Interpolation of 3-D Binary Images based on Morphological Skeletonization

Vassilios Chatzis and Ioannis Pitas

The authors are with the Department of Informatics, Aristotle University of Thessaloniki. Corresponding author: Ioannis Pitas, University of Thessaloniki, GR-54006 Thessaloniki, GREECE, phone,fax: +30-31-996304, e-mails: [chatzis,pitas]@zeus.csd.auth.gr,

Abstract

In this paper, the Morphological Skeleton Interpolation (MSI) algorithm is presented. It is an efficient shape-based interpolation method used for interpolating slices in a 3-D binary object. It is based on morphological skeletonization, which is used for 2-D slice representation. The proposed morphological skeleton matching process provides translation, rotation and scaling information at the same time. The interpolated slices preserve the shape of the original object slices, when the slices have similar shapes. It can also modify the shape of an object when the successive slices have not similar shapes. Applications on artificial and real data are also presented.

Keywords

3-D Binary Image Interpolation, Shape Representation, Mathematical Morphology, Skeletonization

I. Introduction

Many medical imaging systems that use magnetic resonance, X-rays, positron emission or ultrasound, provide 3-D image data formed by stacking a series of 2-D digital images (slices). It is not always possible to acquire the data provided by such systems at the desired discretization level, because of either technological or medical reasons. Usually, the pixel size within a slice is smaller than the inter-slice distance. In order to deal with the problem of visualizing and processing such data, many interpolation techniques have been developed. Interpolation techniques can be divided into two broad categories: *grey-level* and *shape-based.*

Grey-level interpolation techniques estimate the interpolated pixel value by using the pixel values of the neighboring slices within a small pixel neighborhood, without using any kind of shape information. The simplest of these methods is the *nearest-neighbor* technique that is also referred as *zero-order* interpolation [1]. *Linear grey level* interpolation is the most common technique [1], [2], although higher order polynomials may also be used [3]-[5].

However, grey-level interpolation techniques assume that the 3-D image can be modeled by smooth greylevel transitions. Thus, interpolated frames contain in most cases blurred objects and artifacts.

Most of the *shape-based* interpolation techniques assume that the original 3-D greyscale image is first segmented into a 3D binary image that contains only the objects of interest. Several methods have been applied to extract object characteristics and afterwards apply an interpolation procedureusing in many cases grey-level methods. In [6], [7] the object contours are interpolated. Another method using a distance transform have been proposed in [8]. Several modifications of this method have been proposed [9]-[12]. In [13] another method based on set distances has been implemented by using mathematical morphological operators. In [14] feature points are selected from successive frames using the gradient value as a criterion. A correspondence between them is found using a matching procedure. Then, grey-level interpolation is applied for the feature points towards the direction defined by the correspondence. Feature points (cores), are also used in [15], [16] defined by using a fuzzy measure of boundariness and the medial axis transform. In this paper a novel shape-based interpolation method, the Morphological Skeleton Interpolation (MSI), is proposed based on object representation using mathematical morphological skeletonization.

Mathematical morphology is a very rich and powerful tool used for the representation and analysis of binary and greyscale images [17]-[22]. The morphological image representation has been used for the description of the geometrical characteristics of image objects as well as for binary image compression. The morphological skeleton is the most popular morphological method for shape representation. As can be found in several comparative studies [23]-[26], the main disadvantage of the morphological skeletonization is the lack of robustness especially in impulsive noise, something that can be considered as a general characteristic of all the morphological operators [27]. The recently proposed Generalized Fuzzy Mathematical Morphology (GFMM) [28] reduces the sensitivity of morphological operations to impulses and provides robust object skeletonization.

The paper is structured as follows. In section II the formulation of the interpolation problem and the necessary background of the classical and fuzzy mathematical morphologies are presented. The proposed Morphological Skeleton Interpolation method is presented in section III. Experimental results when the MSI is applied on artificial and real-world data are presented in section IV. Conclusions are drawn in section V.

II. Mathematical Morphology background

In the framework of Mathematical Morphology, we are interested in the set of object slice pixels. A 2-D structuring element B can be represented by a binary set. The reflection of the set is given by $\check{B} = \{-b : b \in B\}$ We also denote B^C the complement of the set B given by $B^C = \{b \in Z^2 : b \notin B\}$

Let A, B, X and Y be binary images, subsets of Z^2. *Minkowski addition* and *Minkowski substraction* are the basic operation of mathematical morphology:

$$\begin{aligned} A \oplus B &= \{x \in Z^2 : \exists b \in B, x - b \in A\} && (1) \\ &= \{a + b : a \in A, b \in B\} && (2) \\ A \ominus B &= \{x \in Z^2 : \forall b \in B, x + b \in A\} && (3) \\ &= (A^C \oplus \check{B})^C && (4) \end{aligned}$$

It is well known that all morphological transformations, from the simplest (dilation, erosion, opening, closing) to the more complex ones, are based on Minkowski addition [19].

For example $A \odot B$ is the opening of A by B given by $A \odot B = A \ominus B \oplus B$.

The representation of a 2-D object $\mathcal{O}_k$ using the morphological skeleton is given by [23]:

$$\mathcal{O}_k = \bigcup_{r=0}^{N} S(r) \oplus rB \tag{5}$$

where the sets $S(r)$ known as *skeletal sets* are calculated by:

$$S(r) = (\mathcal{O}_k \ominus rB) \backslash (\mathcal{O}_k \ominus rB) \odot B \tag{6}$$

where $\mathcal{O} \odot B$ is the morphological opening of $\mathcal{O}$ by the structuring element B, $X \backslash Y$ is the set difference, rB is the rth-order homothetic of B ($rB = B \oplus \ldots \oplus B$, r times) and N is the largest integer such that $X \ominus NB \neq \emptyset$. Classical morphological skeletons are sensitive to noise. This drawback can be overcome by using soft morphological operations such as the skeletons defined through the generalized fuzzy mathematical morphology [28].

III. Morphological Skeleton Interpolation

A. *The basic steps of the Morphological Skeleton Interpolation (MSI) algorithm*

Let V be a 3-D image (called a *volume*) defined on the Euclidean grid Z^3. Any point on the lattice, called a *voxel*, is defined as $V(i,j,k)$. The set of greyscale values $V(i,j,k)$ for a certain k defines a slice of the volume. A binary volume V_B takes values in $\{0,1\}$. Usually, an object $\mathcal{O}$ that is contained in a volume, is sampled sparsely in the third direction. Slice interpolation will provide approximate object slices for intermediate k values.

Let us assume that two objects $\mathcal{O}_k$ and $\mathcal{O}_{k+1}$ are contained in two successive slices of an object $\mathcal{O}$ which is contained in a binary volume V_B. The aim is to interpolate L slices, that will contain the objects $\mathcal{O}_{k,1}$, $\mathcal{O}_{k,2}$, ..., $\mathcal{O}_{k,L}$, between the slices k and $k+1$. In order to describe the basic steps of the MSI algorithm we shall first assume that every slice contains only one connected and compact object. The algorithm that is proposed in this paper is presented in Figure 1 and consists of the following steps:

1. **Object skeletonization**
2. **Skeleton matching**
3. **Interpolation transformation calculation**
4. **Skeleton modification**
5. **Object reconstruction**

A.1 Object skeletonization

Object skeletonization is performed on every binary volume slice. The skeletonization is applied by using either classical morphological operations or fuzzy morphological operations that provide more robust skeletal sets. The shape and the size of the structuring element should remain the same for all slices. Skeletonization produces skeletal points, grouped in skeletal sets $S(r)$ defined in (6). Skeletal points are considered as 3-D data (x, y, r) where x, y are the space coordinates of every skeletal point in the slice k and r is the step of the skeletonization procedure. An example is shown in Figure 2 where two artificially generated successive slices are presented in Figures 2a and 2b. The skeletal points of the corresponding objects are presented in Figures 2c and 2d, where the value of r is presented by changes in grey-level shown in the grey-level bar. Generally, the skeletal points with small values of r describe details of an object where the skeletal point with great values of r describe the innermost points of an object, in our example the center of the cross.

A.2 Skeleton matching

In the following, the skeleton is considered as a set S_k of 3-D points for a certain slice k:

$$S_k = \{(x, y, r) : (x, y) \in S(r)_k\} \tag{7}$$

where $S(r)_k$ are the skeletal sets of the object $\mathcal{O}_k$. The matching step is applied on the skeletons S_k and $S_{(k+1)}$ that correspond to the objects contained in two successive slices $k, k+1$. Any matching procedure (e.g. the Iterative Closest Point (ICP) [29]) can be used to provide a pair of transformations $T_{k,k+1}(\cdot)$ and $T_{k+1,k}(\cdot)$:

$$S'_{(k+1)} = T_{k,k+1}(S_k) \tag{8}$$

$$S'_k = T_{k+1,k}(S_{(k+1)}) \tag{9}$$

Since (x, y) are the spatial coordinates of the skeletal points and the third dimension describes the object size, a rotation towards the r-direction has no physical sense. Thus, the matching procedure should not permit rotation towards the r-direction, but only in the x, y plane. Rotation and translation in the x, y plane describes any possible shifting of the object in the certain plane. Translation towards the r-direction describes any possible scale changes of the objects size. Thus, skeletonization matching provides at the same time, translation, rotation and scale information.

A.3 Interpolation transformation calculation

A pair of transformations $T_{k,k+1}$ and $T_{k+1,k}$ is provided by the matching process. Let us assume that we want to produce L interpolated slices equally distributed between the slices k and $k+1$ along the z axis. The transformations $T_{k,k+1}$, $T_{k+1,k}$ are easily modified to provide a transformation pair $T_{k,k+1,l}$ and $T_{k+1,k,l}$, at a certain position l which correspond to the interpolated slice $l = 1, 2, \ldots, L$. The $l = 1$ interpolated slice will be the nearest to the k slice and the $l = L$ slice the nearest to the $k + 1$ slice. The translations along x, y and r direction can be found by multiplying the translation found in the previous step, with the fraction of the distance from the interpolated slice l to the slice k, to the distance from slice k to $k + 1$. When the L interpolated slices are equally distributed, the translation of the transformation $T_{k,k+1}$ is multiplied with $\frac{l}{L+1}$ and the translation of the transformation $T_{k+1,k}$ is multiplied with $\frac{L+1-l}{L+1}$, to provide translations that lead from the points of both original slices to the interpolated slice l.

Rotations should be restricted to the x, y plane. The transformations $T_{k,k+1}$ and $T_{k+1,k}$ provide known rotation angles on the x, y plane. The rotation angles that lead from the points of the original slices to the interpolated slice l, are calculated by using the same factors $\frac{l}{L+1}$ and $\frac{L+1-l}{L+1}$ as in the translation case. The modified rotation and translation form a new pair of transformations $T_{k,k+1,l}$ and $T_{k+1,k,l}$. It is obvious that it is easy to generalize and calculate transformations for every point between the successive slices k,$k + 1$.

A.4 Skeleton modification

The transformations $T_{k,k+1,l}$, $T_{k+1,k,l}$, which were calculated from the previous step, are applied on the skeletons of the objects $\mathcal{O}_k$ and $\mathcal{O}_{(k+1)}$ and two new skeletons $S_{k,k+1,l}$ and $S_{k+1,k,l}$ are produced by:

$$S_{k,k+1,l} = T_{k,k+1,l}(S_k) \tag{10}$$

$$S_{k+1,k,l} = T_{k+1,k,l}(S_{(k+1)}) \tag{11}$$

The modification process follows. The skeletal points of the transformed skeleton that corresponds to the larger object are moved towards the nearest skeletal point of the modified skeleton of the smaller object. The nearest point is found by using the Euclidean distance between the points in the 3-D space. The movement of the skeletal points depends on the distance of the interpolated slice from the reference slice. Thus, a skeleton $S_{k,l}$ that corresponds to the interpolated object $\mathcal{O}_{k,l}$ is produced:

$$S_{k,l} = M(S_{k,k+1,l}, S_{k+1,k,l}) \tag{12}$$

where M denotes this modification. Then, the skeleton matching step is executed again using the produced modified skeleton $S_{k,l}$ and the transformed skeleton of the smaller object which remains unchanged. The interpolation transformation calculation and the skeleton modification steps are then re-applied and another interpolated slice is produced. An example is presented in Figure 2. Two slices should be interpolated between the frames shown in Figures 2a and 2b. The skeleton of the object shown in Figure 2c is modified to the skeletons shown in Figures 2e and 2f, translated towards x, y and r direction. The reduction of the values of r is shown in the grey-level bar. The maximum value (27) that corresponds to the central point of the cross, was reduced to 23 and then to 20. This reduction corresponds to the reduction of the objects size. The r values of the other skeletal points were similarly reduced.

A.5 Object reconstruction

The last step of the algorithm is object reconstruction from skeletons. For every slice pair $\mathcal{O}_k$, $\mathcal{O}_{k+1}$, L new objects $\mathcal{O}_{k,l}$ are reconstructed from the L new skeletons produced after the iterations, by applying (5). For example, the reconstruction process applied on the skeletons shown in Figures 2e and 2f, leads to the objects shown in Figures 2g and 2h.

B. The branching problem

A branching problem arises when the number of connected components of an object that is contained in a certain slice and the number of the connected components contained in the next slice are different. The pair of slices where the branching happens, can be easily detected by using a preprocessing step that counts the connected components of the objects in all slices [30]. The branching problem will be handled by decomposing the problem into a certain number of interpolations between slices that contain only one object. The *branching decomposition procedure* is described in the following, assuming that the connected components are compact sets.

Let k and $k+1$ be two successive slices where the slice k contains one connected object $\mathcal{O}_k$ and the slice $k+1$ contains an object $\mathcal{O}_{(k+1)}$ which consists of P connected components $\mathcal{O}_{(k+1),p}$, $p = 1, 2 \ldots, P$. Then, the slice $k+1$ is decomposed into P slices each one containing one of the connected components $\mathcal{O}_{(k+1),p}$. The object $\mathcal{O}_{(k+1)}$ can be reconstructed from the union of its components:

$$\mathcal{O}_{(k+1)} = \bigcup_p \mathcal{O}_{(k+1),p} \tag{13}$$

Then MSI is applied P times between the object $\mathcal{O}_k$ and each of the component $\mathcal{O}_{(k+1),p}$ producing P sets of interpolated slices $\mathcal{O}_{k,l,p}$. The interpolated slices are calculated as the union of the p interpolated slices at every interpolation level l:

$$\mathcal{O}_{k_l} = \bigcup_p \mathcal{O}_{k,l,p} \tag{14}$$

When interpolation should be applied between slices that both contain more than one connected components then an object correspondence should be found first. Let us assume that the slice $k+2$ contains Q connected components with $Q > P$. The centroids c_p, c_q of the $P+Q$ components and the 2-D distances $d(c_p, c_q)$ between them are calculated. For every component q the minimum distance d_{min} of the distances $d(c_p, c_q)$ is found and each component $\mathcal{O}_{(k+2),q}$ is associated to the component $\mathcal{O}_{(k+1),p}$ which provides the minimum centroid distance d_{min}. The interpolation algorithm is then applied on the pairs of the associated components. This procedure can be easily modified to solve the inverse branching problem that happens when the number of connected components in the next slice is less than the number of connected components in the current slice $Q < P$.

An example of such a case is given in Figure 3a where the tree-like object slices are presented. In the first slice the object consists of one connected component and it is separated into two connected components in the second slice, and into five connected components in the third slice. The problem is decomposed into a set of two interpolations between the first slice and each of the connected components of the second slice, shown in Figure 3c, and another set of five interpolations between the connected components of the second slice and the connected components of the third slice, associated using the minimum centroid distance criterion, shown in Figure 3b. Thirty slices were interpolated between each pair of original slices. The interpolated slices are shown in Figure 4. Two visualizations of the interpolated object are shown in Figures 5a and 5b.

C. The non-compactness problem

A non-compactness problem arises when a connected component of an object contains holes. The slices with non-compact objects can be easily detected by using a preprocessing step [30]. We mention that the existence of holes in an object changes significantly its morphological skeleton representation. Thus, in order to apply the MSI algorithm, the non-compact objects will be decomposed into a certain number of compact objects, by using the *non-compactness decomposition procedure* described subsequently.

Assuming that the branching problem has been handled by using the previously described branching decomposition procedure, the MSI will be applied on objects that contain only one connected component. Let $\mathcal{O}_k$ be the connected object of slice k that contains a certain number of holes H. Then, the slice k is decomposed into 2 slices. The first contains the filled object $\mathcal{O}_{k,f}$ and the second contains the union of the holes $\bigcup_h \mathcal{O}_{k,h}$, where $h = 1, 2, \ldots, H$ denotes the connected components (holes) of the second slice. The object $\mathcal{O}_k$ can be reconstructed as the difference of the 2 slices:

$$\mathcal{O}_k = \mathcal{O}_{k,f} \setminus \left(\bigcup_h \mathcal{O}_{k,h} \right) \tag{15}$$

An example is shown in Figure 6 where ithe object in Figure 6a is decomposed into the filled object shown in Figure 6b and the two holes shown in Figure 6c. The decomposition is applied on all slices that their objects contain holes and two new slice sequences are created: the *filled slice sequence* that contains the filled objects and the *hole slice sequence* that contains the holes. Then MSI will be applied separately on the two slice sequences. When the number of holes in successive slices are different, a branching problem arises and the branching decomposition procedure should be applied on the hole slice sequence.

Finally, the interpolated slices are reconstructed from the results of the two interpolations, by using (15):

$$\mathcal{O}_{k,l} = \mathcal{O}_{k,l,f} \setminus \left(\bigcup_h \mathcal{O}_{k,l,h} \right) \tag{16}$$

This procedure can easily be extended to solve the problem that arises when the hole in an object contains another object that contains another hole etc.

D. Comments on the MSI algorithm

The MSI algorithm uses morphological skeletonization to find an object representation that describes at the same time the object shape and size. The skeletonization and the corresponding reconstruction processes are based on morphological operations which are very fast. Moreover, the skeletal points are a small percentage of the total number of objects points. The skeleton matching, transformation and modification steps are applied only to the skeletal points and not to the total number of the objects points. The MSI algorithm has also the following characteristics. Any number of slices can be interpolated between two successive slices. The interpolated slices may not be equally distributed between the original slices. The skeletonization step is performed only once and can be used to produce interpolation at different interpolation levels. The interpolated slices can be stored in a compressed form using the skeletons. The lack of robustness on impulsive noise, can be overcome by using more robust implementations of morphological operations such as the skeletonization based on the Generalized Fuzzy Mathematical Morphology [28]. Besides, the MSI can embody the known shape based interpolation algorithms, which can be applied during the skeleton modification step, on the reconstructed objects from the pair of object skeletons $S_{k,k+1,l}$ and $S_{k+1,k,l}$. The result is expected to be better since the reconstructed objects have dealt with translation, rotation and scaling problem.

IV. Experimental results

Experiments were performed on artificial and real-world data coming from dental morphology applications. The artificially generated data were selected to present the features of the algorithm. The skeletonization and the reconstruction steps were applied by using a 3×3 square structuring element. The Iterative Closest Point (ICP) algorithm was used at the skeleton matching step to produce the best matching transformations [29]. ICP was slightly modified to permit rotations only in the image plane.

First MSI was applied on slices that contain circular cuts of an object that can be described as a sloping cone. This experiment presents the ability of the algorithm to interpolate slices that contain shifted objects that differ in size. The original slices are binary 256×256 images and are presented in Figure 7. Fourteen slices were interpolated between each pair of them. The interpolated slices between the first two slices of the original object are shown in Figure 8 where the first and the last frame labeled with 0 and 16 are the original slices labeled in Figure 7 with 0 and 1 respectively. In this example it is shown that the shape of the original object is preserved and the object of the first frame is translated and shrinked towards the second original frame. Two projections of the interpolated object are shown in Figures 9a and 9b.

The second example presents the capability of MSI to interpolate slices of a rotating staircase-like object. The original eleven 256×256 slices of a helical object are presented in Figure 10. Ten slices were interpolated between each pair of them. The interpolated slices between the first two slices of the original object are shown in Figure 11 where the first and the last frame labeled with 0 and 11 are the original slices labeled in Figure 10 with 0 and 1 respectively. It is shown in this example that the shape of the original object is preserved and the object of the first frame is rotated to fit the second original frame. Two visualizations of the interpolated object are shown in Figures 12a and 12b.

Two examples of the MSI algorithm applied on a dental morphology study follows. Data comes from several mechanically sliced teeth in order to visualize their morphology. Such studies are very important in endodontology. The first tooth (central maxillary incisor) consists of 22 slices which are presented in Figure 13. Mechanical slicing through a microtome allows only the creation of a small number of slices. The holes correspond to the dental root canal. The non-compactness decomposition procedure is applied and decomposes the problem to an interpolation between compact objects that represent the tooth crown, and another interpolation between compact objects that represent the dental root canal. Sixteen slices were interpolated between each pair of original slices. Two visualizations of the outer surface of the interpolated tooth are shown in Figures 14a and 14b, and two visualizations of the inner surface (the interpolated root canal) are shown in Figures 15a and 15b. The result can be produced from the two interpolated volumes by using an *xor* operation. The branching decomposition procedure on real data is applied on maxillary molar that consists of 30 slices shown in Figure 16. Sixteen slices were interpolated between each pair of original slices. A visualization of the outer surface of the interpolated tooth is shown in Figure 17.

Regarding the compression that can be achieved using the skeletons, notice that the number of the skeletal points varied from 1% to 20% of the object pixels depending on the size and the shape of the structuring element and the object itself. The skeletonization time (step 1) also depends on the size and the shape of the structuring element and the object. When using a 3×3 square structuring element, the skeletonization time (using a SGI Indy R4400, 200MHz, 64MB, IRIX5.3) varied from 2 to 20 seconds depending on the object area, with an average time of about 6 seconds per 10000 objects pixels. The interpolation time (steps 2-5) depends on the number of the skeletal points and on the time of the matching process. An average time of about 5 seconds per interpolated slice was achieved using the ICP matching algorithm.

V. Conclusions

The Morphological Skeleton Interpolation algorithm was presented as a shape-based interpolation method for interpolation of 3-D objects. Morphological skeletonization was used to describe the basic object characteristics. The morphological skeleton matching process provides translation, rotation and scale information at the same time. Results from the application of MSI on artificial and dental data showed that MSI produces interpolated slices that preserve the shape of the original object slices when the slices have similar shapes and can also modify the shape of an object when the successive slices have different shapes. Since MSI is based on morphological operations which are very fast and the main steps of the algorithm are applied only to the skeletal points, which are a small percentage of the total object points. Skeletons can also be used for the compression of the interpolated slices. Therefore MSI has been revealed as a simple, fast and efficient method for interpolation of 3-D binary images.

References

[1] W. K. Pratt, *Digital Image Processing*, Wiley, 1991.

[2] S. M. Goldwasser, R. A. Reynolds, D. A. Talton, and E. S. Walsh, "Techniques for the rapid display and manipulation of 3-d biomedical data," *Comp. Med. Imag. and Graphics*, vol. 12, pp. 1–24, Jan. 1988.

[3] M. Levoy, "Display of surfaces from volume data," *IEEE Comp. Graphics Appl.*, pp. 29–37, May 1988.

[4] J. K. Udupa, G. T. Herman, P. S. Margasahayam, L. S. Chen, and C. R. Meyer, "3d88: A turnkey system for the display and analysis of 3-d medical objects," in *SPIE Proc.*, 1986, vol. 671, pp. 154–168.

[5] S. M. Goldwasser, R. A. Reynolds, D. Talton, and E. Walsh, "Real time display and manipulation of 3-d ct, pet and nmr data," in *SPIE Proc.*, 1988, vol. 671, pp. 139–149.

[6] W. C. Lin, C. C. Liang, and C. T. Chen, "Dynamic elastic interpolation for 3-d medical image reconstruction from serial cross-sections," *IEEE Trans. Med. Imaging*, vol. 7, pp. 225–232, Sept. 1988.

[7] S. Y. Chen, W. C. Lin, C. C. Liang, and C. T. Chen, "Improvement on dynamic elastic interpolation technique for reconstructing 3-d objects from serial cross-sections," *IEEE Trans. Med. Imaging*, vol. 9, pp. 71–84, Mar. 1990.

[8] S. P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects," *IEEE Trans. Med. Imaging*, vol. 9, pp. 32–42, Mar. 1990.

[9] G. T. Herman, J. Zheng, and C. A. Bucholtz, "Shape-based interpolation," *IEEE Comput. Graphics Applicat.*, vol. 12, no. 3, pp. 69–79, 1992.

[10] W. E. Higgins, C. Morice, and E. L. Ritman, "Shape-based interpolation of tree-like structures in three-dimensional images," *IEEE Trans. Med. Imaging*, vol. 12, no. 3, pp. 439–450, 1993.

[11] R. A. Lotufo, G.T. Herman, and J. K. Udupa, "Combining shape-based and grey-level interpolation," in *Proc. SPIE Visualization Biomed. Comput.*, 1992, pp. 289–298.

[12] G. J. Grevera and J. K. Udupa, "Shape-based interpolation of multidimensional grey-level images," *IEEE Trans. Med. Imaging*, vol. 15, no. 6, pp. 881–892, 1996.

[13] S. Beucher, "Sets, partitions and functions interpolations," in *Mathematical morphology and its applications to image and signal processing*, 1998, vol. 12, pp. 307–314.

[14] A. Goshtasby, D. A. Turner, and L. V. Ackerman, "Matching tomographic slices for interpolation," *IEEE Trans. Med. Imaging*, vol. 11, no. 4, pp. 507–516, 1992.

[15] D. T. Puff, D. Eberly, and S. M. Pizer, "Object-based interpolation via cores," in *Proc. SPIE: Med. Imaging 94, Image Procesing*, 1994, vol. 2167, pp. 143–150.

[16] B. S. Morse, S. M. Pizer, and D. S. Fritsch, "Robust object representation through object-relevant use of scale," in *Proc. SPIE: Med. Imaging 94, Image Procesing*, 1994, vol. 2167, pp. 104–115.

[17] R.M. Haralick, "Image analysis using mathematical morphology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, July 1987.

[18] I. Pitas and A.N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publ., Boston, MA, 1990.

[19] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[20] ed. J. Serra, *Image Analysis and Mathematical Morphology, Part II: Theoretical Advances*, Academic Press, London, 1988.

[21] J. Serra and P. Soille, *Mathematical Morphology and its Applications to Image Processing*, Kluwer Academic Publ., Boston, MA, 1994.

[22] P. Maragos, R.W. Schafer, and M. Akmal, *Mathematical Morphology and its Applications to Image and Signal Processing*, Kluwer Academic Publ., Boston, MA, 1996.

[23] P. A. Maragos and R.W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, pp. 228–244, October 1986.

[24] I. Pitas and A.N. Venetsanopoulos, "Morphological shape decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 38–45, January 1990.

[25] J.M. Reinhardt and W.E. Higgins, "Comparison between the morphological skeleton and morphological shape decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 951–957, September 1996.

[26] J.M. Reinhardt and W.E. Higgins, "Efficient morphological shape representation," *IEEE Trans. on Image Processing*, vol. 5, no. 1, pp. 89–101, January 1996.

[27] Demin Wang, Veronique Haese-Coat, Alain Bruno, and Joseph Ronsin, "Some statistical properties of mathematical morphology," *IEEE Trans. on Signal Processing*, vol. 43, no. 8, pp. 1955–1965, August 1995.

[28] V. Chatzis and I. Pitas, "A generalized fuzzy mathematical morphology and its application in robust 2d and 3d object representation," submitted in IEEE Trans. on Image Processing, 1998.

[29] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, February 1992.

[30] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1992.

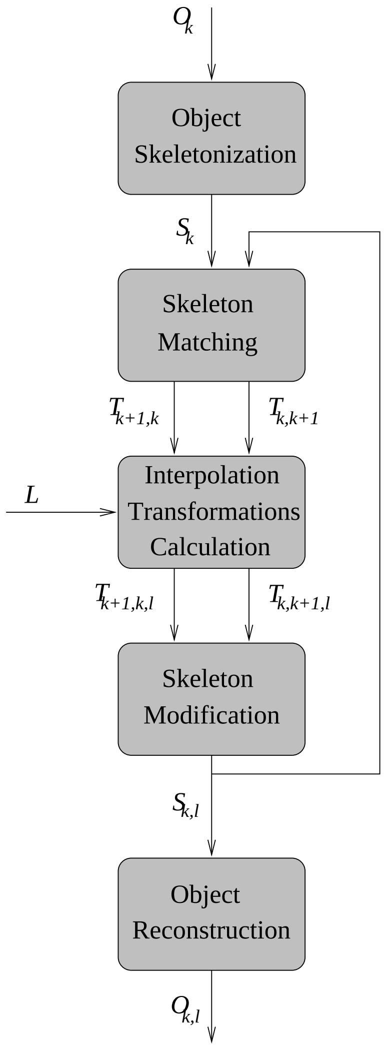


Fig. 1. The basic steps of the Morphological Skeleton Interpolation algorithm.

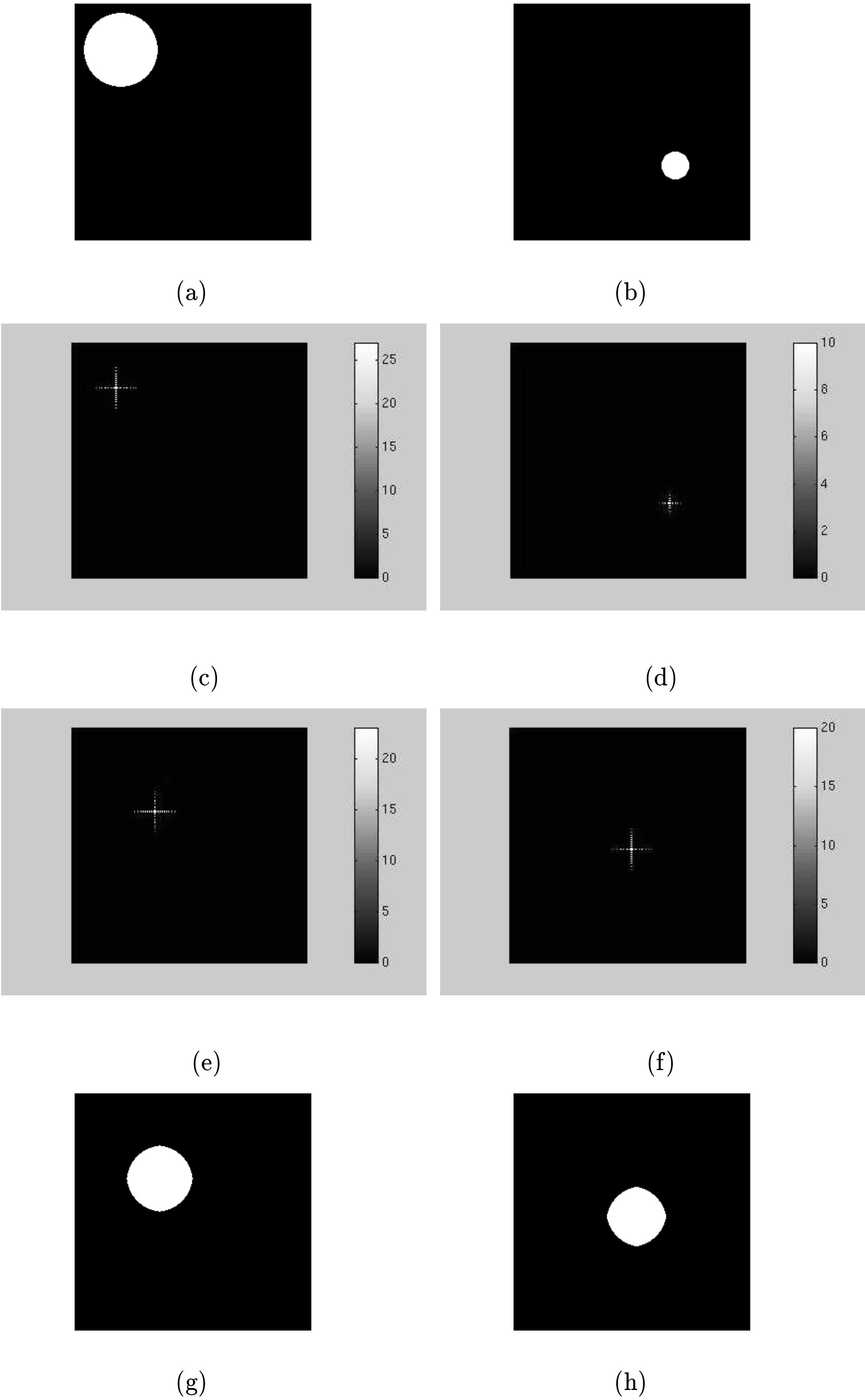


Fig. 2. (a),(b) Two successive slices of an object, (c),(d) the corresponding skeletons, (e),(f) the interpolated skeletons and (g),(h) the interpolated object slices.

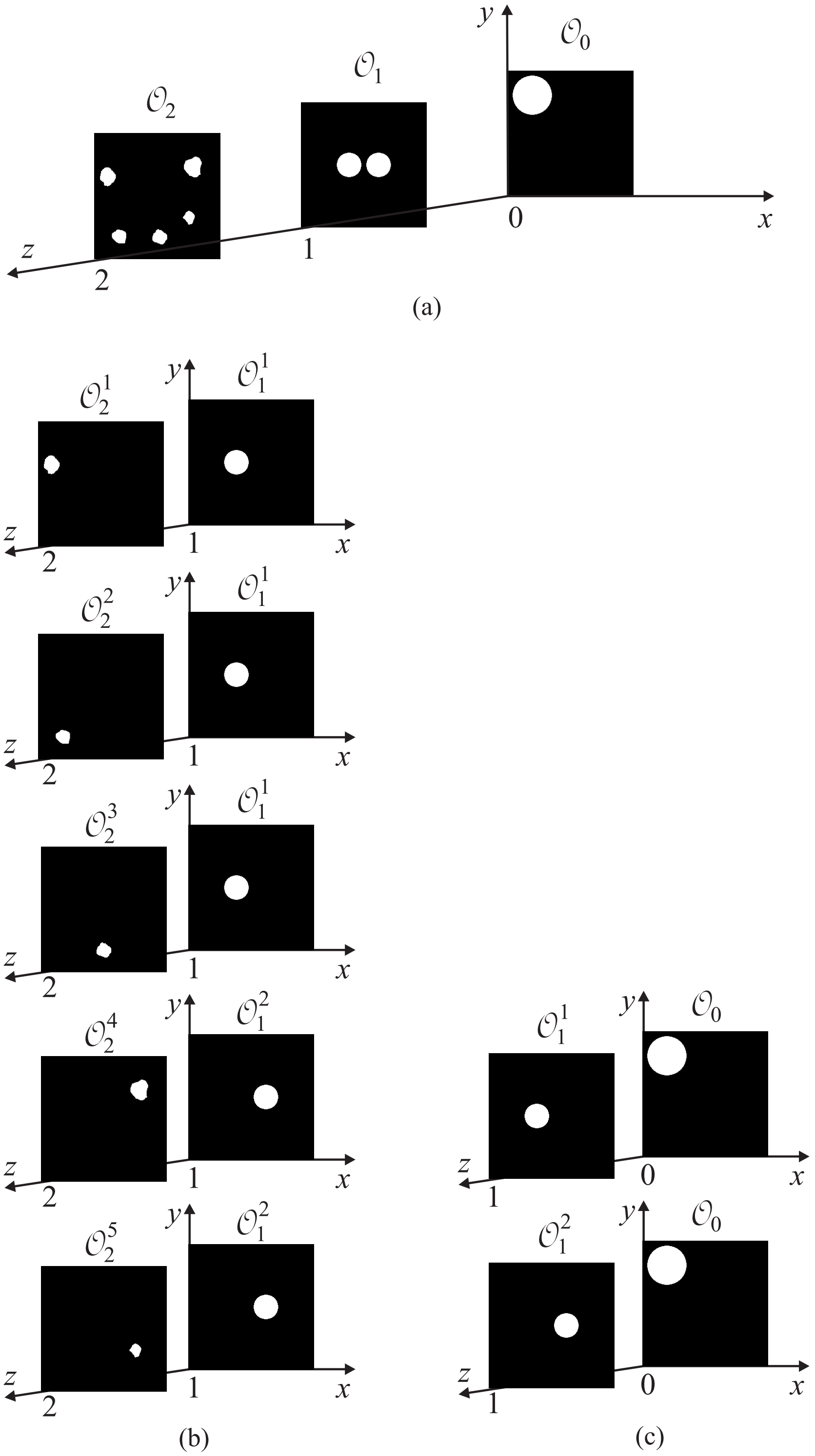


Fig. 3. (a) Three object slices (0,1,2) that consist of one, two and five connected components respectively. (b) The interpolation between the slices 1 and 2 is decomposed into five interpolations. (c) The interpolation between the slices 0 and 1 is decomposed into two interpolations.

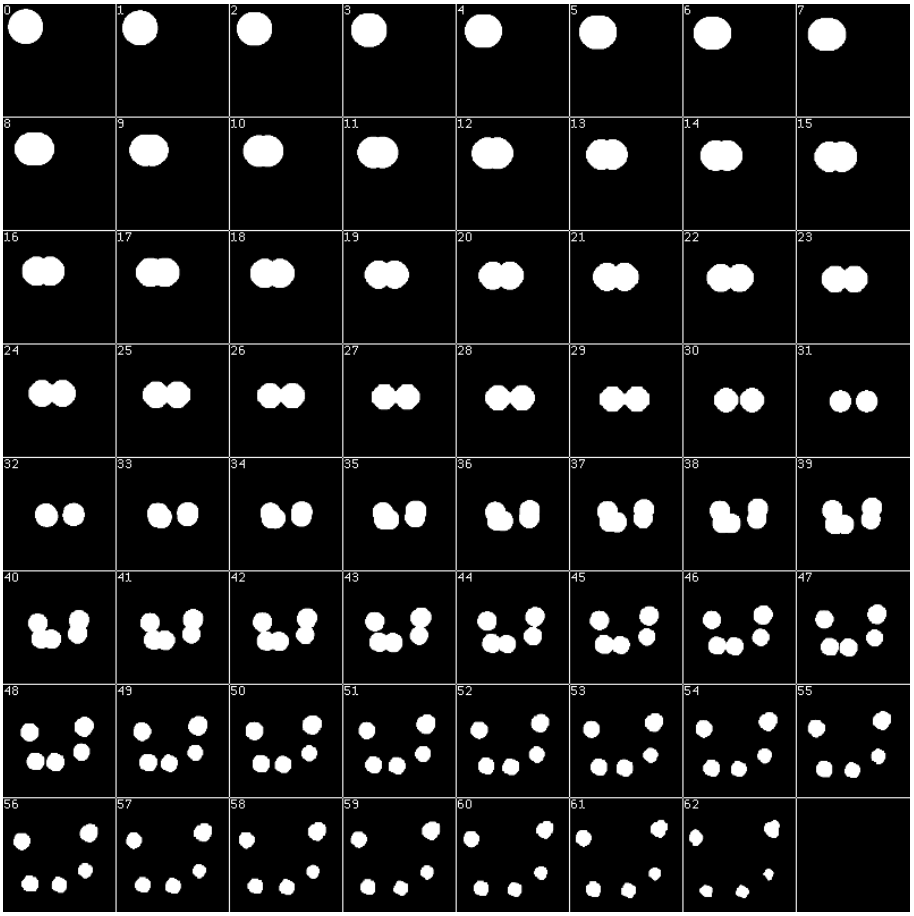
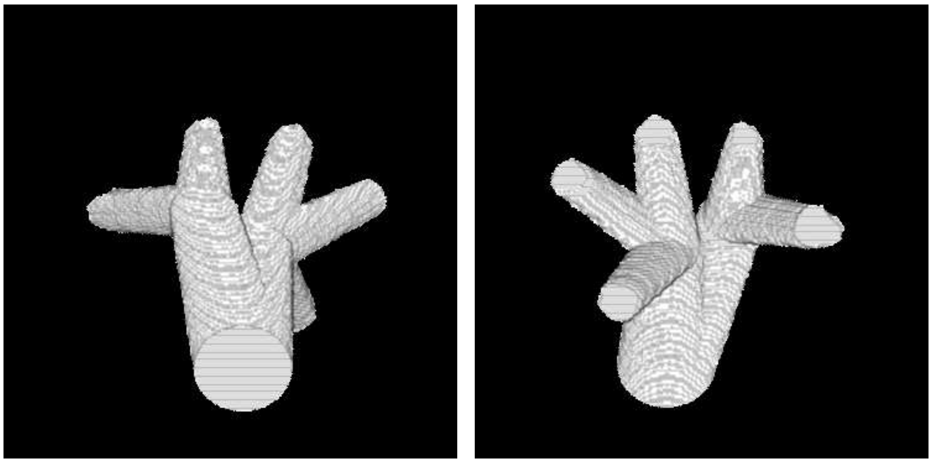


Fig. 4. The sixty interpolated slices between the original three slices (0,31,62) of the tree-like volume.



(a) (b)

Fig. 5. Two visualizations of the interpolated tree-like volume.

(a) (b) (c)

Fig. 6. An object with two holes (a) is decomposed into the filled object (b) and the holes (c).

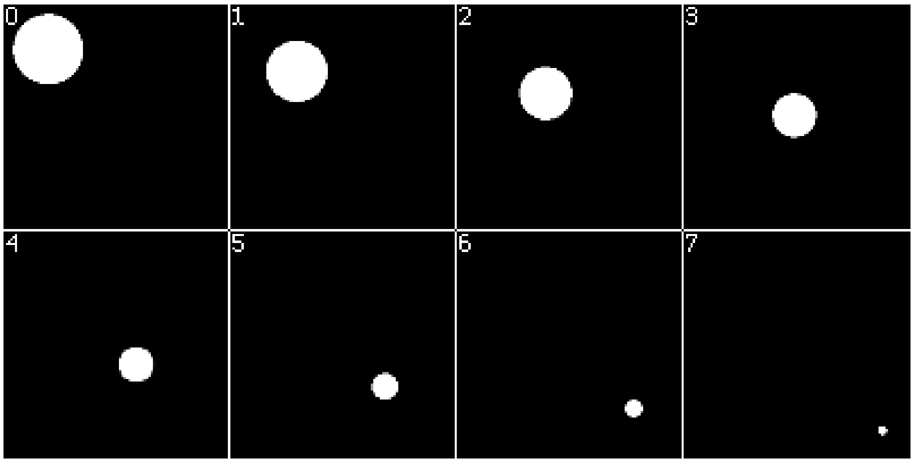


Fig. 7. The original slices of the sloping cone volume.

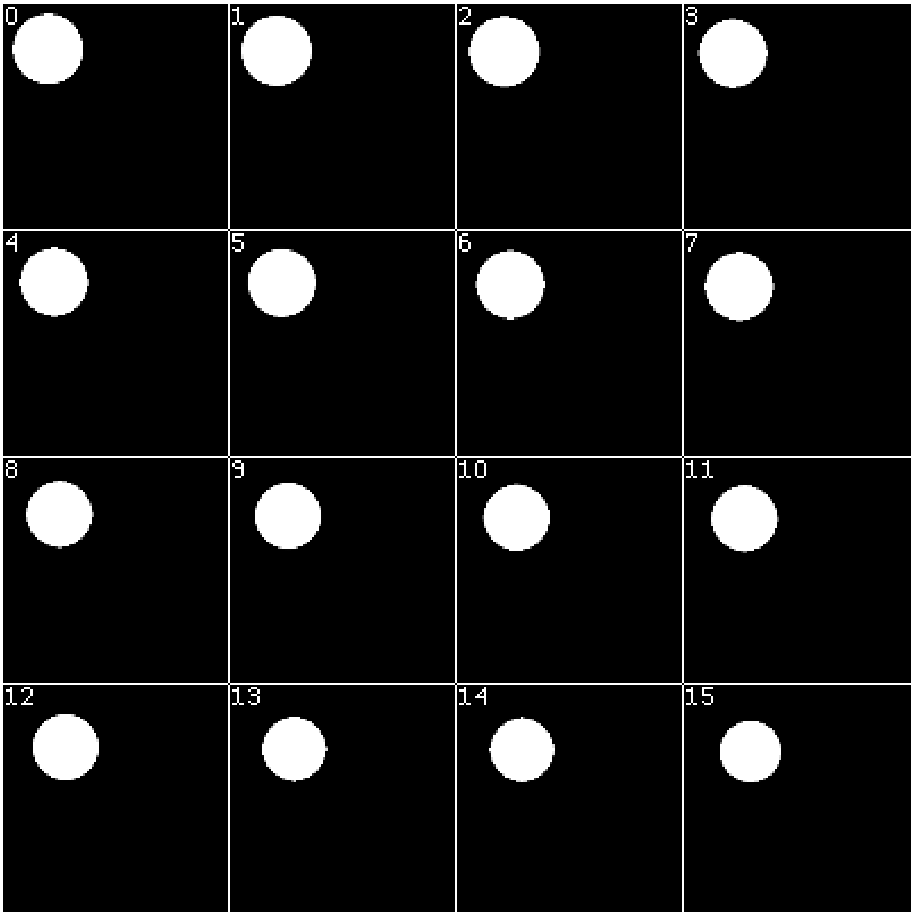
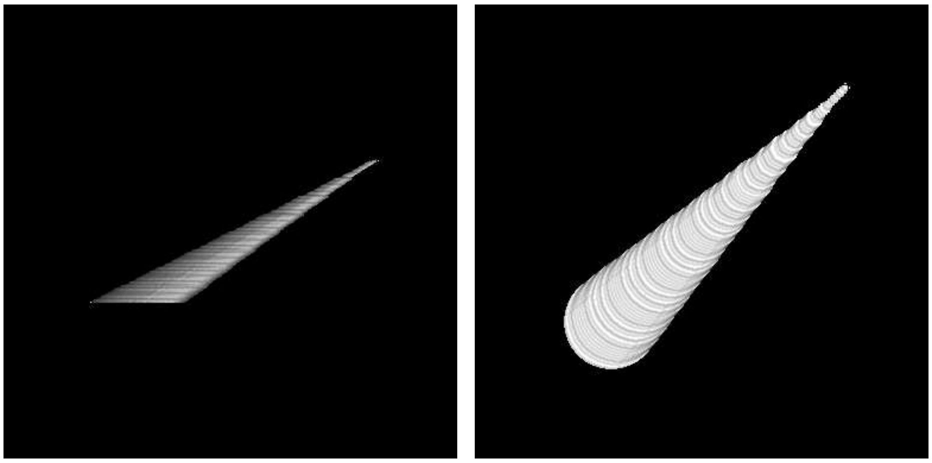


Fig. 8. The interpolated slices (1-14) between the first two slices (0,15) of the sloping cone volume.



(a) (b)

Fig. 9. Two visualizations of the interpolated sloping cone volume.

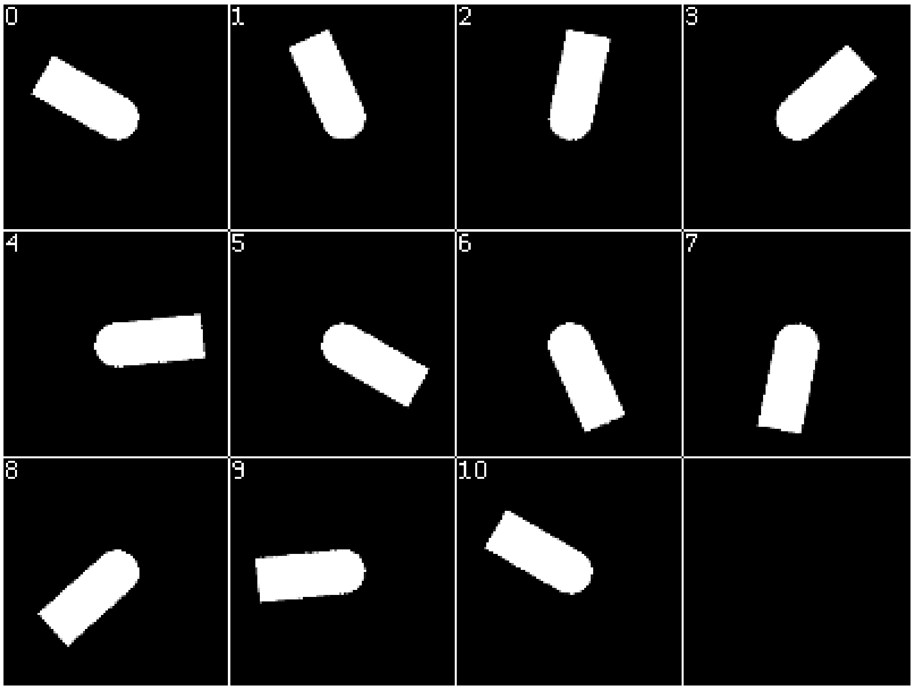


Fig. 10. The original slices of the staircase-like volume.

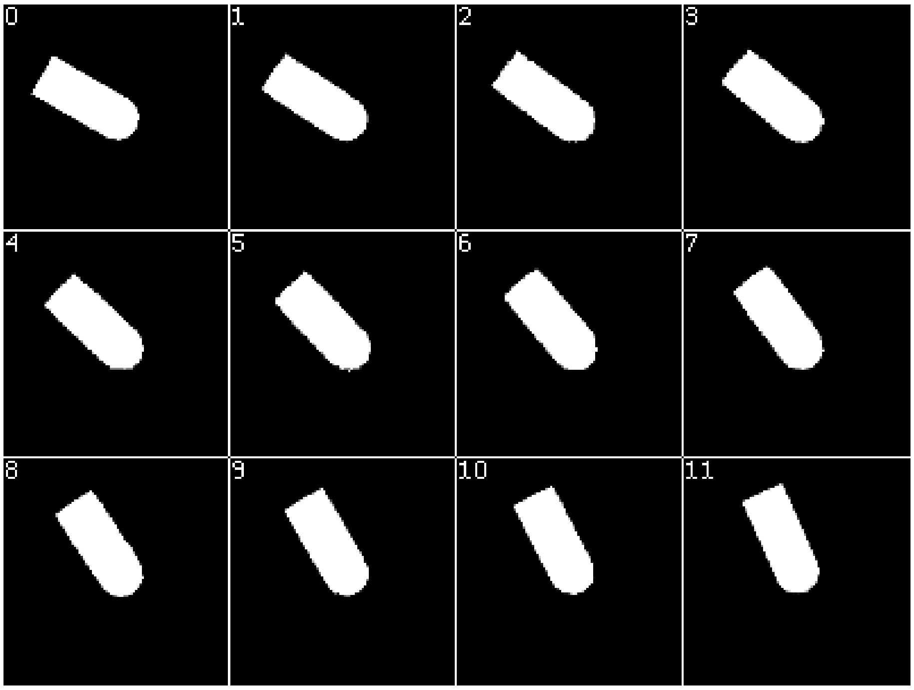
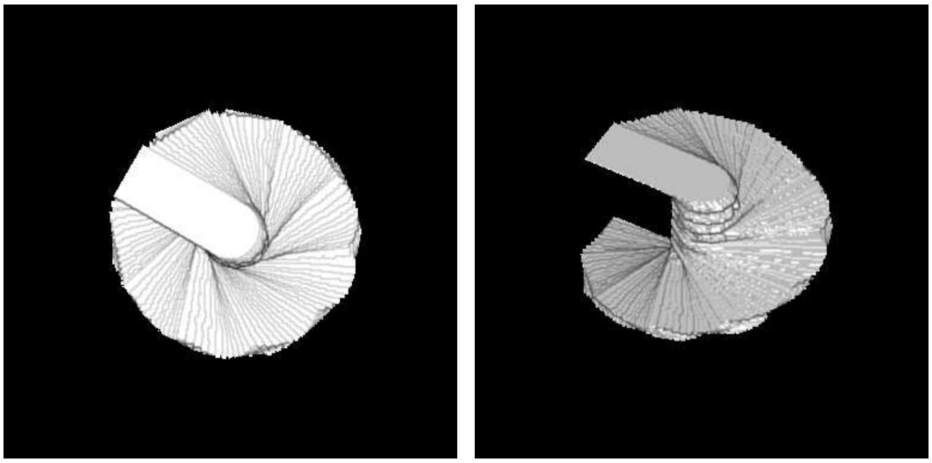


Fig. 11. The interpolated slices (1-14) between the first two slices (0,15) of the staircase-like volume.



(a) (b)

Fig. 12. Two visualizations of the interpolated staircase-like volume.

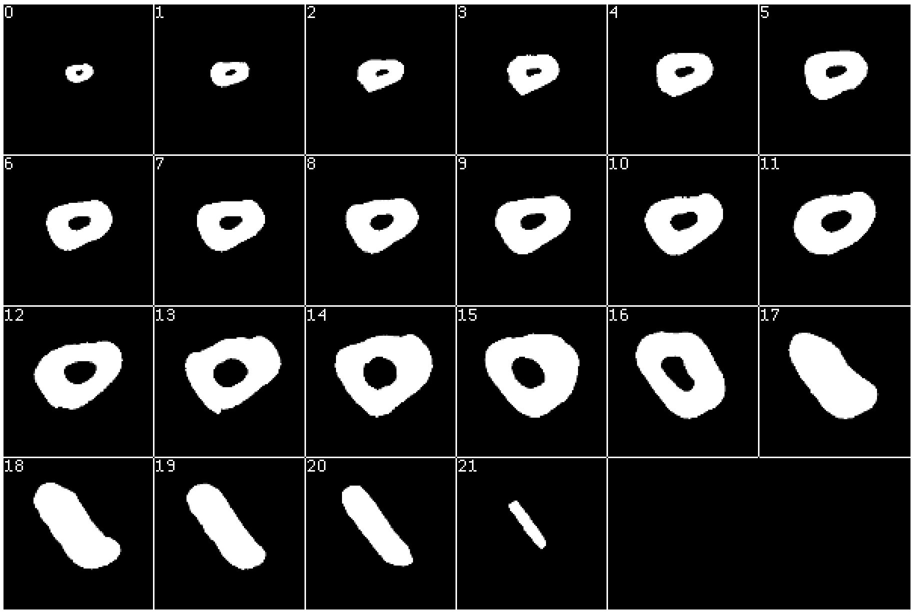
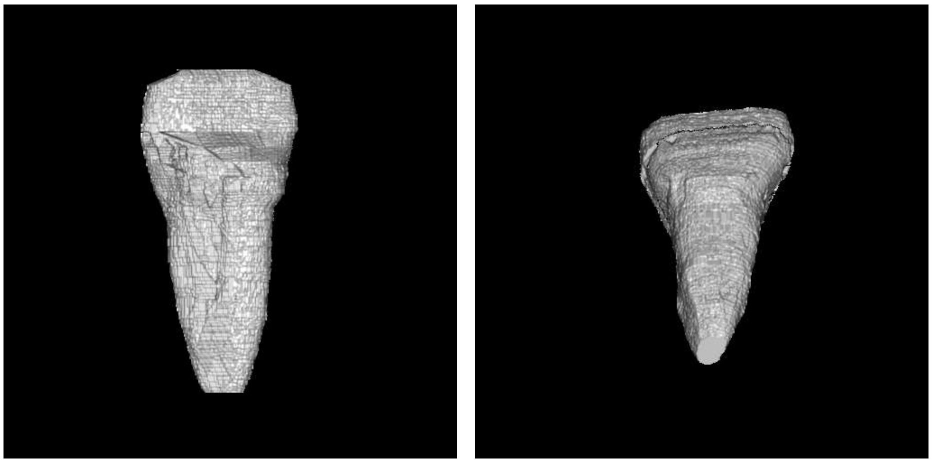
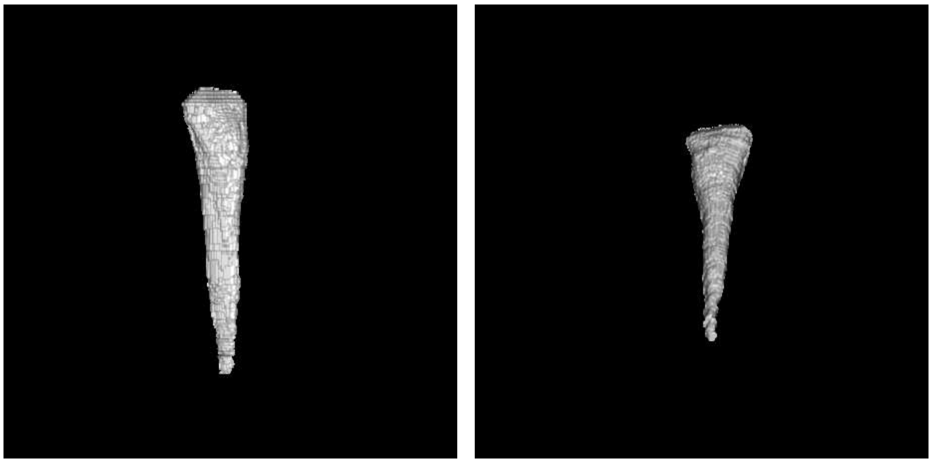


Fig. 13. The original slices of the incisor volume.



(a) (b)

Fig. 14. Two visualizations of the outer surface of the interpolated incisor volume.



(a) (b)

Fig. 15. Two visualizations of the root canal, the inner surface of the interpolated incisor volume.

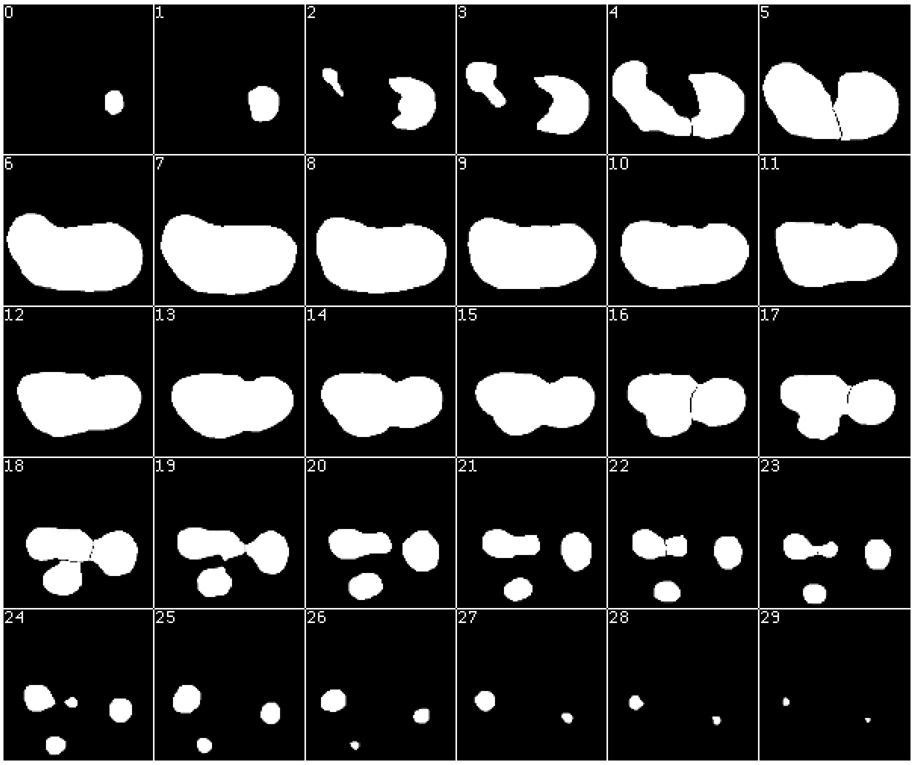


Fig. 16. The original slices of the molar volume.

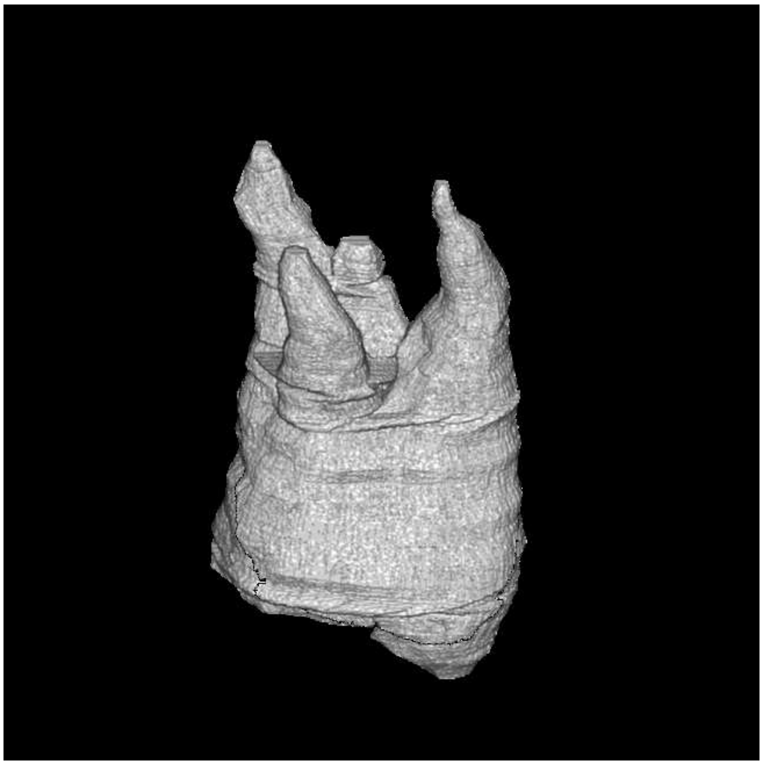


Fig. 17. A visualization of the outer surface of the interpolated molar volume.