DECOMPOSITION OF BRANCHING VOLUME DATA BY TIP DETECTION

Wei Ma[1]*, Bo Xiang*[2]*, Xiaopeng Zhang*[2]*, and Hongbin Zha*[1]

Key Laboratory of Machine Perception (Ministry of Education), Peking University, P. R. China[1]
LIAMA-NLPR, CAS Institute of Automation, P. R. China[2]
Email: mawei@cis.pku.edu.cn, helenexiang@gmail.com
xpzhang@nlpr.ia.ac.cn, zha@cis.pku.edu.cn

ABSTRACT

We present an approach to decomposing branching volume data into sub-branches. First, a metric is proposed for evaluating local convexities in volumetric data, and it is a criterion for global selection of tip points. Second, a multi-path growing strategy is adopted to segment the volumes based on a DFS transformation starting from the tips. Experiments show that this approach is capable of generating desirable components and reasonable segmentation boundaries of a volume.

Index Terms— volume decomposition, feature point detection, distance transformation

1. INTRODUCTION

Shape decomposition gains a great interest in computer vision and graphics due to its broad applications, such as in shape analysis [1], skeleton extraction [12], texture mapping [11], and modeling by examples [5]. Many approaches have been proposed in recent years aiming at different kinds of objects: from medical MRI or CT data [15] to laser data [4], from mesh models [2] [7] to point clouds [13] and volume data [3]. In the paper, we concentrate on the decomposition of branching volume data, which is useful to medical and botanical applications in automatic endoscopic navigation [6] and accurate skeleton extraction for reconstruction [15, 4].

Among the decomposition approaches, features are widely used as seeds for clustering [9, 8, 14, 10]. Katz et al. [8] detected feature points by choosing those with local maxima of geodesic distances in mesh models, and chose those lying on a convex hull as seed features. Zhang et al. [14] sampled seed faces based on distances to several uniformly distributed faces. Yamazaki et al. [13] treated points with local maxima of geodesic distances in point-based models as the feature points, followed by a hierarchical procedure to segment the models composed of supernodes which are determined by the local features. In the approaches mentioned above, feature points are locally detected and there are no quantitative criterions to evaluate potential features in a global view. Krayevoy et al. [10] presented a convexity metric to generate seed patches, which relies only on a single threshold parameter. Their approach is effective for mesh models. However, the metric is not appropriate for volumetric data and the number of the decomposed components is not necessarily an indicator of the volume structures but depends on the threshold. Zhou et al. [15] adopted a DFS (Distance From a Source point) transformation to find the tip features of branching volumes. However, their seed points require manual labeling which is inconvenient when lots of disconnected branches appear in a data set. Furthermore, they provide no description for the local properties of tip features, and thus their decomposition algorithm suffers seriously from noises. In the paper, we present a quantitative criterion to evaluate local convex feature voxels in volumetric data. Feature voxels with large values are treated as tips. Next, with these tips, decomposition is performed by clustering, i.e. voxels on the same branch are grouped together based on the DFS transformation with the same index corresponding to a tip. A further segmentation procedure at ramifications, where more than one sub-branches meet, produces smooth and reasonable boundaries between branches.

The remainder of the paper is organized as follows: In Section 2, we present a tip feature detection algorithm in detail. Section 3 describes how to decompose a branching volume into desirable components. Results in Section 4 demonstrate the validity of our algorithm. Section 5 draws conclusions.

2. TIP FEATURE DETECTION

In this section, we address the issue of tip feature detection in volumetric data. As [10], our feature detection algorithm is based on a convexity criterion. There are two new aspects in the algorithm. 1) It is able to quantify local convexity in the volumetric data by using a score representing local sharpness. 2) It allows for global intuitive feature selection based on the scores.

Our feature detection algorithm is designed for volumetric data sets. Other 3D data representations such as meshes should be converted to voxels first. To compute the sharpness property near a voxel, its L-ring neighborhood (The 1-

978-1-4244-1764-3/08/$25.00 ©2008 IEEE

ring neighborhood of the voxel is its 26 closest neighbors) are used. L is an integer given by users.

Given a voxel $\vec{X}_0$ in a volume, we can intuitively denote the extending quantity of a voxel $\vec{X}_n^l$ in ring l along a direction vector $\vec{V}_i$ as:

$$r_n^l(\vec{V}_i) = (\vec{X}_n^l - \vec{X}_0)^T \vec{V}_i. \tag{1}$$

The sharpness of $\vec{X}_0$ is defined as the sum of the extending quantities of all the L rings of voxels along an optimal direction $\vec{V}_0$:

$$S(\vec{V}_0) = \sum_{l=1}^{L} \frac{1}{N^l} \sum_{n=1}^{N^l} r_n^l(\vec{V}_0), \tag{2}$$

where, N_l is the voxel number in ring l.

An intuitive 2D illustration is given in Fig. 1, in which the voxel $C2$ is said to be sharper than the voxel $C1$ since that with the same L, the neighborhood of $C2$ is obviously more extended along the direction vector $\vec{V}_2$ than that of $C1$ along $\vec{V}_1$. Here, $\vec{V}_1$ and $\vec{V}_2$ are drawn to be nearly optimal by intuition.

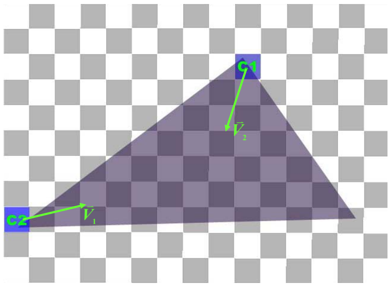

Fig. 1. A voxelized triangle.

To compute the optimal direction $\vec{V}_0$, we maximize the quantity given by:

$$S(\vec{V}_i) = \sum_{l=1}^{L} \frac{1}{N^l} \sum_{n=1}^{N^l} r_n^l(\vec{V}_i), \tag{3}$$

i.e.

$$\vec{V}_0 = \underset{V_i}{\operatorname{argmax}}(S(\vec{V}_i)), \tag{4}$$

The constraints for (3) are:

$$\vec{V}_i^T \vec{V}_i = 1, \tag{5}$$

and

$$r_n^l(\vec{V}_i) > 0, \forall \vec{X}_n^l. \tag{6}$$

However, since the distances between a voxel and its closest neighbors (6 face ones, 12 edge ones and 8 corner ones) are different, we can not simply evaluate the extendedness of voxel $\vec{X}_n^l$ by its Euclidian distance to $\vec{X}_0$ as defined in (1). To solve this problem, we recompute the extendedness ring by ring and replace the real distance from a voxel to one of its neighbors in the upper ring with 1. This calculation scheme is similar to that of the 1-1-1 distance propagation in [15]. $r_n^l(\vec{V}_i)$ is recomputed recursively as follows:

```
$r_n^l = 0$
$l' = l$
while $l' > 0$
    find a closest neighbor $\vec{X}_n^{l'-1}$ for $\vec{X}_n^{l'}$ in ring $l' - 1$
    $r_n^l = r_n^l + \frac{(\vec{X}_n^{l'} - \vec{X}_n^{l'-1})^T \vec{V}_i}{||\vec{X}_n^{l'} - \vec{X}_n^{l'-1}||}$
    $l' = l' - 1$
endwhile
```

To obtain tip features according to the mathematical description above, we first compute the optimal axis $\vec{V}_0$ by maximizing (3) with constraints (5) and (6). The sharpness of $\vec{X}_0$ is then computed by (2). All the voxels' sharpness quantities form a score field. Based on the field, an initial feature selection is performed by a local suppression (in L-ring neighborhood) to eliminate the competition arising from voxels near sharp voxels. Next, we sort the remaining voxels in order of their scores. At last, the N (given by users) top features are chosen as tips used for the following decomposition.

3. VOLUME DECOMPOSITION

In this section, we proceed to decompose a branching volume with its tip features by a DFS transformation [15] and generate reasonable boundaries between segments by a segmentation procedure at ramification points.

3.1. Distance transformation

DFS transformation is a recursive voxel-by-voxel propagation which brings each voxel in the volume a distance value to the single start point. In our algorithm, we perform N (the number of tips) times of DFS transformations on the volume. Each transformation starts from a tip feature point and generates a distance field of the volume to that specified tip.

3.2. Ramification point determination

Based on the distance transformation, voxels with the same distance value can be regarded as a layer, which is similar in part to a sphere wave with the start point as its center. If all the voxels in a layer are tightly connected to each other, they are clustered into the same branch. On the other hand, when the voxels are not neighborhood connected side-by-side, a ramification point appears at this location. Therefore, we check every layer in an order of the related distance value ascending from the start point. If a layer satisfies the ramification condition mentioned above, then more than one branch is detected. And this kind of layers is called ramification layers. Furthermore, according to the local connectivity of the voxels, the discontinuous layer can be divided into $M(M > 1$, the number of post-branches) sets that are self-connected. Each set looks like a mouth to a certain post-branch.

3.3. Segmentation at ramification points

As mentioned above, a ramification layer can be divided into M sets (generally two sets as shown in green in Fig. 2(a)) which belong to different sub-branches. Connecting the centers of the two sets in Fig. 2(a), we get a direction vector. Along this direction, we search for all the voxels linking the two sets with the shortest path. After cutting off these voxels, we mark the remaining connected voxels in pre-ramification layers as parts belonging to the branch from the start tip feature point. Therefore, different components can be distinguished as shown close to a real ramification point in Fig. 2(b). The above algorithm is based on an assumption that sub-branches are locally cylindrical.

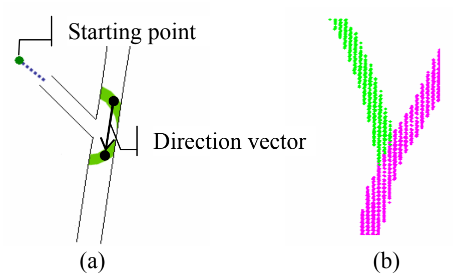

Fig. 2. Segmentation at a ramification point.

3.4. Clustering with a multi-path growing strategy

From the N tip features, we have N paths corresponding to each sub-branch. When a voxel is reached by a certain path, we mark the voxel as a component corresponding to this path. The N paths growing from up to down will stop when they meet ramification points. Then we check each ramification position. 1) If only one mouth near the position is unmarked (open mouth), we choose a principle path with the largest radius from the meeting paths to keep it growing to the next part of the volume that has not been reached before. The rest paths end the growth and will never participate in calculation in the post-loops. 2) If more than one open mouth exist, these paths just stop to wait for other paths passing by. The process is repeated until the number of the growing paths down to one. Then the remaining voxels of the volume are clustered into the only existing path.

Finally, the decomposition of the branch volume is completed with N clusters corresponding to the tip feature points.

4. EXPERIMENTS

We test our algorithm on two branching volumes, a sunflower and a silver willow. Table 1 provides the statistics and parameters during their decomposition. The second column lists the total number of voxels. The third column presents the value of L for controlling the size of the neighborhood rings in feature detection. The fourth and the fifth columns show the consensus of the number N of top features and that of final sub-branches.

Table 1. Statistics and parameters.

Object	Voxels	Rings L	Top Features N	Final Sub-branches
sunflower	12,367	8	13	13
willow	49,883	10	18	18

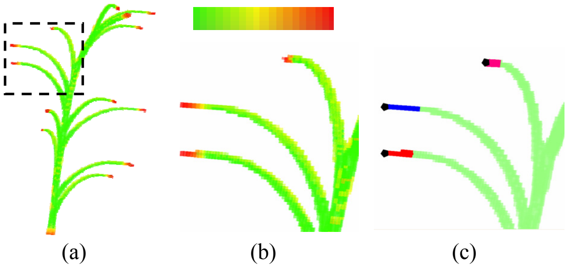

Fig. 3. Tip feature detection of the sunflower. (a) is the score field visualized by a pseudo color bar on the top of (b) which represents ascending scores from left to right. The bottom of (b) is a close-up of the box in (a); (c) illustrates the L-ring (L=8) neighborhoods of three tip features, respectively in magenta, blue and red color.

Some results during the tip feature detection of the sunflower are presented in Fig. 3. A score field (as shown in Fig. 3(a) and (b) in case that L is 8), is built as described in Section 2. After a local (8-ring neighborhood) suppression, 43 convex features are extracted. An illustration for the covering areas of the 8-ring neighborhoods corresponding to three of the features (the black spots) is presented in (c). Next, given the number N by users, N top features will be selected out as the tips of the branching volume, as shown in Fig. 4(a), from which we can see that the tips are correctly indicated by the N top features. For more complex branching volumes, whose tip numbers are difficult to count by users, histograms of scores can be used since the scores of the tips are definitely larger than those of the other convex features in our experiments.

Fig. 4 shows the results of volume decomposition by correctly specified tip features (Fig. 4(a)). The first column is the sunflower and the second column is the silver willow. Different components in each branching volume are distinguished with different colors. The lowest row focuses on the details of the boundaries of the sunflower and the willow respectively. It can be seen that the volume decomposition generated by our algorithm is reasonable and the boundaries between sub-branches are quite accurate. In addition, we can also see that our algorithm is competent to branching volumes with multi-level structures.

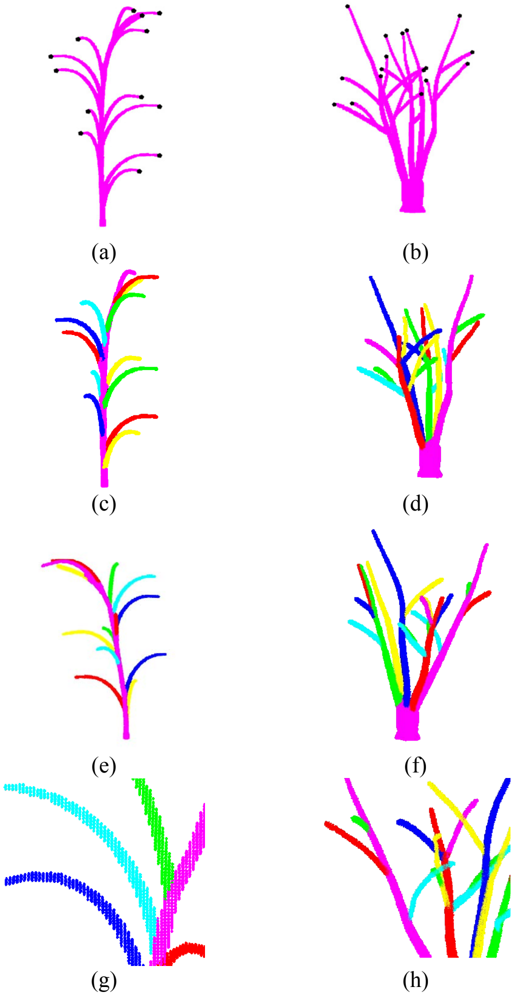

Fig. 4. Volumn decomposition (left column: sunflower; right column: willow). (a) and (b) show the tip features with black spots; (c) (d) (e) and (f) show the segmentation results in different view points; (g) and (h) are two close-ups of the sunflower and the willow respectively.

5. CONCLUSIONS

We have presented an approach to decomposing branching volumes based on a tip feature point detection algorithm and a multi-path growing strategy. The algorithm has two benefits. First, an evaluation criterion of convex features is proposed. With the criterion, tip feature points are selected out in a global view as seeds for decomposition. Second, the boundaries between segments generated by our algorithm are reasonable and go along the natural seams of the models. Additionally, since our convex feature evaluation criterion is not limited to branching volumes, we should apply the algorithm to common shape models in the future.

6. ACKNOWLEDGMENTS

This work was supported in part by NKBRPC (No. 2004CB318000), NHTRDP 863 Grant (No. 2006AA01Z301, No. 2006AA01Z302 and No. 2007AA01Z336), Key Grant Project of Chinese Ministry of Education (No. 103001).

7. REFERENCES

[1] M. Attene, S. Biasotti, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno, “Computational methods for understanding 3d shapes,” *Computers & Graphics, Special Issue on Computer Graphics in Italy*, vol. 30, no. 3, pp. 323–333, 2006.

[2] M. Attene, S. Katz, and M. Mortara, “Mesh-segmentation - a comparative study,” in *Int. Conf. on Shape Modeling and Applications 2006*, pp. 14–25, 2006.

[3] D. Brunner and G. Brunnett, “An extended concept of voxel neighborhoods for correct thinning in mesh segmentation,” in *Proc. of the 21st Spring Conference on Computer Graphics*, pp. 119–125, 2005.

[4] A. Bucksch and H. Appel van Wageningen, “Skeletonization and segmentation of point clouds using octrees and graph theory,” *ISPRS Symp. : Image Engineering and Vision Metrology Vol. XXXVI. ISPRS Commission V Symposium*, pp. 1–6, 2006.

[5] T. Funkhouser, M. Kazhdan, P. Shilane, and P. Min, “Modeling by example,” in *Proc. of SIGGRAPH 2004*, pp. 652–663, 2004.

[6] H. K. Hahn, B. Preim, D. Selle, and H. O. Peitgen, “Visualization and interaction techniques for the exploration of vascular structures,” in *IEEE Visualization 2001*, pp. 395–402, 2001.

[7] A. Jagannathan and Eric L. Miller, “Three-dimensional surface mesh segmentation using curvedness-based region growing approach,” *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2195–2204, 2007.

[8] S. Katz, G. Leifman, and A. Tal, “Mesh segmentation using feature point and core extraction,” *The Visual Computer*, vol. 21, no. 8-10, pp. 865–875, 2005.

[9] S. Katz and A. Tal, “Hierarchical mesh decomposition using fuzzy clustering and cuts,” *ACM Tran. on Graphics*, vol. 22, no. 3, pp. 954–961, 2003.

[10] V. Krayevoy and A. Sheffer, “Variational, meaningful shape decomposition,” in *Proc. of SIGGRAPH 2006 Sketches*, 2006.

[11] B. Levy, S. Petitjean, N. Ray, and J. Maillot, “Least squares conformal maps for automatic texture atlas generation,” in *Proc. of SIGGRAPH 2002*, pp. 362–371, 2002.

[12] J. M. Lien, J. Keyser, and N. M. Amato, “Simultaneous shape decomposition and skeletonization,” in *Proc. of the 2006 ACM symp. on Solid and physical modeling*, pp. 219–228, 2006.

[13] I. Yamazaki, V. Natarajan, Z. Bai, and B. Hamann, “Segmenting point sets,” in *Proc. of IEEE Int. Conf. on Shape Modeling and Applications*, pp. 4–13, 2006.

[14] H. Zhang and R. Liu, “Mesh segmentation via recursive and visually salient spectral cuts,” in *Proc. of Vision, Modeling, and Visualization*, pp. 429–436, 2005.

[15] Y. Zhou and A. W. Toga, “Efficient skeletonization of volumetric objects,” *IEEE Tran. on Visualization and Computer Graphics*, vol. 5, pp. 196–209, 1999.