Voxel-based shape decomposition for feature-preserving 3D thumbnail creation

Pei-Ying Chiang*, C.-C. Jay Kuo

Ming Hsieh Department of Electrical Engineering and Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2546, United States

A R T I C L E  I N F O
Article history:
Received 28 December 2010
Accepted 12 July 2011
Available online 23 July 2011

Keywords:
Voxel-based shape decomposition
Volumetric shape representation
Primitive approximation
Skeleton extraction
Skeletonization
Skeleton refinement
Feature-preserving 3D thumbnail
Skeleton-guided shape decomposition

A B S T R A C T
In this work, we study the problem of voxel-based shape decomposition and simplification to create simplified 3D models, called 3D thumbnails, which preserve the salient features of their original models to facilitate users’ interactive browsing. The proposed method decomposes a 3D model into multiple parts and simplifies each decomposed part individually. In this process, a 3D model is first converted into a voxel-based shape representation and a rough skeleton is extracted using the thinning operation. Then, a skeleton refinement algorithm is proposed to fine-tune the thinned skeleton and decompose the skeleton into multiple groups. The remaining processing steps include: (1) taking body measurements for each part with PCA transformation and (2) creating the 3D thumbnail by primitive approximation. It is shown by experimental results that the proposed voxel-based scheme outperforms the mesh-based scheme in the sense that the resultant 3D thumbnail can preserve more features when it is greatly simplified.

1. Introduction
The number of 3D models grows rapidly due to their popularity in several industrial sectors. There is an increasing demand on effective management of 3D models (e.g., archival, indexing and retrieval of 3D models in a large repository) since users can benefit from effective reuse of existing models. The conventional 3D search engine displays static 2D thumbnails on the search page for the browsing purpose. However, a 2D thumbnail may not represent a 3D model well. Some researchers attempted to automate the process of 2D snapshot taking of a 3D object by selecting the best viewing angle [1]. However, it is difficult to find an ideal angle selection rule for generic objects. Even if one can capture the best angle of a 3D object by a 2D thumbnail, there are still features that cannot be seen from this angle. To overcome the shortcoming of 2D thumbnails, a feature-preserving 3D thumbnail creation system was presented in [2] with an objective that users can browse multiple 3D models by viewing their 3D thumbnails interactively at a lower cost (e.g., smaller memory space and faster rendering speed).

Although there exist quite a few mesh simplification techniques, most of them are not designed to preserve the salient features of a given model. For example, the limbs and the body of a human model can meld together when it is extremely simplified. To preserve the shape features of a 3D model, a mesh-based shape decomposition scheme, which was initially proposed in [3], was adopted in [2] to decompose a model into multiple meaningful parts and, then, each part was simplified individually. However, the mesh-based shape decomposition scheme has its limitations. For example, although it can decompose the protruding parts from the main body successfully, other meaningful parts (e.g., the head and the neck) may not be properly separated. Besides, the surface of the decomposed mesh may become fragmented. The surface radius measured using the fragmented mesh may not be accurate enough to yield a good representative thumbnail.

Here, instead of finding patches to resolve problems arising from mesh decomposition and simplification, a volumetric shape decomposition scheme is presented to overcome the above-mentioned difficulties. To be specific, we propose a voxel-based shape decomposition scheme in this work. With this scheme, a 3D model is first converted into a 3D voxel representation, and the skeleton of the voxelized model is extracted by the thinning operation. Then, a skeleton refinement process is used to fine-tune the thinned skeleton, the refined skeleton is decomposed into multiple groups, and the voxelized model is decomposed into meaningful parts accordingly. The other operations for the 3D thumbnail generation remain the same as that described in [2]. They include: taking body measurements for each part with the PCA transformation, and generating the final thumbnail by approximating each part with fitting primitives.

The voxel-based decomposition method has a challenge of its own. That is, the skeleton obtained by the thinning operation (or other skeletonization operations) may contain artifacts to result in a messy 3D thumbnail. For example, the skeleton may contain noisy or redundant skeleton-voxels (SVs), the skeleton may be jagged and skewed, or the skeleton may not represent the principal axis of a shape correctly. These artifacts often affect the shape decomposition
process and lead to an inaccurate 3D thumbnail. We show that the skeleton refinement process plays an important role in reducing artifacts of skeletons yet preserving sharp features and will discuss this process in detail. The main contribution of this research lies in the development of a robust voxel-based shape decomposition method.

The rest of this paper is organized as follows. A review on related previous work and an overview of the proposed system are given in Section 2. The skeleton refinement process that links, groups and fine-tunes the skeleton is discussed in Section 3. The voxel-based shape decomposition scheme is described in Section 4. Experimental results and subjective visual tests are detailed in Section 5. Finally, concluding remarks and future research directions are given in Section 6.

2. Background review and system overview

Our work on 3D thumbnail representation is highly related to mesh simplification [4][5]. Besides, mesh decomposition is needed to preserve salient features of 3D objects. In this section, we will first review previous work on mesh simplification and mesh decomposition in Section 2.1 and Section 2.2, respectively. Then, we will give an overview of the proposed voxel-based shape decomposition system in Section 2.3.

2.1. Mesh simplification

Since a complex 3D model contains millions of polygons and requires a large amount of memory and time to process, it can degrade the runtime performance significantly. Many applications may require (or can only support) a low resolution model rather than its original model of full resolution. For example, when an object is far from the camera, its rendering time can be shortened if its low-resolution version is used. Previous work on mesh simplification can be categorized into surface-based, voxel-based and hybrid approaches.

Garland et al. [6] developed a surface-based simplification algorithm to produce high quality approximations of polygonal models. This algorithm used iterative contractions of vertex pairs to simplify meshes while maintaining the minimum surface approximation error using a quadric metric. Cohen et al. [7] proposed an error-driven optimization algorithm for geometric approximation of surfaces. They used an idea similar to the Lloyd algorithm, which reduced the distortion error through repeated clustering of faces into best-fitting regions. Their approach did not require parameterization or local estimations of differential quantities. While only approximating planes were considered in [7], Wu et al. [8] extended the optimization technique by allowing different primitives such as spheres, cylinders and more complex rolling-ball blend patches to represent the geometric proxy of a surface. They segmented a mesh model into characteristic patches and constructed a geometric proxy for each patch. Marco et al. [9] proposed a hierarchical face-clustering algorithm for triangle meshes with various fitting primitives. All the above-mentioned schemes used an iterative approach to produce a fine-to-coarse approximation. The mesh was simplified incrementally during its approximation process.

He et al. [10] proposed a voxel-based mesh simplification method. It used sampling and low-pass filtering to transform an object into a multi-resolution volume buffer, and then applied the marching cubes algorithm [11] for building a multi-resolution triangle-mesh surface hierarchy. Nooruddin et al. [12] adopted a hybrid approach that integrated the voxel-based and the surface-based simplification methods. They converted a polygonal model to a volumetric representation, and then repaired and simplified it with the 3D morphological operators. Visually unimportant features, such as tubes and holes, can be eliminated by the 3D morphological operators. The volumetric representation was then converted back to polygons, and a topology preserving polygon simplification technique was used to produce the final model.

2.2. Mesh decomposition

To manipulate meshes efficiently in different applications such as modeling, compression, collision detection and shape retrieval, mesh decomposition algorithms were developed to decompose a mesh into small parts based on certain properties. Mesh decomposition techniques were surveyed by Shamir [13] and Attene et al. [14]. Generally speaking, mesh decomposition can be achieved by considering either semantic features or geometric primitives.

Mesh decomposition methods relying on semantic features take human perception and psychological cognition into account. Rules arising from psychology (e.g., the minimal rule, separate theory and visual salience) have been used to analyze components of 3D objects. Li et al. [15] proposed a collision detection algorithm, which employed edge contraction for skeleton extraction and adopted the space sweeping technique [16] along the skeleton path to segment a polygonal mesh into meaningful components. The idea of our shape decomposition algorithm is similar to theirs, yet a different method for volumetric shape decomposition is applied. Katz et al. [17] introduced a fuzzy cutting method to avoid over-segmentation and jagged boundaries between sub-objects. An elevation based on multi-dimensional scaling, prominent feature point representation and core extraction was conducted. Liu et al. [18] developed a part-aware surface metric by extending the work of [17] for shape retrieval. Liu et al. [3] proposed a decomposition scheme based on cognitive psychology where evaluation was made with respect to protrusion, boundary strength and the relative size of a part. We present an enhanced algorithm which can decompose shapes more precisely in this work.

Mesh decomposition methods relying on geometric primitives decompose a mesh based on its geometric properties (e.g., planarity, curvature, etc.) and yield partitioning surface patches. Examples include [7–9] as given in Section 2.1. Another segmentation scheme was described in [19], which employed hierarchical clustering to merge points to larger regions using slippage similarity scores. Local points and regions with similar slipping motion are clustered together to form segments. In practice, it is possible to develop a hybrid scheme that integrates both semantic features and geometric primitives.

2.3. Overview of proposed system

Most of previous mesh simplification methods were not much concerned with preserving significant parts of a model. For example, a greatly simplified human model tends to meld limbs and the body together. How to address this issue will be the main focus of this work. The block diagram of a feature-preserving 3D thumbnail creation system introduced in [2] is depicted in Fig. 1. The main difference between our current work and [2] is that a voxel-based shape decomposition scheme is employed here while a surface-based shape decomposition scheme was adopted in [2].

Our main objective is to improve the performance of shape decomposition by introducing three new blocks highlighted in orange in Fig. 1. They are explained below.

- **Voxelization and thinning**
  A polygonal model is first rasterized into a binary 3D voxel grid. Then, a coarse skeleton is extracted from the volumetric model using a thinning algorithm [20]. These tasks can be accomplished with tools in [21 and 22], respectively. In the following, we use object-voxels and skeleton-voxels to denote the volumetric model and the thinned skeleton, respectively, as illustrated in Fig. 2(a).
Skeleton refinement

The skeleton refinement process links and groups discrete SVs obtained with the thinning operation. Since the thinned skeleton often contains defects that affect the shape decomposition result, a skeleton refinement process is developed to enhance the extracted skeleton.

Shape decomposition

A shape decomposition method is used to decompose the shape into meaningful parts according to the grouping of skeletons. By assigning object-voxels to the group associated with their nearest skeleton, a shape can be decomposed roughly. A refined skeleton decomposition method is developed to re-group SVs more precisely so that the shape can be decomposed more accurately as well.

Once a shape is decomposed into multiple parts, each part will be handled individually. As shown in Fig. 1, the remaining processes include: PCA transformation, body measurement, and primitive approximation. The PCA transformation is applied to each part individually for pose normalization. The body measurement (i.e., the radius of the surrounding surface along the skeleton) is taken for each part. Then, a primitive approximation method approximates each part with fitting primitives based on body measurement results to yield the 3D thumbnail. Moreover, the shape descriptor and the thumbnail descriptor describing the simplified shape of the original model are generated off-line and stored in the database. Consequently, the thumbnail can be downloaded and rendered efficiently on-line. For more details, we refer to [2].

In the following, we will give in-depth treatment on two new blocks in Fig. 1. They are: (1) the skeleton refinement process and (2) the shape decomposition process, which will be examined carefully in Sections 3 and 4, respectively.

3. Skeleton refinement

After the voxelization and thinning process, thinned SVs consist of a set of discrete voxels that are to be linked, grouped and fine-tuned to represent a meaningful structure of the model. This process, called “skeleton refinement”, is discussed in this section.

3.1. Skeleton Voxel (SV) classification and linking

We classify SVs into the following five categories.

1. End-SV: The end point of a skeleton which has only one neighbor;
2. Joint-SV: The joint of a skeleton which has more than two neighbors;
3. Peak-SV: The turning point of a skeleton, which has two neighbors and is a local peak;
4. Orphan-SV: An isolated voxel which has no neighbors;
5. Normal-SV: A voxel which has two neighbors and is not a local peak.

The classification tasks are often easy, since most of them can be accomplished by counting the neighbors of a voxel. We consider
a $3 \times 3 \times 3$ grid surrounding a central voxel and view the surrounding 26 voxels as its neighbor as illustrated in Fig. 3. The classification of Peak-SV demands some extra effort, which will be performed after SVs are linked since a local peak is more difficult to extract based on discrete voxels. Examples of End-SV, Joint-SV and Normal-SV are shown in Fig. 2(b).

In the linking process, we link neighboring SVs and divide them into groups according to the joint location. In practice, we create each group with either one of End-SVs or one of Joint-SVs, and continuously link unvisited adjacent voxels to this group until another End-SV or Joint-SV is met. As a result, each skeleton group will consist of linked SVs whose two ends are either an End-SV or a Joint-SV as shown in Fig. 2(c). However, a thinned skeleton may have defects to affect the decomposition result. This will be handled by some post-processing techniques as described in Section 3.2.

The turning point is used to separate parts that can be bent such as the separation of the lower arm and the upper arm. After the linking process, a Peak-SV can be extracted by analyzing the curvature. However, since the thinned skeleton may be jagged (i.e., containing many unwanted local peaks), extracting turning points by detecting local peaks may not work properly. To address this issue, a hybrid scheme that integrates two commonly used feature extraction methods is developed. The turning point is first located using the global distance and then adjusted based on the local curvature.

The concept of the global distance is illustrated in Fig. 4. For each group of SVs, we first draw a straight line from one end to the other end. Then, we compute the perpendicular distance from each point along the curve to this straight line. At each iteration, a voxel whose distance to the line is greater than a distance threshold and whose angle (formed by both ends with itself in the middle) is smaller than an angle threshold will be selected as a candidate. The candidate voxel whose distance is the greatest is then picked as the turning point, and the curve is divided into two sub-curves at this voxel. The process is applied recursively in these two sub-curves until there is no voxel that qualifies as a candidate.

However, the turning point extracted using the global distance alone may be located at a flat region. The angle constraint is then used to adjust the extracted turning point by examining the local curvature at each iterative. For each turning point extracted using the global distance criterion, we examine the curvature in its local neighborhood. The point whose curvature is the largest in this neighborhood will be reclassified as the turning point instead of the original one. Fig. 2(d) shows an example of extracted turning points. Some meaningful parts such as the head and the neck of a 3D object can be separated using these turning points. However, it is worthwhile to point out that we do not divide the skeleton with turning points in the beginning since it will complicate the following processes.

3.2. Skeleton post-processing techniques

If obtained SVs are clean and smooth, a skeleton can be linked and grouped into several meaningful parts easily. If SVs obtained by the thinning operation have artifacts, the skeleton-based decomposition result will become messy. In this subsection, post-processing techniques are proposed to handle three types of artifacts: (1) clustered SVs, (2) jagged and skewed skeletons, and (3) Sub-branches.

3.2.1. Clustered SVs

Ideally, a thinned skeleton should have the width of one voxel while retaining the connectivity of the original shape. However, the extracted skeleton may contain clusters of adjacent voxels as shown in Fig. 5(a) and Fig. 6(a). These clustered SVs as represented by yellow cubes are classified as Joint-SVs since they all have more than two neighbors.

Connecting two Joint-SVs that are adjacent (or very close) to each other using the linking process may result in a redundant group or an incorrect tiny loop as shown in Fig. 5(c) and Fig. 6(a). These are caused by the ambiguous relationship of thinned SVs. When two SVs are adjacent to each other, they may not be linked definitely. For Joint-SVs that have more than two neighbors, it is ambiguous to choose the proper adjacent voxel for linking. Wang et al. encountered the same problem in [23] and proposed to check whether a connection will cause a 3-edge cycle before it is added to the skeleton. However, it was not discussed which edge is to be discarded among a 3-edge cycle. Furthermore, they did not handle some other cases that may lead to tiny groups or loops as well. Here, we propose the use of two filters (i.e., the re-classifying filter and the replacing filter) to solve this linking problem without deleting any SV.

For each cluster of adjacent joints, the reclassifying filter chooses one of them as the representative and reclassify the rest as redundant joints. The representative is the one that has the largest number of adjacent joints, where a tie can be resolved by choosing the one that is closest to the center of this cluster. We do not allow a redundant joint to directly link to another redundant joint that is associated to the same representative joint. In other words, the path between each pair of redundant joints in the same cluster will be discarded. The effect of applying the reclassifying filter is shown in Fig. 5.

The replacing filter aims to solve the problem when two joints are too close to each other such as the example shown in Fig. 6(b). To address it, the replacing filter iteratively searches a pair of joints that are connected by the shortest path with their distance less than a certain threshold (e.g., 5 voxels) and, then, selects the midpoint of this path as a new representative to replace the original two joints. The shortest path is split into two halves at the new representative joint and merged into their adjacent groups, respectively. An skeleton post-processing example using the reclassifying filter and the replacing filter in cascade is shown in Fig. 6.

3.2.2. Jagged and skewed skeletons

The thinned skeleton is often jagged and skewed. It is desirable to smoothen it while preserving key sharp features in the process.
To avoid removing sharp features, we first extract important features using the global distance criterion as described before. The extracted features will not be moved in the following smoothing operation. Afterwards, an iterative curve smoothing operation is applied. At each iteration, we examine every voxel along the skeleton and adjust its location according to the distribution of SVs within a local window. The window size will increase at the next run until it reaches the maximum size (e.g., the path length). For each examined voxel, we will move it according to the averaged direction of SVs in the local window. Besides, a voxel can be moved to a new location only if it meets the following two conditions.

1. The new location is inside the 3D model (i.e., a non-empty object-voxel).
2. The new location is still close to the original skeleton (e.g., it cannot be more than two voxels away).

The above process is iteratively applied until there is no further movement. To avoid an infinite loop, a maximum number of iteration is also set. With the application of the proposed smoothing process, a jagged line can be straightened without losing sharp features. Three skeleton smoothing examples are shown in Fig. 7.

3.2.3. Sub-branches

As shown in Fig. 8(a), a thinned skeleton may have short sub-branches such as the one around the horse neck. These sub-branches will result in incorrect shape decomposition as shown in Fig. 8(b). It is challenging to check whether a given branch is a main or a sub-one. Although most sub-branches are short, classification based on the length is not robust since short branches can still be important parts of a skeleton in some occasions and removing short branches may remove key features such as toes or tails.
Although a sub-branch is not necessarily shorter than its adjacent voxel groups, it typically lies inside the surrounding voxel region of another main branch. Besides, one of its end voxels has to be an End-SV. A sub-branch removal algorithm is developed based on the above two observations.

Skeleton group \( G_1 \) is said to lie within its adjacent group \( G_2 \), if the following three conditions are all satisfied

1. \( G_1 \) is smaller than \( G_2 \), i.e., \( G_1 \) contains a smaller number of SVs.
2. The angle, \( \theta \), at the joint of branches \( G_1 \) and \( G_2 \) is within the range from 50 to 130 degrees.
3. The distance from \( G_1 \)'s SV, denoted by \( K_1 \), to its nearest SV \( K_j \) in \( G_2 \) is smaller than the radius of the surrounding surface around \( K_j \). This condition can be expressed mathematically as

\[
d(K_1, K_j) < \text{Radius}(K_j), \quad \forall K_j \in G_1,
\]

where \( K_j \) is the nearest skeleton of \( K_1 \) in \( G_2 \). \( d(K_1, K_j) \) is the distance between two voxels as described in Eq. (2), and \( \text{Radius}(K_j) \) is the radius of the surrounding surface around \( K_j \).

The radius of the surrounding surface around each SV can be estimated by assigning each surface voxel to its nearest SV. The surface voxel is the object-voxel which does not have 26 adjacent neighbors in a \( 3 \times 3 \times 3 \) grid. Its nearest SV is searched by computing the Euclidean distance together with a connectivity check. For example, the length of the path from one finger tip to another is longer than their Euclidean distance since there is no straight path between them. The distance between surface voxel \( V_j \) and SV \( K_i \) is defined as

\[
d(V_j, K_i) = \begin{cases} \|V_j - K_i\| & \text{if a straight path } V_j K_i \text{ exists.} \\ \infty & \text{otherwise.} \end{cases}
\]

To check if a straight path exists between \( V_j \) and \( K_i \), line segment \( V_j K_i \) is created. All voxels in \( V_j K_i \) can be derived by interpolation. If there is an empty voxel in \( V_j K_i \), the length of \( V_j K_i \) is set to infinity. Then, all surface voxels can be assigned to their nearest SVs.

Each \( K_i \) has a list, denoted by \( \text{List}(K_i) \), to record its associated surface voxels. This list is used to calculate the average radius, \( \text{Radius}(K_i) \), of its surrounding surface. Mathematically, we have

\[
\text{List}(K_i) = \{ V_j | V_j \text{ is the nearest SV} \},
\]

\[
\text{Radius}(K_i) = \frac{\sum_{V_j \in \text{List}(K_i)} d(V_j, K_i)}{\text{size of List}(K_i)},
\]

where \( \text{Radius}(K_i) \) is estimated by averaging the distances from all surface voxels in \( \text{List}(K_i) \) to \( K_i \).

The sub-branch removal process can be summarized as follows.

1. We associate all skeleton groups with some clusters using the following rule. All groups adjacent to each other belong to the same cluster. In other words, all skeleton groups in a cluster share a common joint. Note also that a group can belong to two clusters if both its two ends are joints.
2. For each cluster, we sort skeleton groups by their sizes, and examine each group sequentially from the smallest to the largest one. A skeleton group will be removed if it satisfies the three conditions discussed above. More than one group may be removed from a given cluster in this process.
3. Skeleton groups previously partitioned by the joint of a removed sub-branch will be merged.

A sub-branch removal example is shown in Fig. 8(c), where the two groups adjacent to the joint of the removed sub-branch around the neck are merged. More examples of sub-branch removal are shown in Fig. 9.

4. Shape decomposition

The goal of the shape decomposition process is to divide a 3D shape into meaningful parts. We begin with a simple object-voxel assignment process to decompose a 3D object roughly and, then, use the skeleton decomposition result to guide the shape decomposition to get a more accurate result. This idea is described below.

- **Step 1: Initial Shape Decomposition**
  The initial shape decomposition is achieved by assigning each object-voxel to its nearest SV based on Eq. (2). Thus, every object-voxels can be assigned to a part by following its associated SV. To reduce the complexity of shape analysis, we only

---

**Fig. 7.** Examples of three jagged skeletons and their smoothing results.

**Fig. 8.** Removing a sub-branch from the skeleton of a horse model: (a) a sub-branch in the neck region of a horse model, (b) incorrect shape decomposition around the neck caused by this sub-branch, (c) the sub-branch is removed and its adjacent skeleton groups are merged, (d) the shape decomposition result with the new decomposed skeleton.
consider surface voxels of the object-voxel and ignore interior voxels in this process. Fig. 2(e) shows an example of the assignment result. Each object-voxel is assigned to the same group as its nearest SV shown in the same color. After the assignment, each SV has an associated list recording its affiliated surface voxels and an estimated radius of its surrounding surface as described in Section 3.2. These data will be used in the next step.

• Step 2: Skeleton-guided Shape Decomposition

A skeleton-guided decomposition process is proposed to improve the initial shape decomposition result. This is motivated by the fact that most skeleton branches intersect at the base skeleton, and two problems in the initial shape decomposition arise accordingly. First, portion of the base part (i.e., the main body) can be mistakenly decomposed to protruding parts (e.g., the limb). Second, the base part can be mistakenly separated into two parts at the intersection. For the example in Fig. 2(e), the arm skeleton protrudes into the main body and intersects with the body skeleton. As a result, a portion of the main body is mistakenly assigned to the arm and the main body is mistakenly divided in the middle. The skeleton-guided shape decomposition process is used to fine-tune the result. It consists of the following three sub-steps.

- Step 2.A: Distinguish the base part (i.e., the main body) from protruding parts (e.g., limbs).
- Step 2.B: Delete a protruding skeleton that goes beyond its boundary and re-build the skeleton of the base part accordingly.
- Step 2.C: Divide a protruding skeleton into sub-groups depending on turning points.

In the following, we will discuss them in detail.

Step 2.A: Base Part Identification

To identify the base part, we define weighted accumulated distance for each part $P_i$ as

$$
\rho(P_i) = d_i(cen(P_i), cen(P)) \times s(P_i) + \sum_{j \neq i} d_j(mid(P_i), mid(P_j)) \times s(P_j),
$$

(5)

where $cen(P_i)$ and $cen(P)$ are the centers of mass of $P_i$ and the model $P$, respectively, $d_i$ is the Euclidean distance, $s(P_i)$ is the number of object-voxels belonging to $P_i$, and $mid(P_i)$ is the SV of $P_i$ that is closest to $cen(P_i)$. Since the distance between two parts is not equal to their Euclidean distance, we define the distance from $P_i$ to $P_j$ as

$$
d_r(mid(P_i), mid(P_j)) = ||path(mid(P_i), mid(P_j))||,
$$

(6)

where $path(mid(P_i), mid(P_j))$ contains all SVs along the path from the $mid(P_i)$ to $mid(P_j)$, and $|| \cdot ||$ is the path length. The idea is illustrated in Fig. 10(a). Finally, the base part that is closest to all other parts can be identified by finding the part that has the minimal accumulated distance. A few examples obtained by this algorithm are shown in Fig. 10(b), where the identified base part of each model is colored in red.

Step 2.B: Skeleton Invalidation and Mergence

To invalidate or merge protruding skeletons, we check all protruding skeleton groups connected to the base skeleton and classify them to either the invalidation or the merging category. The protruding group which intrudes the main body should be invalidated while the protruding group which is mistakenly separated from the base skeleton should be merged. The classification can be done based on the relative angle between the protruding skeleton and the base skeleton. If they are nearly parallel (i.e., the angle is close to $180^\circ$), this protruding skeleton is likely to belong to the base skeleton and should be classified to the merging category. Otherwise, it is classified to the invalidation category. The direction of a protruding skeleton is estimated from the joint connected to the base skeleton to its nearest turning point or to the other end if there is no other turning point. The direction of the base skeleton is estimated from one end, which is connected to the protruding part, to the other end.

After the classification job, we search for the boundary between the protruding part and the main body. Only the segment that goes beyond the protruding boundary will be invalidated or merged. The boundary can be detected by exploiting the fact that the radius of the surrounding surface along the protruding skeleton increases drastically when it crosses the boundary between the protruding part and the base part. For example, the surrounding surface along the arm skeleton in Fig. 2(e). The SV whose surface radius increases

Fig. 9. More examples of sub-branch removal, where the three skeletons in the top row contain sub-branches while the three skeletons in the bottom row are obtained by the proposed sub-branch removal algorithm.
When a protruding skeleton is long or connected to other subparts, the variation of the surrounding surface radius may not be a reliable indicator for the boundary of the body part and the protruding part. For example, the head skeleton of the crane model in Fig. 11(a) (shown in orange color) is bending and the variation of the surrounding surface radius along the skeleton is irregular. It is difficult to select a threshold for this path. To resolve this problem, we narrow down the search region of the protruding boundary. For each protruding group classified to the invalidation category, the boundary search region is confined to a small area controlled by the boundary point and the intersection joint. That is, if the average surface radius surrounding the base skeleton is $\delta$, only protruding SVs whose distance to the base skeleton close to $\delta$ need to be examined. We can express this condition mathematically as

$$\delta_1 - e \leq d(K_i, K_j) \leq \delta_2 + e, \quad \forall K_i \in [K_s, K_e],$$

where $[K_s, K_e]$ is the search region along the protruding skeleton, $K_i$ is the SV of the protruding group, $K_j$ is the SV of the base group that is nearest to $K_i$, and $\delta_1$ and $\delta_2$ are the average and the largest surface radii surrounding the base skeleton, respectively, and $e$ is an offset to slightly increase the search region. If no point in the search region meets the above criterion, we select $K_e$ as the boundary point. All protruding SVs between this boundary point and the intersection joint are invalidated. Furthermore, SVs belonging to these invalidated skeletons are assigned to their nearest base skeleton.

Fig. 11(b) shows the improved shape decomposition result based on skeleton merging is shown in Fig. 11(c).

**Step 2.C: Shape Decomposition with Turning Points**

Some protruding group can be further decomposed into subgroups with turning points. For example, meaningful parts of animal models, such as legs can be decomposed by analyzing the curvature of the skeleton. In Fig. 11(d), we show an improved shape decomposition result, where protruding parts such as legs and the neck of the crane model are decomposed into subgroups with selected turning points. After the skeleton decomposition process, we do not need to re-calculate or re-assign the object-voxel to its nearest skeleton-voxel. Instead, we only update its belonging group since its belonging SV might be merged or divided to a different group.

**5. Experimental results**

In the experiment, we adopted a collection of polygonal 3D models used in [17]. All of them were pre-converted into the same obj format, normalized to the same range, and voxelized to the corresponding volumetric models. In the voxelization process, we chose a smaller voxel size so as to get a higher resolution model in order to capture more details. As a tradeoff, its processing demanded more time and larger memory size, and its thinning result was more noisy. The resolution was finally chosen to be a $80 \times 80 \times 80$ grid since it stroke a good balance between model quality, memory size and computational complexity.
5.1. Comparison of decomposed 3D models

The effect of the skeleton refinement process is illustrated in Fig. 12, where we show the extracted skeletons of 12 models. The left one was obtained by the thinning operation alone while the right one was obtained by the thinning operation followed by the skeleton refinement process. It is clear that the skeleton refinement process does improve the quality of the extracted skeletons. We also provide the shape decomposition result in the same figure, where the identified base parts are shown in red and protruding parts are shown in different colors. With the improved skeleton, each model can be decomposed into its significant parts more accurately.

Based on decomposed 3D models, we constructed feature-preserving 3D thumbnails in Fig. 13, each of which consisted of 90 primitives. These thumbnails preserve sufficient details of their original models. We show thumbnails with 7, 11, 15 and 30 fitting primitives in Fig. 14, which demonstrates the robustness of the proposed voxel-based shape decomposition scheme.

Next, we compare the performance of the surface-based decomposition scheme [2] and the proposed voxel-based decomposition scheme for a crane model in Fig. 15. We see clearly from Fig. 15(a2)–(a4) that the decomposed mesh using the surface-based scheme does not represent the shape well since it becomes fragmented with adjacent parts being taken apart in the simplified crane model. Its skeleton and body measurement extracted from the base part lean toward the upper body, which is caused by missing pieces in the bottom part. This problem is fixed by the proposed voxel-based approach as shown in Fig. 15(b1)–(b3). In this comparison, we kept the file size of both thumbnails about the same (i.e., 8 KB per thumbnail), each of which contained 90 primitives.

5.2. Computational time and file size

The experiment was run on a desktop computer with an Intel Core 2 Duo 2.53 GHz CPU, and 4G RAM. The average processing time for creating a 3D thumbnail composed by 50 primitives is given in Table 1. Note that the processing time for primitive approximation will vary according to the number of primitives assigned. The primitive used in this work was the deformable cylinder (or d-cylinder in short), which was introduced in [2]. Approximating a thumbnail using 50 d-cylinders took 0.24 s on the average. 20 and 40 d-cylinders took 0.072 and 0.142 seconds, respectively.

For 3D models whose original file sizes were in the range of 160 – 870 KB, the size of their thumbnail composed by 7, 20, 50 d-cylinders were about 0.9 KB, 1.7 KB, 4 KB, respectively. Since the size of the thumbnail is primarily decided by the number of primitives (rather than the size of the original model). The size of a thumbnail can be much smaller than that of its original file for a complex model. It is worthwhile to emphasize that, since the extracted skeleton and the body measurement of a model are not

![Fig. 12. Comparison of skeletons obtained by the thinning operation alone (left) and by the thinning operation followed the skeleton refinement process (right).](image1)

![Fig. 13. Thumbnails obtained by the proposed voxel-based shape decomposition scheme, where the original 3D models were decomposed into multiple parts and each part was approximated by a fitting primitive.](image2)
affected by the number of assigned primitives, they can be reused to create thumbnails of different resolutions.

5.3. Subjective evaluation

We conducted a subjective test on the performance of the thumbnails obtained by the proposed method and Garland’s mesh simplification method [24]. A total of 30 people took part in this experiment. We randomly chose 18 models and obtained their simplified models using the above-mentioned two methods. Each model was simplified into 4 resolutions (with 10, 20, 40 and 90 primitives). We selected pairs of simplified models (called Model A and Model B) in a random order and placed them side-by-side with their original model. Each subject was requested to choose one of the three options: (1) Model A is closer to original one; (2) Models A and B are about the same; (3) Model B is closer to original one. One set of such test models was shown in Fig. 16.

The subjective test results for 4 different resolutions are shown in Table 2, where the number in each box shows the percentage of subjects who preferred that particular model.
people preferred the corresponding algorithm. For example, the first column shows that all subjects preferred our algorithm when the simplified model was composed by 10 primitives. It is clear that our method outperforms Garland’s method in all resolutions in the test. However, the gap becomes narrower when the number of primitives increases.

5.4. Discussion

Finally, we would like to point out one shortcoming of the proposed voxel-based method. That is, the decomposition result is highly affected by the quality of the extracted skeleton. If the skeleton does not represent the correct structure of a 3D model well, the proposed method may fail to decompose the model. Two such examples are given in Fig. 17. The skeleton in the middle part of the cactus model is skewed, and it does not have the same direction as the upper part and the lower part. As a result, the base part of the cactus fails to extend to the correct direction. The alien model has two big holes in the face and the skeleton obtained by thinning cannot capture this feature correctly. Then, one half of the face is missing in the created thumbnail. To improve these cases, we need a better skeletonization technique for 3D models which is an item for future research.

6. Conclusion

An innovative voxel-based shape decomposition method was presented in this work. The skeleton was first extracted and decomposed into multiple groups. Then, the volumetric model was decomposed into significant parts guided by the skeleton decomposition result. The significant parts of a 3D model can be well-preserved by its thumbnail representation. As compared with the surface-based shape decomposition scheme in [2], the new voxel-based shape decomposition scheme can represent the shape of each part better and decompose the model more accurately even if the original model is greatly simplified.

References