Hierarchical Clustering of Streamtubes

Song Zhang David H. Laidlaw*

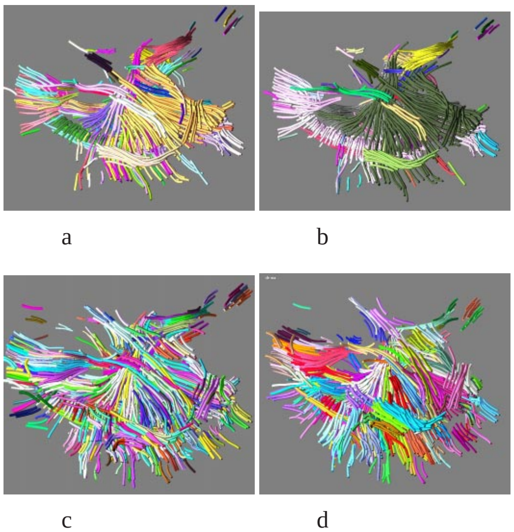

Figure 1: Results from applying clustering methods to the streamtubes. The streamtubes within one cluster share a single color. (a,b) show the clustering results from the minimum algorithm. (b) has fewer clusters than (a). (c,d) show the clustering results from the maximum algorithm. (d) has fewer clusters than (c).

1 Introduction

We apply hierarchical clustering methods on streamtubes for visualization and analysis.

Streamtubes are integrated in the major eigenvector field of the DTI data set. In a $256 \times 256 \times 50$ data set, our algorithm can generate tens of thousands of streamtubes. It is hard to find features in a dense set of undistinguished tubes. Thus it is important to impose some structural information on the streamtubes for visualization and interpretation purposes. Hierarchical clustering produces a dendrogram that groups objects into different number of clusters in a continuous way. We apply some clustering methods on a set of streamtubes and found that the streamtubes correlating to major neural structures tend to cluster together because of their shape similarities. Also, different distance criteria produce different types of clusters. The dendrogram produced by the hierarchical clustering methods has the potential to be utilized by visualization applications to interactively display the streamtubes at different level-of-details.

2 Method

We first generate a set of streamtubes, and then cluster them with two different hierarchical clustering methods.

*Box 1910, Brown University, Providence, RI 02912; {sz,dhl}@cs.brown.edu

2.1 Generating streamtubes

The algorithm to generate streamtubes works as following (for details please see [3]):

First, we define linear anisotropy region in the DTI data set using Westin's metrics [1]. And then we integrate streamtubes in the major eigenvector field of the DTI data. The integration starts from a seed point in the data volume, and stops when the streamtube hits the data boundary, enters a low linear anisotropy region, or makes a high curvature change of direction.

There are several parameters used by the streamtube generating algorithm. For example, the placement of the seed points, the threshold for linear anisotropy region, the threshold for the length of the streamtube, etc. Most of them affect the number of streamtubes generated.

2.2 Clustering streamtubes

We use agglomerative hierarchical clustering methods [2] to cluster the streamtubes. The general algorithm starts with n singleton clusters and form the sequence by successively merging clusters. The algorithm works as follows:

1. Given a set of n singleton clusters.
2. Merge the two nearest clusters
3. Repeat 2. until the specified number of clusters are generated.

If we set the specified number of clusters to 1, we produce a dendrogram. The interesting part of the algorithm lies within the word "nearest". We have to define the distance between two streamtubes, and the distance between two clusters of streamtubes.

The distance between streamtubes are defined as following:

$$D_l = \frac{\int_{s_0}^{s_1} \max(\mathrm{dist}(s) - T_l, 0)\mathrm{d}s}{\int_{s_0}^{s_1} \max(\frac{\mathrm{dist}(s)-T_l}{|\,\mathrm{dist}(s)-T_l|}, 0)\mathrm{d}s},$$

where s parameterizes the arc length of the shorter trajectory, and $\mathrm{dist}(s)$ is the shortest distance from location s of the shorter trajectory to the longer trajectory. T_l ensures that we label two trajectories as different if they differ significantly over any portion of the arc length.

If the minimum distance between any two streamtubes from two clusters is used for the distance between these clusters, the method is called the nearest-neighbor cluster algorithm, or minimum algorithm. This algorithm tends to generate long chains of clusters if there are bridges between two otherwise separated clusters. This is often called "chaining effect".

When the maximum distance between any two streamtubes from two clusters is used for the distance between these clusters, the algorithm is called the farthest-neighbor clustering algorithm, or maximum algorithm. This algorithm tends to minimize the diameter of a cluster.

3 Results

The DTI and T2-weighted MRI data set is collected from a normal volunteer. We generate a set of streamtubes from the DTI data set. The seed points are picked from every third data point in each dimension, the average linear anisotropy on the streamtubes are higher than 0.3. We have generated a set of $1,403$ streamtubes. We then applied the minimum algorithm and the maximum algorithm on this set of streamtubes. Figure 1 shows the results. The color is used only to distinguish between different clusters. Figure 1 (a,b) shows the two clusters generated from the minimum algorithm. They are from the same dendrogram but at the different level, Figure 1 (a) is lower in the dendrogram and has more clusters. From Figure 1 (a,b) we found that streamtubes in the vicinity of major neural structures tend to cluster together by the minimum algorithm. For example, in Figure 1 (a), the yellow, pink, and light blue tubes occupy the internal capsule area, and the streamtubes in the middle section of the corpus callosum are all within the green cluster. It is even more obvious in Figure 1 (b), where the internal capsule area is almost totally occupied by the green cluster. This phenomena is partly due to the “chaining effect”. Although the streamtubes around corpus callosum are distributed over a long distance, they are still connected by locally similar streamtubes, thus forming few clusters.

On the other hand, Figure 1 (c,d) show the clusters generated by the maximum algorithm. The streamtubes around internal capsule and corpus callosum are broken into a number of clusters, since remote streamtubes will not be clustered into one group by the “bridges”. Similar streamtubes form local clusters that has small intra-cluster distance. If we choose a representative single streamtube for each cluster, the abstractive set of streamtubes from these clusters have the potential to catch the overall structure while getting rid of the visual clutters.

4 Acknowledgments

This work was supported in part by grant CA69246 from the National Institutes of Health, by the Human Brain Project (NIDA and NIMH), by grants from the National Science Foundation (CCR-0086065, CCR-0093238, and EIA-9724347). Thanks to Dr. Susumu Mori at Johns Hopkins for the human brain DT-MRI dataset.

References

[1] Westin C.-F., Peled S., Gubjartsson H., Kikinis R., and Jolesz F.A. ”Geometrical diffusion measures for MRI from tensor basis analysis.” In *Proceedings of ISMRM*, 1997.

[2] Duda R.O., Hart P.E., Stork D.G. “Pattern Classification 2nd Edition.” Wiley-Interscience Publication.

[3] Zhang S., Demiralp C., Keefe D.F., DaSilva M.J., Laidlaw D. H., Greenberg B.D., Basser P.J., Pierpaoli C., Chiocca E.A., and Deisboeck T.S. ”An immersive virtual environment for DT-MRI volume visualization applications: a case study.” In *IEEE Visualization*, 2001.