Robust Region Detection via Consensus Segmentation of Deformable Shapes

Tongxi Lou

Informatik Department - Technische Universität München

Abstract

In recent years, recognition of 3D object based on regions instead of points is more and more popular. For this purpose, it is necessary to segment the mesh into several parts in a robust way and then characterize each partitions. In paper [4], a stable and efficient algorithm for 3D shape segmentation is proposed. The basic idea is to build a consensus segmentation based on a set of segmentations generated randomly by clustering points and consensus clustering. Intuition of this solution is that, a classifier built through optimization out of several classifiers should be more stable than them. The result of this approach is stable, accurate and suitable also to partial matching of a shape. At the end of this report, evaluations of robustness and runtime performance of this algorithm will be presented.

1 Introduction

1.1 Motivation

Detecting deformable 3D object has always been an interesting problem in Computer Vision and Computer Graphics. However due to the high complexity of 3D world, it is still a challenge. Previous works focus mainly on point based methods, which studies the pointwise correspondence between objects. Some recent researches present an alternative way by constructing region detectors. Some highlights include randomized cuts [1] and component tree [2]. Advances of Spectral Shape Analysis on each point of 3D mesh also make Region Detection and Characterization possible.

The basic idea in this paper is that, different segmentations of a shape may overfit or underfit some properties of shapes, but they can share some intrinsic information, so a segmentation based on several fine segmentations should be robust. For this purpose, the segmentation samples should be overfitting and contain more segments than we expect from an “ideal” segmentation. Then if a segmentation agree with these samples well, it can be a pretty good result that fits the model and contains little disturbance which should be random.

In this report an algorithm that build robust segmentation out of an ensemble of different segmentations is presented. The first step is building feature vectors for each point with Spectral Shape Analysis. Then aggregate the points with point-distance based on feature vectors, to get a set of segmentations. After that a measure of distance between segmentations is suggested and we optimize the segmentation by minimizing the segmentation-distance to all segmentations in set. This optimal segmentation is named “Consensus Segmentation”, as it consists with multiple reasonable segmentations we build.

1.2 Outline

Figure 1 shows the outline of this algorithm. First step is to feature each point of the shape with geometric spectrum. Laplace-Beltrami operator and its eigenfunctions are used. Then several samples of segmentation would be generated through clustering method to make an ensemble. Finally we can apply consensus clustering with/without forging base segments(for optimization) to get consensus segmentation as the robust segmentation we need.

2 Featuring the points

The segmentation of a shape in discrete world is actually aggregation of points and/or surface triangles. For the sake of universality, points are considered here. A measurement of points is
needed for aggregation, which is invariant of scale and deformation. The spectrum of the Laplace-Beltrami operator is isometrically invariant \[8\], so it is suitable here. The Laplace-Beltrami operator on a manifold is like below

\[ \Delta_M f = \nabla \cdot \nabla f \] (1)

The spectrum of it can be computed by solving Helmholtz equation, which leads to concept eigenvalues and eigenfunctions of Laplace-Beltrami operator

\[ \Delta_M \Phi_i = \lambda_i \Phi_i \] (2)

\(\lambda_i\) and \(\Phi_i\) is a couple of corresponding eigenvalue and eigenfunction.

Figure 2: Eigenfunctions

From left to right, the figures are \(\Phi_0, \Phi_1, \Phi_2, \Phi_3, \ldots\). Note that eigenvalues are non-negative, usually people order eigenvalues and \(\Phi_i\) with order \(\lambda_0 = 0 < \lambda_1 < \lambda_2 < \cdots\) and define \(\Phi_0\) to be constant.

Based on the concept “correlation of eigenfunctions”, people define “Global Point Signature(GPS)” as the feature of points as below

\[ g(p) = \left( \frac{\Phi_1}{\lambda_1}, \frac{\Phi_2}{\lambda_2}, \frac{\Phi_3}{\lambda_3}, \ldots \right)^\top \] (3)

The factor \(\frac{1}{\lambda_i}\) makes GPS invariant[2]. In continuous space, the GPS of a point is a vector with infinite length. In discrete space, the length of GPS is same as number of points.

The distance between two points is measured with inner product of GPS as below
Theorem 4.

\[ G(p, q) = \langle g(p), g(q) \rangle \]  

(4)

To make the computation applicable, we truncate the GPS and only use the first eigenfunctions. The influence of GPS length will be checked in Discussion session.

3 Ensemble of segmentations

With measurement of distance between points, it is possible to aggregate the points into several groups (segments) using clustering algorithms. One of them is k-means clustering. Suppose we need \( s \) segments for each segmentation, it starts by initiating \( s \) random centroids among all points. Then iteratively segmentation is built by assigning each point to segment with nearest centroid and recomputing centroids as mean/weighted mean of GPS in each segment until convergence[7]. The result of k-means clustering may come to local minimum or have extra groups, so the segmentations are not same and are not guaranteed to be the wanted robust segmentation.

The image on the left shows several segmentations built with k-means clustering method. In this case the mesh is segmented into 10 parts. Please note that they are just 5 samples of segmentation. The set of \( m \) segmentations after clustering is called Ensemble of Segmentations. It contains \( m \) random possibilities of segmentation.

The way of segmenting is same on some regions, for example the paws and tail are separated from other parts. But the elements differ from each other at some regions, like at neck and ears. A robust segmentation should follow the rules at commonly segmented regions and make compromises where conflicts occur. What we want is a segmentation that is as close as possible to every element in ensemble. For this optimization problem, definition of distance between segmentations is necessary.

4 Distance of segmentations

Segmentation shows the correspondence between points and segments. If two segmentations are the same on a pair of points, then the pair should be segmented into either same segment or both in different parts. It is tricky because different segmentations may have different labels for same segments. Example in figure Ensemble of Segmentations, the paws are separated from other parts successfully, but in some segmentations left paw belongs to “grew” segment, in some segmentations belongs to “red” or “purple” segments.

One solution of measuring how close two segmentations are, is

Theorem 5.

\[ d(x, y) = \sum_{i,j \in \text{points}} w_{i,j} (T_{x_i = x_j} - T_{y_i = y_j})^2 \]  

(5)

\( T_{x_i = x_j} \) equals 1.0 when \( x_i = x_j \) and 0.0 otherwise, \( w_{i,j} \) is product of areas around point \( i \) and \( j \)

Hence \( (T_{x_i = x_j} - T_{y_i = y_j})^2 \) equals to 1 when points \( i \) and \( j \) are segmented in different ways (\( i \) and \( j \) belong to one segment in one segmentation and are separated in the other segmentation). It reveals how two segmentations are different from each other locally. \( w_{i,j} \) is large when both points are significant (vertices of large triangles) and positive, so it is a good weight of how important this pair is.

5 Consensus segmentation
Now we have the measurement of how “close” two segmentations are, optimizing the distance of one segmentation from given segmentations is possible. Fréchet mean of the ensemble with above distance definition is a good candidate of optimal solution[6].

The optimization problem then becomes

\[ x^* = \arg\min_x \sum_{x_i \in \text{ensemble}} d(x, x_i) \]  \hspace{1cm} (6)

\( x^* \) is the Consensus Segmentation we pursue. It can be approached with Consensus Clustering in paper [3]. The final number of segments will be decided by clustering algorithm, but upper bound is necessary for optimization process.

Figure 4 shows an example of consensus segmentation of the cat model. We can see that consensus clustering shrinks the number of segments in consensus result down to 9.

6 Base segments

The optimization problem in last session is pointwise optimization. However there could usually be over 10k points or even more in a 3D shape. Then the algorithm would be tremendously memory- and time-consuming. So it is necessary to reduce the amount of computation.

One way is randomly sampling points or setting weights of points to zero as the common process in Big Data. There are usually redundant points in shape to assure the accuracy of model, but many points don’t contribute much to the robustness of segmentation. The work in !!! shows that the quality of segmentation maintains after setting most of weights to zero.

Another way is to represent a set of point by one point. As we see in Ensemble of Segmentations, the segmentations in ensemble usually are similar to each other. Some point-pairs are always segmented in same parts, some are segmented in same regions in many segmentations. So it is possible to first merge the points into several patches, then optimize the segmentation with respect to those patches. These patches are Base Segments.

We define base segment as the sets of points, whose elements are always segmented equivalently.

\[ \mathcal{B} = \{\{p\} | p \in \text{points}, x_i(p) \sim x_j(p), \forall x_i, x_j \in \text{ensemble} \} \]  \hspace{1cm} (7)

\( x_i(p) \sim x_j(p) \) means \( \exists f_{ij} : \{s\} \rightarrow \{s\}, f_{ij}(x_i(p)) = x_j(p) \), there exists a mapping \( f_{ij} \) from \( s \) labels to \( s \) labels, such that the segmentations \( f_{ij}(x_i) \) and \( x_j \) are same on point \( p \).

Generally there are much fewer base segments than points. Figure 5 show the base segments of cat model. There are large base segments like around eyes and at paws, there are also small strips like at neck where the boundaries vary from segmentation to segmentation. Because in consensus clustering we minimize the distance between segmentations in ensemble and consensus segmentation, and by definition the segmentation of points and of base segments are equivalent, clustering of segments is equivalent to clustering of points.

If we denote the segmentation on base segments as \( y_i \), namely \( y_i(\mathcal{B}) = x_i(p) \forall p \in \mathcal{B} \), then consensus clustering becomes

\[ y^* = \arg\min_y \sum_{y_i \in \text{ensemble}} d_B(y, y_i) \]  \hspace{1cm} (8)

The difference between \( d_B \) and \( d \) is the weights. \( d \) uses product of areas around two points as weight of a pair, while \( d_B \) uses product of areas of two base segments.
7 Experimental results

The result of single object has been shown in session Consensus segmentation. It looks fine but doesn’t represent the performance of this algorithm. We will present the results with deformations and disturbance, how consensus segmentation classifies regions to show the algorithm’s robustness, and run-time performance to show its efficiency.

7.1 Segmentation with deformation and disturbance

In figure 6 results of consensus segmentation are presented(dataset from SHREC’ 10). The first image shows the original shape, the second one and the rest are deformed edition. The third and fourth images are scaled. At the second line non-isometric deformations and noise are added to shape.

In this experiment, we apply region descriptor to classify the consensus segments. One constructor of region descriptor is average of point descriptors in region

\[ \beta(R) = \sum_{p \in R} \alpha(p)da(p) \] (9)

\( da(p) \) is neighboring area of a point,
\( \alpha \) is point descriptor,
\( \beta \) is region descriptor

We can see that the consensus segmentation and region classification based on GPS is robust under deformation(isometric and non-isometric) and degradation.

7.2 Segmentation for shape matching

The real world is complicated and camera often takes information of unknown or incomplete objects. So it is better if segmentation separate objects in a constant way. Figure 7 show the result of segmentation and classification using consensus segmentation and region descriptor. This algorithm separate different parts from different shapes robustly, even when the shapes are locally and overall distinctive. The arms, trunks and heads are clearly and nicely devided and decided.

7.3 Detector repeatability

The last subsections show intuitivly how fine Consensus Segmentor is, now we test the segmentation mathematically with repeatability about the exactness of consensus segmentation. Overlap is the ratio of area of overlapping region to area of summed region of a segment in two shapes.

Figure 6: Consensus segmentations on a human shape, with deformations and disturbances

Figure 7: Consensus segmentations on a human shape, with deformations and disturbances
Theorem 10.

\[
o(R) = \frac{a(R_1 \cap R_2)}{a(R_1 \cup R_2)} \quad (10)
\]

\( R_i \) is the region of segment \( R \) in shape \( i \), \( a \) is the area.

If the segmentation is absolutely exact, then overlap is 1.0; if the segment is totally different in two shapes, then common area is 0.0, thus overlap is 0.0.

Repeatibility is defined above overlap. It shows the amount of segments whose overlap is higher than a given criterion. If all segments are exactly separated with high enough overlap, then repeatability is 1; if only half of segments satisfy the given overlap, then repeatability is 0.5. Below is the repeatability of some segmentations with given number of segmentations in ensemble.

According to Figure 8, when \( m \geq 5 \), most segments of consensus segmentation are fairly exact. For example when we generate ensemble with 5-means clustering, 80% of segments have overlap equal to or over 0.8. So consensus segmentation is not only intuitively robust, but also mathematically reliable.

### 7.4 Time efficiency

<table>
<thead>
<tr>
<th>m</th>
<th>10</th>
<th>25</th>
<th>50</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td>ensemble</td>
<td>1.94</td>
<td>4.64</td>
<td>9.24</td>
<td>14.05</td>
</tr>
<tr>
<td>sconsensus</td>
<td>0.02</td>
<td>0.12</td>
<td>0.32</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 1: Run time performance

With the implementation of base segments, the algorithm should be time efficient at consensus clustering part and run time of this part should be less relevant to the size of shape.

This algorithm has been tested on real machine with Intel i7 CPU(4 cores), 8GB RAM and segmenting TOSCA dataset. The parameter of ensemble is \( k = 5 \), \( s = 10 \) and \( len(g) = 50 \) (length of GPS vector is 50). Ensemble of segmentations is build in Matlab, consensus clustering is implemented with C++. Table 1 shows the run time of algorithm w.r.t. different number of segmentations(ensemble size). We can see that the first part is more time consuming, perhaps because of the running environment and complexity of calculating eigenfunctions based on the entire shape. The second part runs faster but run time grows rapidly, because it is an optimization problem with iterative approach. But overall, this algorithm is efficient and base segment works well on reducing computation.

### 8 Conclusion

In paper [4], a robust and efficient algorithm for 3D shape segmentation based on spectral shape analysis, clustering and consensus clustering is proposed. The result, consensus segmentation, is invariant of deformation and disturbance, and offers meaningful partitions for partial recognition between different kinds of objects. With region descriptor from point features built in first step, it can also be a fine 3D region detector. To make the algorithm efficient and scalable to large mesh, base segment that make consensus clustering switch from points to patches is also proposed. At last, intuitive and quantitative evaluation of this algorithm is presented.
References


