# Volume Decomposition for Hierarchical Skeletonization

Xiaopeng Zhang[1,2], Jianfei Liu[1,2], Marc Jaeger[3] and Zili Li[4]

[1] Sino-French Laboratory LIAMA, Institute of Automation, CAS, Beijing 100190, China.
[2] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China.
[3] Project DigiPlante, CIRAD-AMAP /INRIA-Saclay, 34398 Montpellier CEDEX 5, France.
[4] Department of Electronics & Info., Huazhong Univ. of Sci. & Tech., Wuhan 430074, Hubei, China.

***Abstract*—Hierarchical skeletons and shape components are important shape features, and they are useful for shape description and shape understanding. Techniques to extract shape components and hierarchical skeletons from volume data are analyzed in this paper based on multiple distance transformations. The application of volume decomposition for the extraction of hierarchical skeletons is emphasized and specified here. This work includes an establishment of the hierarchical structure of the object volume, a decomposition of the volume into simple sub-volumes, an extraction of compact skeleton segments corresponding to each independent sub-volume, and a connection of these skeleton segments into a hierarchical structure reflecting the organization the initial data.**

***Index Terms*—Volumetric Shape, Decomposition, Skeleton.**

## I. INTRODUCTION

With the development of shape data acquisition techniques, more and more high-resolution shape data sets are acquired. Increasing demand for compact shape description has created the need to reduce this data to a description of more concise remnant.

Skeletonization is a promising and efficient process for data abstraction due to its linearity and simplicity. Skeletons are also ideal shape representations with efficient memory cost and data compression while highlighting topological structures. Skeletons provide meaningful cues for a number of applications in information visualization, pattern recognition, and geometric modeling. Shape decomposition is a practical process for simplifying the complex structure of an object into some simple components. The process begins to be applied to shape manipulation [1], shape matching [2], shape retrieval [3], and collision detection [4].

It was indicated that good shape decomposition will result in the extraction of a high quality skeleton [1], and that a high quality skeleton may produce a good decomposition [4]. But there is more work to do than to have done in specifying the relationship between shape decomposition and skeletonization, especially in 3D shapes [5]. The definition of shape decomposition of a volume is still not easy since it is hard to specify the cutting-surface.

This paper is an extension of our conference paper [6]. The main work of this paper is still to develop the relationship between shape decomposition and skeletonization for volume data and the way to decompose the volume. The application of volume decomposition to the extraction of hierarchical skeletons is emphasized and specified with technical details.

## II. RELATED WORK

The skeleton of a volume is defined as a 1D structure representing the shape and topology of the volume data [7], and it is a concise representation of shapes. Shape decomposition is a basic technique to understand its complex structure. Skeletons and shape components are typical shape features. Related work on skeletonization, shape decomposition and their applications are briefly discussed in this section.

### *2.1 Distance based volume skeletons*

Distance transformation of a volume converts voxels into different layers according to the silhouette surface or the shape tips. A skill conception was proposed by Zhou [8] to find a shortest path skeleton through the combination of DFS-distance (Distance From a Starting point) transformation with DFB-distance (Distance From Boundary) transformation.

Gagvani [9] adopted a thinness parameter to control the candidacy of a voxel on a skeleton based on boundary distance transformation.

Bitter [10] provided a global minimum cost path-searching algorithm with penalty distance through a heuristic combination of DFS-distance and DFB-distance. Sato [11] extended this algorithm into branched skeletons with an adaptive sphere. Bitter [12] further improved the penalty distance algorithm to correct mistakes in [10] and [11].

Wan [13] solved the corner-cutting problem of volume skeletonization by delivering a centered path rather than a shortest with exact Euclidean distance and minimum-spanning tree. But the influences of side-branches on the main branches are not considered.

### *2.2 Thinning based volume skeletons*

Parallel volume thinning is an efficient way to find the centerline thorough deleting outer voxels if this deletion does not break the original topology of the object.

Manuscript received on December 10, 2007
E-Mail: xpzhang@nlpr.ia.ac.cn

Ma [14] proposed a fully parallel, connectivity-preserving thinning algorithm to reduce computational costs. Expensive testing of simple points is avoided by matching the 26-neighborhood of the points with specific templates. However, the final centerline may be disconnected, and many spurious branches may be generated in skeleton. The connectivity of the skeleton is kept in [15] by adjusting a point deleted with its neighbor points in different iterations. A length parameter is also applied to remove the creation of spurious branches in [15]. However, these methods are ineffective if the length of a spurious branch is longer than that of a real one.

### *2.3 Shape decomposition and skeleton hierarchy*

Shape decomposition and skeleton hierarchy are significant to many applications, like shape analysis and understanding of 2D models [16, 17], 2D images [18], boundary represented models [1, 19], and volumes data [20].

Katz [1] presented an approach to progressively decompose the mesh into patches at regions of deep concavities. Lien [21] uses approximate convex decomposition (ACD) to partition the mesh into nearly convex components and its skeleton is then extracted from the convex hulls of these nearly convex components. Au presented in [22] a simple and robust skeleton extraction method based on mesh contraction. The 1D skeletal shape is obtained by performing geometry contraction using constrained Laplacian smoothing.

The concept of hierarchical curve-skeleton was proposed and by Cornea et al [20] and [23]. A family of hierarchical curve-skeletons is extracted robustly for different types of 3D objects (volumetric, polygonal and scattered point sets). The algorithm is based upon on computing a repulsive force field over a discretization of the 3D object. Topological characteristics of the resulting vector field, such as critical points and critical curves, are used to extract the curve-skeleton.

### *2.4 Applications of shape features*

Shape decomposition and skeleton hierarchy provide shape features of a volume for various applications. Skeletons are extracted for shape description in a new method of alternating thinning and a novel skeleton pruning routine in [24]. The extracted skeletons preserve topology and shape curvatures of the volumetric model. This is a simple and meaningful work, but the result cannot be used for segmentation, matching and recognition at this moment.

Hierarchical skeletons can be used for solid reconstruction. Local differential geometry of different kinds of points forming 3D symmetry sets was analyzed [25]. With this approach, the extracted skeletons can be used to reconstruct the original surface with full geometrical information. Linked with a real-time simulation of stroke therapy, Luboz et al [26] proposed a mechanism to segment and to reconstruct 3D human vasculature models with a good balance of smoothness, number of triangles, and distance error.

Skeletonization of a range image from real trees is a very new development. The range images are used to reconstruct the geometric model of the tree in [27] and [28]. The approach of [27] is for producing full polygonal models of range scanned trees through skeletonization of the trunk and main branches of the tree, and that of [28] is through 2D skeletonization and cylinder fitting.

To our knowledge, shape decomposition of volume data has not been fully considered yet. A novel method is proposed in this paper for hierarchical structure of the volume data and a hierarchical structure of compact skeleton, so that the final hierarchical skeleton represents the shape of the original volume well.

## III. ALGORITHM OVERVIEW

The object of the data is a tree-like volume in this paper. The main technical path of this paper is to decompose the volume data into sub-volumes, and hierarchical skeletons are obtained with the help of volume decomposition.

*Volume decomposition* is the separation of a complex branched volume around its nodes (also known as ramification points) into a series of simple sub-volumes, each of which is topologically equivalent to a single column. The term *hierarchical* in this paper does not mean a hierarchical process for skeleton extraction. It means a hierarchical decomposition of a volume into sub-volumes, where all skeletons are extracted in each sub-volume respectively, and they are connected to form a hierarchical structure. The object data in this approach is a tree-like volume, and it should have an evident root seed of the tree-like volume.

Three main concepts of this approach are: classification-skeleton for volume decomposition, and the decomposition applied for hierarchical connection of skeletons. *Classification-skeleton* shows the hierarchical structure of the object volume, so it is chosen as the key criterion for volume decomposition. *Hierarchical connection* means a connection of skeletons into a tree-like structure with the same topology of the original volume after decomposition. Classification-skeleton is the standard for this connection.

The classification-skeleton is extracted through distance transformation from the boundary, through distance transformation from a seed point, and central cluster graphing. With the help of classification-skeleton, volume decomposition is performed by a cross-section, called as decomposition cross-section, of each branch close to the nodes, so that the deviating affect of side branches is greatly reduced. This is shown in the ramification node of Fig. 8(b), Fig. 8(f), and Fig. 8(j), where the upper branch in Fig. 8(b), and that in Fig. 8(f), is all ”attracted” to the left branch, but that in Fig. 8 (j) is close to the center.

Finally, the hierarchical branched skeleton is organized by connecting all compact skeleton segments according to the classification-skeleton, so that the final hierarchical skeleton well represents the shape of the original volume.

This algorithm includes three main aspects: (1) point classification into a *hierarchical structure* through integer seed point distance transformation and boundary distance transformation, then classification-skeletons for branching structure; (2) *volume decomposition* at nodes through real-value distance transformation with the help of classification-skeletons; (3) skeletonization of each decomposed sub-volume through real-value distance transformation; and connection of hierarchical skeleton on reference of classification-skeletons. Fig. 1details the contents and the structure of this algorithm.

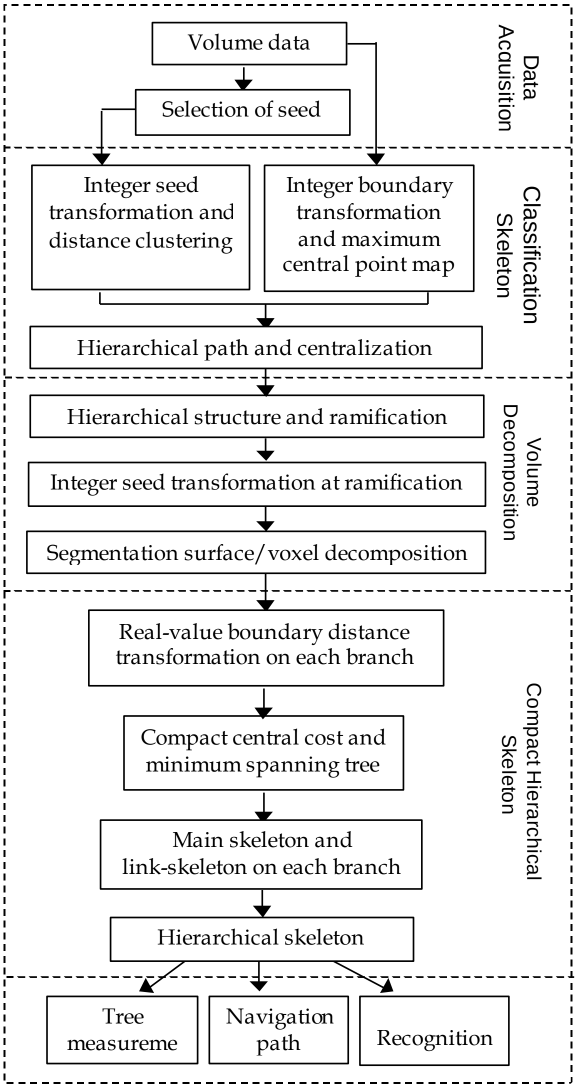

Volume data
Selection of seed
Data Acquisition
Integer seed transformation and distance clustering
Integer boundary transformation and maximum central point map
Classification Skeleton
Hierarchical path and centralization
Hierarchical structure and ramification
Integer seed transformation at ramification
Segmentation surface/voxel decomposition
Volume Decomposition
Real-value boundary distance transformation on each branch
Compact central cost and minimum spanning tree
Main skeleton and link-skeleton on each branch
Hierarchical skeleton
Compact Hierarchical Skeleton
Tree measureme
Navigation path
Recognition

Fig. 1. Overview of the algorithm.

## IV. VOXEL CLASSIFICATION

Classification of voxels in a volume data is in five aspects.

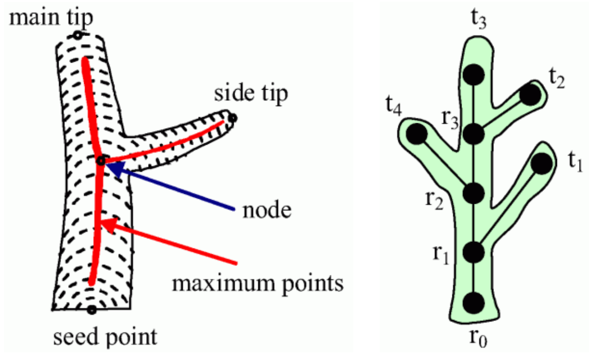
### 4.1 Distance transformation

Distance transformation is a voxel value propagation process from a sub-set $S$ to the whole set $V$. At the beginning, the value of each voxel in $V$ is set as the initial value, often 0 or 1. We take $W$ as the work set, and $S$ as the initial value of $W$. In each next step of the propagation, all voxels with $n$-neighborhood of any one voxel in $W$ are set as $m_n+m$ if its initial value is smaller than this, where $m$ is the distance of its $n$-neighborhood voxel in $W$, and $m_n$ is the scheme for distance transmission between two $n$-neighboring voxels. Often the evaluation rule is represented as $m_6$-$m_{18}$-$m_{26}$, where $m_6$, $m_{18}$, and $m_{26}$ represent the evaluation transmission to 6-neighborhoods, 18-neighborhoods, and 26-neighborhoods respectively. They are chosen differently for different transformation precision, such as 1-2-3, 4-5-6, and $1\text{-}\sqrt{2}\text{-}\sqrt{3}$.

$m_n$ can be an integer number, or more precisely, a real number. Integer number is good for concise classification, and real number is good for precise shape acquisition. In this paper, before volume decomposition, integer number is adapted, such as 1-2-3 or 3-4-5, so that distance can be used as a criterion to distinguish voxels according to branch structures. After volume decomposition, real number is adapted for a more precise skeleton.

Voxel points can be organized after integer distance transformation. A *cluster* is defined as a set of consecutive points with the same integer distance. *Classification skeleton* is a structure of skeletons linking the volume root to each branch tip, which shows the hierarchical structure of the object volume. *Volume decomposition* is the separation of a complex branched volume around its nodes into a series of simple sub-volumes, each of which is topologically equivalent to a single column. *Hierarchical connection* means a connection of skeletons into a branched structure with the same topology of the original volume.

Classification skeleton is, in this case, an application of integer distance transformation. It is used to find four useful aspects of information: tip points, nodes, hierarchical structure, and a rough central skeleton. The skeleton obtained this way is not well central and not 26-neighboring, but it can be used as a criterion for voxel classification due to its integer value.



(a) S-cluster (b) Branch graph

Fig. 2. Central clustering on root distance.

### 4.2 Central clustering

Integer seed point transformation is performed with the evaluation scheme 1-2-3 and 6-neighborhood propagation, where the seed point is the volume root specified by the user. The main purpose of this transformation is to find a tip point of each branch and all the nodes, so that decomposition could be properly performed at nodes.

An *S-cluster* is the connection of all the points having the same S-distance value, which is similar in part to sphere waves with the center as the seed point (Fig. 2(a)). S-cluster is often abbreviated as a *cluster* in this paper. All clusters constitute a graph according to their neighboring relation. The *positive* *direction* of this graph is defined as from the root to one tip.

A *tip cluster* is composed of points with maximum S-distance, which corresponds to a volume tip. A *node cluster* is the one from which the number of next neighboring clusters is at least one more than this level in the positive direction of the cluster graph. Seeing locally from a node cluster in the positive direction, the S-distance of the next neighboring clusters will increase by one. A node cluster corresponds to a volume node. The tip cluster and node cluster represent the topological structure of the cluster graph.

A *transition cluster* is the one that is not a seed cluster, not a tip cluster, and not a node cluster. A transition cluster has only one previous and only one following cluster. After omitting all transition clusters, the cluster graph will be a branched directed graph with each cluster as a node (Fig. 2(b)). This graph is referred to as a *cluster graph*.

### *4.3 Construction of the branch-link path*

A *branch-link path* is a sequence of points with two ends as the root and one tip of the volume. This way all branches are correctly connected with branch-link paths. This path is not necessarily the centerline of the volume, but it is the base for finding a central path of the volume in next sub-session.

A branch-link path is calculated as follows. All tip clusters are found first. For each tip cluster, the maximum B-distance point, or the barycentric point of all the maximum B-distance points if they are more than one, is chosen as the first current point. For each current point, we search for the point in its 6-neighborhood with least S-distance as the next point. This process is repeated until the root seed point is reached, so that a sequence of points is created, starting from a tip point to the root point in the order of decreasing S-distance.

The cluster containing the next point must be a neighboring cluster or the same cluster as the one containing this point, since the distance between two neighboring clusters is no more than 1. Therefore, a branch-link path is guaranteed to converge to the root point in its inverse direction.

Since 6-neighborhood is used, for any integer number between1 and the distance from tip point to the seed point, there will be a point in this series having the same distance as this number. Therefore, this sequence represents all the clusters between the tip cluster and the seed cluster.

After deleting all the representatives belonging to the same cluster, the connection of all representatives from the root to the tips will become a bundle graph or a divergent graph. This graph is called the *branch-link path*. This path is not the centerline of the volume, so it is not a central skeleton.

### *4.4 Centralization of the branch-link path*

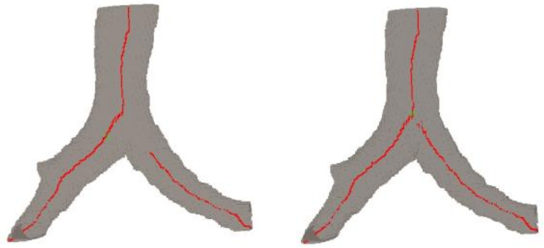
Centralization of the branch-link path means correcting this path or replacing each point in this path with another point in the same cluster closer to the cluster center, so that the entire path is close to the central points of this volume. Two concepts are useful for centralization. *Maximum central point set C* is made of all points, which are both in the maximum central point map and in this cluster also. *Boundary-distance maximum point set B* is the point set with maximum boundary distance.

The centralization begins from the tip point of each branch. The direction of centralization is decreasing with S-distance. If the maximum central point set of this cluster is empty, the barycenter of the boundary maximum point set is selected as the representative point of this cluster, and the boundary distance is recorded as the corresponding radius. If the maximum central point set and barycenter of boundary maximum point set are all not empty, the point in all these two sets with smallest distance to the immediately preceding point is chosen, and the boundary distance is also the corresponding radius.

With all other methods in [29], the point with the maximum B-distance is chosen, or the barycentric point of all the points with maximum B-distance if there was more than one maximum B-distance point is chosen. However, since the connection relationship of neighboring clusters is not well considered, this choice of central point may destroy the original topology of the volume. The green circle in Fig. 3(a) shows this wrong node location and wrong topology. We exploit the Maximum central point set *C* and Boundary-distance maximum point set *B* to centralize the branch-link path, so that the final path is correct in centrality and topology (Fig. 3(b)).

### *4.5 Main branch and branched structure*

Determination of the main branch is to find a primary and secondary order of all branches around each node, to select the principal path, and to unite all of them around the node according to their mutual distances. The determination of foliations and brotherhood depends on the measurement of the radius of each branch at the node. A simple way is to judge by the length, but it sometimes fails since the main branch is not always the longest, but more often the thickest.



(a) Improper topology (b) Proper topology

Fig. 3. Comparison of the cluster graph.

After the determination of all paths and all nodes, all filiations are recorded, and all these relationships constitute the branched structure. Considering the computation errors due to integer distance transformation and the influence of side branches on the parental branch, this hierarchical skeleton is not ideal, but is used to find the node location and volume decomposition at the node. It contains the information of hierarchical classification, centrality and connectivity, so it is called the *classification skeleton*.

## V. VOLUME DECOMPOSITION AROUND NODES

*Volume decomposition* means the separation of a branched volume into a hierarchy of sub-volumes, each of which is topologically equivalent to single columns. The crucial technique for decomposition is to decide the cutting surfaces to separate the volume into more parts.

The object volume is regarded as of clear hierarchical structure of parent-children relation at each node. After separation, the part with the main branch is called a *parent branch* or a *principal branch*, and the other is called a *subordinate branch*. All decomposition surfaces are cross sections almost orthogonal to the skeleton direction. The basic consideration in constructing a cutting surface is to find one surface on each branch, and to avoid any two cutting surfaces intersecting with each other. Additionally, all cutting surfaces are as close to the node as possible, so that as many voxels as possible will be assigned to the subordinate branch from the parent branch. The search for cutting surfaces starts from tips to the root.

The general topology of the volume and hierarchical connection relationship of each sub-volume can be deduced through the classification-skeleton obtained above.

For clarity, we only describe the construction algorithm for bifurcating nodes in this paper. We take any skeleton node location as the seed point and perform an integer seed point transformation for the entire volume. This is the first stage of transformation for volume decomposition. For each integer distance value, there must be a cross-section surface or some independent cross-section surfaces with this distance. The branched structure of the volume around this node is responsible for these possible independent surfaces with the same distance. We call them wave-pairs. Each surface is similar to some part of a sphere, and it is almost orthogonal to the classification-skeleton. Each wave pair is chosen to be as far as possible from the node position. Then a pair of surface transformations is performed backward from the wave- surface-patch pair surfaces to the node location until these pair surfaces intersect with each other. This is the second stage of volume decomposition. The reason the progress from farthest to nearest is to make the cutting surface as orthogonal to the skeleton as possible. Also, cutting surfaces too far from the node location will leave more voxels in the parent branch.

We set the cutting surface into the filial branch, and it is called the root-surface of the filial branch. The center of this root-surface is called as the *main source point* of the filial branch. The corresponding surface consists of voxels in the parent branch but 26-neighboring to some voxels of the root-surface of the filial branch. It is called the side-surface of the parent branch.

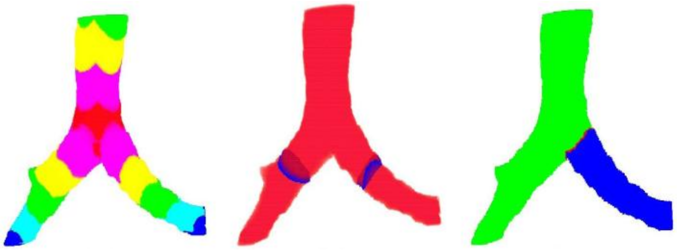

(a) (b) (c)

Fig. 4. Cross-section surfaces and volume decomposition. (a) the seed point transformation with the node as the seed point; (b) a pair of wave clusters are selected from the seed point transformation; (c) the cutting surface

Fig. 4 depicts an example of volume decomposition at a node. Fig. 4(a) is the seed point transformation of an airway at the node. Fig. 4(b) displays two separate cross section surfaces with the same seed distance to the node in two different branches. Fig. 4(c) illustrates the final result of the volume decomposition with green corresponding to the parent branch, blue corresponding to the filial branch, and red representing the cutting surface.

After decomposition, the shape of each sub-volume has several mouths, shown in Fig. 5. The barycenter voxel of the main tip mouth of this branch sub-volume is called the main tip. The barycentric voxel of the mouth corresponding to its filial volume is called a side tip. The barycentric voxel of the mouth cut from its parent branch or the root mouth is called a main source, which is the seed point for this sub-volume.

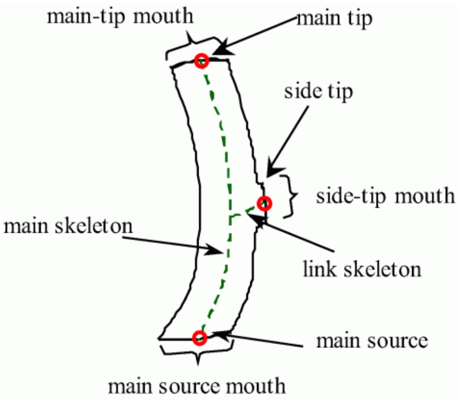


Fig. 5. Volume structure after decomposition.

## VI. COMPACT AND HIERARCHICAL SKELETONS

The results of volume decomposition are used to extract skeleton segments corresponding to each sub-volume *V*, to record the connection relation, and to connect them with more skeleton segments into the hierarchical topology equivalent to that of the volume.

Our method of hierarchical skeleton extraction takes three steps:(a) generation of a 3-D directed weighted graph from each sub-volume data, (b) using a minimum spanning tree (MST-tree) to get a skeleton segment for each sub-volume, and (c) connection of all skeleton segments. For each sub-volume, the main source point is chosen as the seed for point distance transformation. The evaluation scheme is real value $1\text{-}\sqrt{2}\ \text{-}\sqrt{3}$ with 26-neighborhood propagation, for better precision. Then, a *Compact Centrality Cost* Function, i.e. *C-Cost* function *C*(p), is constructed. Finally, a branched structure is constructed of minimum central cost connecting the main source point and the main tip point, and this topological structure is equivalent to that of the volume before volume decomposition.

### *5.1 Compact centrality cost*

The central line is the inverse feature of the boundary, so the distance to the boundary is chosen the basic opposing element of Centrality. In order to have centrality and compactness connection of the skeleton both be considered, *C*(p) is defined as composed of two parts in (1): boundary cost, *B-cost B*(p),and compactness cost, *Q-cost Q*(p).

$$C(\mathrm{p})=B(\mathrm{p})+Q(\mathrm{p}) \qquad (1)$$

where $B$(p) is defined as 1/b(p), and compactness cost $Q$(p)is defined as a scaled accumulated seed distance cost $T$(p)in (2).

$$Q(\mathrm{p})=k(V)\ \xi(V)\ T(\mathrm{p}) \qquad (2)$$

where $k(V)$, $\xi$, $T(p)$ are calculated in (3), (4) and (5) respectively.

$T(p)$ is defined as a recursive process starting from the main source point in Fig. 5, where the main source point is chosen as the seed point for transformation, and its $T(p)$ is set as 0 at this point. Then the $T(p)$ is evaluated recursively in (3), like a seed distance cost expansion from the seed point.

$$T(p)=\min\{S(q)+(1-M*b(p))E(p,q)\mid q\in N_n^*(p)\} \qquad (3)$$

where, $M=\min\{B(p)\mid p\in V\}$ is the minimum cost in current sub-volume $V$, so that $(1-M*b(p))$ is between 0 and 1. The effect of $b(p)$ is to make point p closer to the center. $E(p,q)$ is defined as $d(p,q)$ 0.75, $q$ $N(p)$, where $d(p,q)$ is the Euclidean distance. The effect of -0.75 is to make the influence of the distance between neighboring voxels lesser. For any two neighboring voxels p and q, $d(p,q)$ $[1, \sqrt{3}]$, so $E(p,q)$ is in the interval $[0.25,0.919]$.

$\xi$ $(V)$ in (2) is defined as the minimum absolute difference of any two B-distances of two voxel points in volume $V$ in (4).

$$\xi=\min\{|b(p)-b(q)|:|b(p)-b(q)|\geq v;\ p,q\in V\} \qquad (4)$$

Thus the Q-cost at any point must be smaller than any difference between any two points in the volume. Considering that real-value B-distance value could make $\xi(V)$ very small, a double precision real number is used to represent this cost. In another aspect, we perform a simple transformation on all of the B-costs, such that when the difference of any two B-costs is smaller than a small value $v$, they will be treated as equal. Therefore, the effect of $\xi(V)$ is that Q-cost will differentiate two points of the same B-cost, such that a voxel closer to the center has a bigger Q-cost. For voxels of different B-cost, Q-cost will have no effect, since they will have no impact on the two different B-costs. Therefore, the selected point will be closer to the center and closer to the seed point simultaneously, therefore centrality and compactness are both satisfied.

The coefficient $k(V)$ in (2)is a constant for sub-volume $V$ and it is defined as (5).

$$k=1/T\ ,\ T=\min\{T(p)\mid p\in V\} \qquad (5)$$

Therefore, the effect of centrality cost, C-cost, is the global cost of non-centrality and non-compactness.

### 5.2 Construction of a minimum spanning tree

After the determination of the compact centrality cost for each voxel, the volumetric dataset is converted into a 3D directed weighted graph. This graph includes all the links of any two 26-neighboring voxels. The connection directions of any two voxels are bi-directional, with different values for different directions. The size of this weight is the compact-centrality cost.

Similar to [13], all sub-volume data are transformed to 3D directed weighted graphs at first, and then a minimum- spanning tree are constructed for each sub-volume. This is expressed in the following procedure.

- Step1) Mark source point $S$ as the seed point for C-cost. Mark $S$, set its parent pointer *parent-link* to NULL.
- Step 2) Select the seed point as the beginning current point $T$.
- Step3) For each current point $T$, push each of its unmarked 26-neighborhood $B_i$ into a sorted heap in increasing order of centrality cost, so that the one with the smallest C-cost is at the top of the heap. If the centrality cost of the *parent-link* of the node $B_i$ is larger than the cost at $T$, set the *parent-link* of $B_i$ to $T$.
- Step 4) Pop the head of the heap, mark it, and set it as the current node.
- Step 5) Repeat Step 3) and Step 4) until the heap is empty.

This is a little similar to, but is an improvement on the algorithm described by Wan [13], since node case is not considered. Also, it is more concise, compact, and more robust thanks to formula (3), so that the skeleton always stays away from the boundary and the tendency to go roundabout is avoided.

### 5.3 Skeleton extraction and hierarchical connection

Skeleton extraction is in two steps. The main tip and side tips are calculated at first for the minimum-spanning tree of the current sub-volume (Fig. 5). The skeleton segments with two ends of these two tips are called the *main-skeleton* and *side-skeleton*, respectively. The main-skeleton connects the main source and the main tip, whereas the side skeleton connects one side tip to the main-skeleton, so that the centrality of the skeleton, hierarchy, and topological connection are maintained.

Then the main-skeleton is extracted by tracing along the MST back from the main tip point until the main source is met. The radius values associated with the skeleton segment are the distances of the path points to its corresponding boundary points. Skeleton extraction is finished after recording this path along with the associated radii. For the link-skeleton, the process is similar, except that the beginning point is chosen as the side tip point, until one voxel in the main-skeleton is reached. From the definition of compact centrality cost, the link-skeleton will converge to the main-skeleton, since the latter is the skeleton of this sub-volume.

After the extraction of skeleton segments, main skeletons and link skeletons, they will be connected together according to their original topological relations, including all paths from each tip to root passing all nodes. This is where the hierarchical skeleton can help, with its excellent centrality and classification.

If the link-skeleton is not used and only the main skeleton is used as the feature, these skeletons can be used for pattern recognition of 3D objects.

## VII. EXPERIMENTS

We use branched volume data, tree models and airway models, to test our algorithm. These data sets have been tested on a P4 PC with 2.4GHz CPU and 1G RAM.

TABLE 1 lists the details about the data, including the number voxels in the volume, the number voxels in the skeleton, and the number of branches. TABLE 2 shows the seconds spent on each step of the algorithm for each test data set. The following abbreviations are adopted in these two tables: Dcmp. (Decomposition), Extr. (Extraction), Skel. (Skeleton), and Vol. (Volume).

TABLE 1: VOLUME AND SKELETON SPECIFICATIONS

| Name | Vol. voxel No | Skel. Voxel No | Branch No. |
|---|---|---|---|
| Airway1 | 552595 | 408 | 3 |
| Airway2 | 581539 | 1229 | 14 |
| Larch | 29657 | 910 | 12 |
| Willow | 49883 | 1622 | 18 |

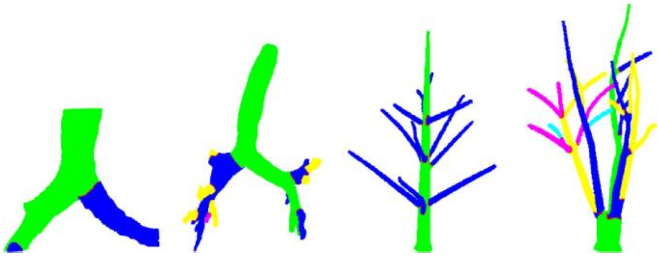


Fig. 6. Hierarchical volume decomposition.

TABLE 2: TIME COST ON EACH OPERATION IN SECONDS

| Vol. Name | Cluster graph | Vol. Dcmp. | Skel. Extr. |
|---|---|---|---|
| Airway1 | 20.08 | 114.09 | 377.78 |
| Airway2 | 51.91 | 460.67 | 572.38 |
| Larch | 3.48 | 148.39 | 31.09 |
| Willow | 10.75 | 352.66 | 74.02 |

Fig. 6 shows the results of volume decomposition, where green, blue, yellow, pink and cyan represent the levels of branches in increasing order. Objects here are two human airways, above year-old larch tree and a four year-old willow tree. It can be seen that each sub-volume has a single-column shape and their borders are kept well. Fig. 8 also demonstrates that our volume decomposition technique is valid for multi-furcations.

The results of extracted skeletons by Ma's algorithm [14], Wan's approach [13], and our proposed algorithm are compared in Fig. 7. The results of Ma's are shown in the first row, the results of Wan's are illustrated in the second row, and the results of our algorithm are listed in the third. The first column displays the result of the three algorithms at the node of an airway. It can be seen that spurious branches are pruned in our algorithm while they still exist in the other two algorithms. The second column illustrates the results of three algorithms in a simple node of a willow tree. Here we find that our algorithm can be used to eliminate the influence of deviation, but it still exists in the other two approaches. The third column is the skeleton of a willow branch. Ma's method generates some spurious branches in the node; Wan's algorithm generates a "roundabout" shape in the skeleton, but all these mistakes are removed in our algorithm. The fourth column illustrates the skeleton in a multi-node of the willow tree. Therefore, our algorithm keeps the skeleton in the center while the skeletons generated by the other two algorithms do not follow the medial axis.

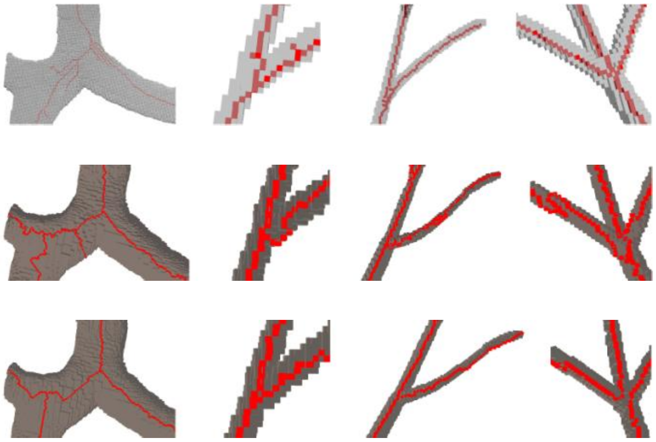


Fig. 7. Effect comparison among three approaches on three samples.

Overlaying of the final skeletons over the objects are illustrated in Fig. 8 of all four models from Fig. 6, where they are viewed from three different angles. A special visualization method is used to show non-boundary voxels with a nontransparent single color display, and to show boundary voxels with a semi-transparent and lighting display. The incoming boundary voxels are set semi-transparent, and inverse voxels are set to be nontransparent. The results demonstrate that skeletons extracted with our algorithm are more centered and more compact , and the deviation is eliminated in the node part through compact centrality cost when there exist many candidate points with the same distance from the boundary.

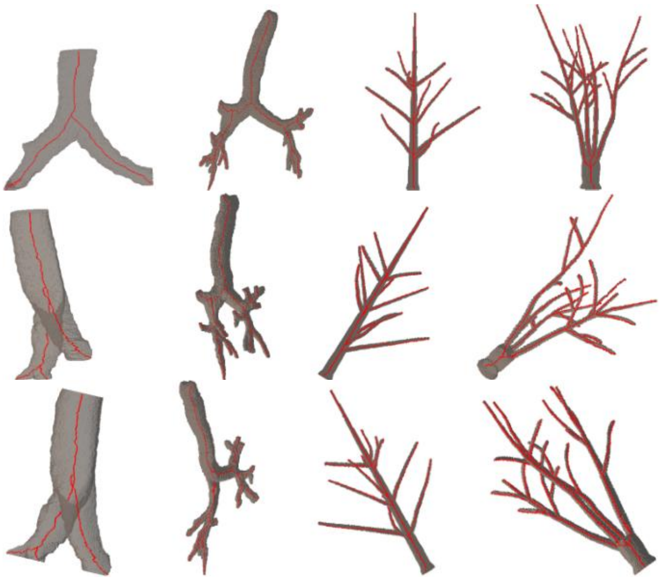


Fig. 8. Overlaying skeletons on 3D models.

## VIII. CONCLUSION AND FUTURE WORK

We proposed a volume decomposition and hierarchical skeletonization algorithm for branched objects. The extracted skeletons are highly centered. One important step of our work is the help of volume decomposition to hierarchical skeletons, so that the final extracted skeletons keep the structure of the volume data well.

A direct application of this approach is a hierarchical guide for internal navigation inside human organs [30]. Such skeletons are also efficient to represent the structure and the geometric shape of real trees for many plant modeling and visualization applications [31]. Besides, the method, and more precisely the hierarchical skeleton, can contribute to fast tree measurements and geometry reconstructions from volume data, both known to be difficult [32].

This work is limited to branched volumes rather than a general volume, failing in case of loop shapes. In the future, this work could be extended to more general shapes with complex topology. To do so, volume decomposition around a node according different explanations should be revisited, understanding that the simple equidistant surface criteria at node location is for now too restrictive.

## ACKNOWLEDGEMENT

This work is supported in part by National Natural Science Foundation of China with projects No. 60672148 and No. 60872120; in part by the National High Technology Development 863 Program of China under Grant No. 2008AA01Z301, No. 2006AA01Z301, and 2008AA10Z218; in part by the French National Research Agency within project NATSIM ANR-05-MMSA-45; and in part by the Project Arcus 2006 Languedoc-Roussillon/Chine.

## REFERENCES

[1] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Computer Graphics (Proc. of SIGGRAPH 2003)* vol.*22*, no. 3, 954–961, 2003.

[2] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz and D. Dobkin. Modeling by example. *ACM Trans. Graph.* vol.23, no. 3, 652–663, 2004.

[3] T. Tung and F. Schmitt. Augmented reeb graphs for content-based retrieval of 3d mesh models. *Shape Modeling Applications, 2004. Proceedings*, pp.157–166, 2004.

[4] X. Li, T. W. Woon, T. S. Tan and Z. Huang. Decomposing polygon meshes for interactive applications. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, pp.35–42, 2001.

[5] J. -M. Lien and N. M. Amato. Approximate convex de-composition of polyhedra. In *Symposium on Solid and Physical Modeling*, pp.121–131, 2007.

[6] Zhang, X., Liu, J., Li, Z. And Jaeger, M. Volume decomposition and hierarchical skeletonization. In *VRCAI '08: Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Indus-try*, ACM, New York, NY, USA, 2008.

[7] G. J. Brostow, I. Essa, D. Steedly and V. Kwatra. Novel skeletal representation for articulated creatures. In *Proceedings of the European Conference on Computer Vision (ECCV04)*, vol. *III*, pp.66-78, 2004.

[8] Y. Zhou and A. W. Toga. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and ComputerGraphics5*, no.3, pp. 196–209, 1999.

[9] N. Gagvani, and D. Silver. Parameter controlled volume thinning. *Graphical Models and Image Processing 61*, no.3, pp.149–164,1999.

[10] I. Bitter, M. Sato, M. Bender, K. Mcdonnell, A. Kaufman and M. Wan. Ceasar, A smooth, accurate and robust centerline extraction algorithm. In *Proc. Visualization 2000*, pp. 45–52, 2000.

[11] M. Sato, I. Bitter, M. Bender and A. Kaufman. Teasar. Tree-structure extraction algorithm for accurate and ro-bustskeletons.In*Proc.8thPacific conf. Computer Graphics and Applications*, pp. 281–289, 2000.

[12] I. Bitter, A. Kaufman and M. Sato. Penalized-distance volumetric skeleton algorithm. *IEEE Transactions on Visualization and ComputerGraphics7*, no.3, pp. 195–206, 2001.

[13] M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter and A. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Transactions on Medical Imaging 21*, 12, pp. 1450–1460, 2002.

[14] C. M. Ma and M. Sonka. A fully parallel 3D thinning algorithm and its applications. *Computer Vision and ImageUn-derstanding64*, no.3, pp. 420–433, 1996.

[15] J. Liu, X. Zhang and F. Blaise. Distance contained skeleton for virtual endoscopy. In *IEEE International Symp. Biomedical Imaging Macro to Nano (ISBI)*, pp. 261–264, 2004.

[16] H. Rom and G. Medioni. Hierarchical decomposition and axial shape description. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, no.10, pp. 973–981 1993.

[17] M. Simmons and C. H. Sequin. 2d shape decomposition and the automatic generation of hierarchical representations. *International Journal of Shape Modeling*, no.4, pp. 63–78, 1998.

[18] A. Telea, C. Sminchisescu and S. Dickinson. Optimal inference for hierarchical skeleton abstraction. In *ICPR (4)*, pp. 19–22, 2004.

[19] J. Lien and N. Amato. Simultaneous shape decomposition and skeletonization using approximate convex decomposition. In *Technical Report, TR05-004, Parasol Laboratory, Department of Computer Science, Texas A M University*, pp. 44–47, 2005.

[20] N. Cornea, D. Silver, X. Yuan and R. Balasubrama-Nian. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer21*, no.11, pp. 945–955, 2005.

[21] J. -M. Lien and N. M. Amato. Approximate convex decomposition of polyhedra. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Posters*, ACM Press, New York, NY, USA, 2. 2004.

[22] O. K. -C. Au, C. -L. Tai, H. -K. Chu, D. Cohen-Or and T. -Y. Lee. Skeleton extraction by mesh contraction. *ACM Trans.Graph.27*, 3, pp. 1–10. 2008.

[23] N. D. Cornea and P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13*, no.3, pp. 530–548, 2007.

[24] T. Ju, M. L. Baker and W. Chiu. Computing a family of skeletons of volumetric models for shape description. *Comput. Aided Des,* vol.39, no.5, pp. 352–360, 2007.

[25] P. Giblin and B. Kimia. A formal classification of 3d medial axis points and their local geometry. *IEEE Trans. Pattern Anal. Mach. Intell,* vol. 26, no.2, pp. 238–251, 2004.

[26] V. Luboz, X. Wu, K. Krissian, C. -F. Westin, R. Kikinis, S. Cotin and S. and dawson. A segmentation and reconstruction technique for 3d vascular structures, In *Proceedings of medical image computing and computer assisted intervention (MICCAI)*, 2005.

[27] H. Xu, N. Gossett and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.26*, no.4, pp. 19. 2007.

[28] C. Cheng, X. Zhang and B. Chen. Simple reconstruction of tree branches from a single range image. *Journal of Computer Science andTechnology22*, no.6, pp. 846–858, 2007.

[29] A. Shahrokni, R. Zoroofi and H. Soltanian-Zadeh. Fast skeletonization algorithm for 3d elongated objects, In *Proc. SPIE 4322*, 323–330. 2001.

[30] D.Bartz. Virtual Endoscopy in Research and Clinical Practice, *Computer Graphics Forum24*, no.1, pp. 111–126. 2005.

[31] I. Shlyakhter, M. Rozenoer, J. Dorsey and S. Teller. Reconstructing 3d tree models from instrumented photographs, *IEEE Comput. Graph. Appl*, vol.21, no.3, pp. 53–61, 2001.

[32] A. Kanitsar, T. Theußl, L. Mroz, M. Srámek, A. V. Bartrolí, B. Csébfalvi, J. Hladuvka, D. Fleischmann, M. Knapp, R. Wegenkittl, P. Felkel, S. Guthe, W. Purgathofer and E. Gröller. Christmas tree case study: computed tomography as a tool for mastering complex real world objects with applications in computer graphics, In *Proceedings of the conference on Visualization '02*, pp. 489–492, 2002.

**Dr Xiaopeng Zhang** received his MS degree in Mathematics from Northwest University in Xi'an in 1987, and his PhD degree in computer graphics from the Institute of Software, Chinese Academic of Sciences (CAS), in Beijing in 1999. He is an associate professor in the Sino-French Laboratory LIAMA and National Laboratory of Pattern Recognition at CAS Institute of Automation. His main interest is computer graphics and pattern recognition.

Dr. Zhang was invited as a foreign specialist for forest visualization in INRIA, the French National Institute for Research in Computer Science and Control. He is also an associate professor of Beijing Graduate School of CAS, and the course he is teaching there is *computer graphics and application*. He received the National Scientific and Technological Progress Prize (Second Class) in 2004. He has more than 70 publications and he advised 7 PhD students.

He received the National Scientific and Technological Progress Prize (Second Class) in 2004. He is presiding some projects on computer graphics, pattern recognition and virtual reality from National Natural Science Foundation and National High-Tech Research and Development 863 Plan of China. He is the conference co-chair of PMA06: The Second International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications. And he is also the program co-chair of Edutainment 2008, the third International Conference of E-Learning and Games.

**Mr Jianfei Liu** received his BS degree in electrical engineering from University of science and technology, Beijing, China, in 2001. At the same year, he joined LIAMA-NLPR, Institute of Automation, Chinese Academy of Sciences to pursue his master degree in Intelligent System. In 2005, he joined University of North Carolina at Charlotte to pursue his Ph.D degree in computer science. His research interests include computer vision, medical image analysis, pattern recognition, computer graphics, and visualization. He got NIH (National Institute of Health) summer fellowship twice in 2006 and 2007.

**Dr Marc Jaeger** is a computer graphics scientist (IEEE member) interested in natural phenomena. Graduate from University Louis Pasteur of Strasbourg (PhD, 1988) he specialized on plant growth simulation and visualisation at CIRAD Amap laboratory. From 1990 to 1998, he led a small team working on volume imaging. He joined then the GreenLab project (LIAMA, China), dedicated to plant functional structural modelling, simulation and visualisation. Back to France within DigiPlante team (associate project of INRIA, CIRAD and Ecole Central of Paris), his current research topics concern mainly landscape functional simulation and visualization.

**Dr Zili Li**, born in 1965, Ph.D degree gained in 2002 in Huazhong University of Science & Technology, Associate professor in Department of Electronics & Information Engineering in Huazhong University of Science & Technology. He is member of China Society of Image and Graphics (CSIG), commissioner of Virtual Reality Specialty Committee in CSIG, commissioner of Digital Art Specialty committee in CSIG in CSIG, member of China Computer Federation (CCF). His interest of research is computer graphics, virtual reality & 3D Game Development and multimedia.