoi center can be constrained to lie on said plane.

The advantages of exploiting this known redundancy are two-fold: Firstly, if a shape is known to have a certain global symmetry, then the segmentation is assured to have the same symmetry by construction. And secondly, the dimensionality of the optimization domain is reduced by removing parameters which can effectively be generated from others.

The user need only provide a constraint function which takes in the non-redundant parameters and produces the others from known symmetry.

Figure 2 illustrates this in our result on the bulldog model by constraining each pair of legs to be symmetric to each other, and constraining the body and head centers to lie on the plane of global symmetry. Similarly in Figure 4 we can constrain one wing to be symmetric to the other and the body and tail to lie on the plane of global symmetry.

9. Handling shapes under articulation

Finally, as a possible concern, one might point to the potential difficulty a space partitioning scheme might face when attempting to segment a highly articulated shape.

To address this concern we allow for the application of multi-dimensional scaling (MDS) as a pre-processing step which has proven useful in other segmentation approaches [KLT05, LZ07]. This approach pre-computes all pairs of geodesic distance between surface points and then finds a 3D embedding that approximately maps these geodesic distances to Euclidean ones. The result is an unfolding of ar-

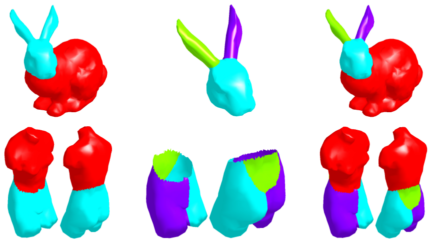


Figure 8: *Our approach can easily be used to segment by multiple objectives hierarchically. Here we first segment to maximize planar symmetry of parts, then by convex components, and then integrate the result.*

ticulation. One may now find a weighted Voronoi partition in the MDS space and optionally evaluate properties in this space or the original undistorted one as is most convenient on a case by case basis. The resulting segmentation is trivially mapped by mesh correspondence to the original mesh. Figure 7 illustrates this approach by cleanly segmenting a highly articulated octopus model.

10. Objective function definitions and results

In this section we define the objective functions used in our examples.

Given a segmented surface, let P be the surface segment associated with a given label. Define $P.scale_i$, $i \in [1,3]$ as the part's scales resulting from PCA, *i.e.* as the square root of the eigenvalues of the surface's 3×3 covariance matrix, sorted in descending magnitude. We may define the following unary objectives named with the adjective to which they intuitively correspond

$$narrow(P) := \frac{(P.scale_1 + P.scale_2)}{2 \cdot P.scale_3}$$

$$flat(P) := \frac{P.scale_3}{2 \cdot P.scale_1} + \frac{P.scale_3}{2 \cdot P.scale_2}$$

$$compact(P) := 1 - narrow(P)$$

Let us further define $P.c$ as part P's centroid and $P.axis_i$, $i \in [1,3]$ as the part P's i-th eigenvector also resulting from PCA. We may then define the objectives

$$planarsymmetric(P) := \min_{\mathbf{i}} surfdist(P, reflect(P, P.c, P.axis_i))$$

$$ellipsoidal(p) := surfdist(P, covarellipsoid(P))$$

We define $surfdist(S_1, S_2)$ as the sum area-weighted squared distance of points from S_1 to S_2 normalized by total area of S_1 and by its squared bounding box diagonal length. In turn $reflect(S, p, \vec{n})$ represents the planar reflection of surface S about the plane determined by point p and normal $\vec{n}$. This is similar to the symmetry metric used by Simari *et al.* [SKS06] but is non-iterative. Naturally, we define the $covarellipsoid(P)$ as the covariance ellipsoid of part P determined by its centroid, and PCA scales and axes.

Let us also define the following n-ary objectives that refer to label interrelations. Specifically

$$perpendicular(P_1, P_2) := |P_1.axis_1 \cdot P_2.axis_1|$$

$$similarproportions(P_1, P_2, \ldots, P_k) := \frac{1}{d^2} \sum_i ||P_i.scale - \hat{s}||^2$$

where d is the global surface's bounding box diagonal, $P_i.scale$ refers to the part's entire 3×1 scale vector and $\hat{s} = \frac{1}{k} \sum_i P_i.scale$.

Finally we define the global objective

$$convexparts(Seg) := \frac{1}{V^2}((\sum_{P \in Seg} H(P)) - V)^2$$

where Seg refers to the segmentation, $H(P)$ is part P's convex hull volume and V is the volume enclosed by the total original surface.

The table in figure 9 describes the specific objectives and constraints (where applicable) used in each of the results of Figures 1, 2, 4, 5 and 7. Figure 8 illustrates how, in addition to simultaneous objectives, our approach also easily allows for the optimization of hierarchical ones.

Our prototype was implemented in Matlab. Initializations vary from ~30 sec. for the k-means approach to ~10 min. for our evolutionary approach. Pattern search optimization time is an additional ~10 min. on average, but of course this will depend on the objective functions used and the efficiency of their implementation. All experiments were run on a Pentium M 2.13Ghz processor with 2Gb RAM.

11. Conclusions and future work

We have introduced the notion of multi-objective shape segmentation and the use of multiplicatively weighted Voronoi space partitioning as an approach to segmentation parameterization. We proposed seeding approaches to initialize the Voronoi centers, including a novel general-purpose evolutionary approach. We then presented strategies for automatically matching segments to their corresponding labels, including an efficient solution optimal for unary objectives. We showed how our approach can accommodate symmetry constraints which effectively reduce the dimensionality of the optimization domain when prior knowledge of the shape is available. Finally, we showed how even shapes under complicated articulation can be handled by our approach by using multi-dimensional scaling.

Specialized and effective segmentation algorithms tailored to specific objective functions will always have their place. However, we believe the area of general purpose segmentation algorithms which make minimal assumptions on the objectives is highly worthy of study. We hope this work will generate interesting research possibilities as there are several directions open to explore, including alternative segmentation parameterization schemes, as well as initialization and optimization approaches.

Model	*Labels*	*Aggregate objective function*
Hammer	*handle, head*	$3*narrow(handle)+perpendicular(handle,head)+convexparts(Seg)$
Horse	*head, body, leg_1, leg_2, leg_3, leg_4*	$planarsymmetric(head)+planarsymmetric(body)+ellipsoidal(body)+\dots$ $10*\sum_i narrow(leg_i)+similarproportions(leg_1,leg_2,leg_3,leg_4)+convexparts(Seg)$
Bulldog	*As above*	*As above plus* $compact(head)$ *Constraints that head and body centers lie on plane of global symmetry and parameters for leg_2 and leg_4 are reflected from those of leg_1 and leg_3 respectively.*
Dove	*body, $wing_1$, $wing_2$, tail*	$narrow(body)+10*\sum_i(flat(wing_i)+narrow(wing_i))+10*flat(tail)+\dots$ $compact(tail)+convexparts(Seg)$ *Constraint that body and tail lie on plane of global symmetry and parameters for $wing_2$ are reflected from $wing_1$.*
Octopus	$head, arm_1,\dots,arm_8$	$ellipsoidal(head)+\sum_i narrow(arm_i)+similarproportions(arm_1,\dots,arm_8)+\dots$ $convexparts(Seg)$ *(Objectives evaluated in MDS space.)*

Figure 9: *Aggregate objective functions used to obtain segmentations of each model.*

Acknowledgements: We would like to thank Richard (Hao) Zhang and Frank (Rong) Liu for providing us with the segmentation results of their method. Meshes used were obtained from the Princeton Shape Benchmark, Aim@Shape Project, and 3D Cafe. This research was partly funded by MITACS.

References

[AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Visual Computer 22*, 3 (2006), 181–193.

[AJ03] AUDET C., JR. J. E. D.: Analysis of generalized pattern searches. *SIAM Journal on Optimization 13*, 3 (2003), 889–903.

[CDST97] CHAZELLE B., DOBKIN D. P., SHOURABOURA N., TAL A.: Strategies for polyhedral surface decomposition: an experimental study. *Computational Geometry: Theory and Applications. 7*, 5-6 (1997), 327–342.

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM SIGGRAPH* (2004).

[JKS05] JULIUS D., KRAEVOY V., SHEFFER A.: D-charts: Quasi-developable mesh segmentation. *Eurographics* (2005).

[KJS07] KREAVOY V., JULIUS D., SHEFFER A.: Model composition from interchangeable components. *Pacific Graphics* (2007).

[KLT05] KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 21*, 8-10 (2005), 649–658.

[KS06] KRAYEVOY V., SHEFFER A.: Variational, meaningful shape decomposition. *ACM SIGGRAPH Sketches* (2006).

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM SIGGRAPH* (2003).

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 1-2 (1955), 83–97.

[LA05] LIEN J.-M., AMATO N. M.: Approximate convex decomposition of polyhedra. *ACM Symposium on Solid and Physical Modeling* (2005).

[LMLR06] LIU S., MARTIN R. R., LANGBEIN F. C., ROSIN P. L.: Segmenting reliefs on triangle meshes. *ACM Symposium on Solid and Physical Modeling* (2006).

[LMLR07] LIU S., MARTIN R. R., LANGBEIN F. C., ROSIN P. L.: Segmenting geometric reliefs from textured background surfaces. *Computer-Aided Design and Applications 4*, 5 (2007).

[LMM] LUKÁCS G., MARTIN R., MARSHALL D.: Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. *ECCV*.

[LZ07] LIU R., ZHANG H.: Mesh segmentation via spectral embedding and contour analysis. *Eurographics* (2007).

[MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *ACM SIGGRAPH* (2006).

[Mic98] MICHALEWICZ Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3 ed. Springer, 1998.

[MPS*04] MORTARA M., PATANÈ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. *ACM Symposium on Solid Modeling and Applications* (2004).

[Mun57] MUNKRES J.: Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics 5*, 1 (1957), 32–38.

[MW99] 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.

[OBSC00] OKABE A., BOOTS B., SUGIHARA K., CHIU

S. N.: *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2 ed. Wiley, 2000.

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3d shapes. *ACM SIGGRAPH* (2006).

[SKS06] SIMARI P., KALOGERAKIS V., SINGH K.: Folding meshes: Hierarchical mesh segmentation based on planar symmetry. *Eurographics Symposium on Geomtetry Processing* (2006).

[SS05] SIMARI P., SINGH K.: Extraction and remeshing of ellipsoidal representations from mesh data. *Graphics Interface* (2005).

[Tor97] TORCZON V.: On the convergence of pattern search algorithms. *SIAM Journal on Optimization 7*, 1 (1997), 1–25.

[TW05] THRUN S., WEGBREIT B.: Shape from symmetry. *ICCV* (2005).

[ZTS02] ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. *Computers and Graphics 26*, 5 (2002), 733–743.