Hole filling in 3D volumetric objects

Marcin Janaszewski a,*, Michel Couprie b, Laurent Babout a

a Computer Engineering Department, Technical University of Łódź, Stefanowskiego 18/22, 90-924 Łódź, Poland
b Université Paris Est, LABINFO-IGM, A2SI-ESIEE 2, boulevard Blaise Pascal, Cité DESCARTES BP 99 93162 Noisy le Grand CEDEX, France

ARTICLE INFO

Article history:
Received 27 October 2008
Received in revised form 14 April 2010
Accepted 18 April 2010

Keywords:
Hole filling
Hole closing
Skeleton
Digital topology
Crack propagation images

ABSTRACT

The construction of hole filling (or hole segmentation) method for 3D volumetric images is a new challenging issue in computer science. It needs a geometrical approach since from a topological point of view 3D holes (tunnels) are not well-delimited subsets of three dimensional space. In this paper, the authors propose an original, efficient, flexible algorithm of hole filling for volumetric objects. The algorithm has been tested on artificial objects and very complicated crack propagation tomography images. The qualitative results, quantitative results and features of proposed approach are presented in the paper. According to our knowledge it is the first algorithm of hole filling for volumetric objects.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

In material science a hole filling algorithm can contribute to the quantification of damage phenomena that can help to understand and further optimise the resistance of the material to damage. Indeed, computed tomography allows reconstruction of 3D volumetric images which reveal microstructural details of a material [1]. Experiments on material damage processes can result in 3D images of crack propagation inside a material and interaction with the microstructure. For instance, in the case of stress corrosion cracking in austenitic stainless steel [2], a crack usually propagates through the weakest areas of the material sample, so if it meets the areas of stronger resistance, it tends to bypass the region by branching, subsequently rejoining and creating a hole of the crack path (see Fig. 1). Therefore, each hole in a crack corresponds to a material portion (also called ligament) of stronger resistance. The tendency for crack bridging is a measure of crack propagation resistance, and there is a need to quantify the distribution of ligaments. Segmentation of such portions is being done manually by visual inspection of 3D images. Unfortunately, such a strategy is time consuming and potentially unreliable [3], so an automatic system of hole segmentation for volumetric images seems to be a valuable alternative. Other possible medical applications may consist in filling small noisy holes in 3D tomographs of human organs or segmentation of important holes in order to plan a surgery.

From the topological point of view the presence of a hole (or tunnel) in an object is detected whenever there is a closed path which cannot be transformed into a single point by a sequence of elementary local deformations inside the object [4]. Based on this definition a sphere has no hole, a torus has one hole and a hollow torus has two holes. Moreover, it is important to emphasise the difference between holes and cavities. The latter are hollows in an object, or more formally, bounded connected components of the background. Unfortunately, from topological point of view a hole is not a well-delimited subset of 3D space so it cannot be easily segmented. Therefore, taking into account the topological point of view we study holes from a geometrical point of view, by considering a notion of the thickness of an object and trying to interpolate the thickness in the corresponding hole filling volume.

The hole filling task is related to a hole closing task, which consists of replacing an object X that has some holes by an object Y such that X ⊆ Y, Y has no holes and Y is minimal in some sense for these properties (see Fig. 2(d)). It is worth mentioning the difference between hole filling and hole closing. The hole closing volume represented in red colour in Fig. 2(d) is one voxel thick, independently on the thickness of the corresponding object while the hole filling volume (see Fig. 2(b)) exactly matches the thickness of the corresponding object. According to the above, there are many articles dealing with geometric hole closing (void closing) in outer surface of a 3D object e.g. [5–8] and only one contribution [9] about algorithm for hole (tunnel) closing not only in the outer surface of 3D objects but also in solid volumetric objects whose thickness is greater than one (see Fig. 2(d)).
The article presents an original algorithm of hole filling for volumetric images; according to our knowledge, no such algorithm has been published until now. As the first step the algorithm utilises a filtered Euclidean skeleton [10] which generates homotopic, object centred, surfacic skeleton whose voxels represent thickness of the input object. On the second step all holes in the skeleton are closed with the use of a modified version of the hole closing algorithm [9]. The third step consists of propagation of the object thickness inside the patches which close holes and finally the last step corresponds to the dilation of the patches to obtain the thickness of the input object.

2. Related works

We divided related works into three groups. The first group represents methods of 3D geometric hole closing (void closing) in a triangular mesh—a set of vertices and a set of oriented triangles that join these vertices. Assuming that the corrected mesh should be a closed manifold, geometric holes (voids) are easily detected in such objects by searching for discontinuities (i.e., boundary edges—edges adjacent to only one triangle). These methods usually close voids in triangular mesh of a 3D object’s outer surface obtained as a result of laser scanning. Voids in scanned surfaces are result of occlusion—recesses are too deep to observe the entire surface. This group contains works which apply geometric approach to close voids in a triangulated object which result with smooth continuation of surface into missing areas e.g. [5,11,12]. Some methods e.g. [13] use global shape information provided by templates in order to solve difficult cases where smooth continuation of surface into voids does not give correct reconstruction. In case of volumetric images we are working on, for example the solid torus in Fig. 2(a), even if we convert it to an object which outer surface is represented by a triangular mesh there is not any boundary edge in the mesh so the cited methods do not find any void to be closed. On the other hand, consider a (thin) sphere with a small disc removed. From our point of view there is no hole (tunnel), but these methods will try to build a patch to replace the missing disc.

The second group consists of methods which perform void closing on the outer surface of an object represented as a cloud of points. 2D scans produced by a scanner are combined into a 3D unorganised set of points and voids are recognised as regions of the visible physical surface not covered with sample points. One class of voxel image methods interpolates the original samples using balls or alpha shapes [14]. The next class of methods iteratively evolves a surface or a special 3D signed function, which zero level set is the intended surface until it approximates the data e.g. [6,15]. Some of these methods e.g. [6] deal with very complicated voids from geometrical and topological point of view. In a classification based approach each voxel is classified either to inside or to outside of an input object based on distances of the voxel to different fragments of the outer surface of the input object e.g. [16]. In [17] the classification is improved by minimisation of the repaired outer surface area. There are also sophisticated approaches which do not assume generic smoothness priors on the missing surface and, thus, do not create a smooth patch that covers the void but they analyse the characteristics of the given surface and repairs the geometric holes in a context-sensitive manner. For example Breckon et al. [8] propose a method which not only realises geometric completion of missing surface but also restores surface relief. Algorithm presented in [18] restores large parts of outer surface based on matching with correct objects from a database. In [7,19]}
the multi-scale characteristics of a surface being repaired is analysed and voids are iteratively filled by copying the best matched patches from valid regions of the surface. Similar to algorithms from the first group, methods from the second group rather close geometric holes than fill them as they use one point thick objects to repair one point thick outer surface.

Interesting work which does not belong to both presented groups was done by Aktouf et al. [9]. They constructed fast algorithm which closes holes (tunnels) in voxel images taking advantage of local, topological characteristic of points and could be applied with success for objects represented with outer surface and volumetric, solid objects as well. We utilise the algorithm in our approach to hole filling.

To sum up, according to the above review, terminology introduced in previous section and the authors knowledge, there are many approaches to close voids in outer surface of 3D objects, there is one method for hole closing in volumetric objects [9] and no method for hole filling in volumetric object as all cited works do not restore thickness of an object into detected hole area.

3. Background

In this section, we recall some topological notions for binary images, which are necessary to understand the sequel of the paper. A more extensive review is provided in [4].

We denote by $\mathbb{Z}$ the set of integers, by $\mathbb{N}$ the set of nonnegative integers, by $\mathbb{R}_+$ the set of strictly positive integers and by $\mathbb{R}$ the set of positive real values. Moreover, assume that $E = \mathbb{Z}^3$.

A point $x \in E$ is defined by $(x_1, x_2, x_3)$ with $x_i \in \mathbb{Z}$. We consider the three neighbourhoods: $N_0, N_{1\beta}, N_{2\beta}$ defined by, $\forall x \in E$:

$$
N_0(x) = \{x \in E; |x_1-x_1'| + |x_2-x_2'| + |x_3-x_3'| \leq 1\} \\
N_{1\beta}(x) = \{x \in E; |x_1-x_1'| + |x_2-x_2'| + |x_3-x_3'| \leq 2\} \cap N_{2\beta}(x)
$$

The set $N_{1\beta}(x)$ is called the $n$-neighbourhood of $x$. We define $N_{1\beta}(x) = N_0(x)$, when $n=6,18,26$. Any point $y$ of $N_{1\beta}(x)$ is said to be a $n$-adjacent ($n=6,18,26$) to $x$. Now we consider the $n$-neighbour of $x$.

We set $N_{2\beta}(x) = N_{0\beta}(x) \cap N_{1\beta}(x)$ and $N_{1\beta}(x) = N_{0\beta}(x) \cap N_{2\beta}(x)$. Two points $x$ and $y$ are said to be strictly $n$-adjacent ($n=18,26$) if they are $N_{2\beta}$-adjacent. An $n$-path $\pi$ is a (possibly empty) sequence of points $x_0, x_1, \ldots, x_k$ with $x_i$-adjacent to $x_{i+1}$, for $i = 1, \ldots, k$. If $n$ is not empty, the length of $\pi$ is equal to $k$. If $x_0 = x_k$, $\pi$ is closed.

An object $X \subseteq E$ is said to be $n$-connected if, for any two points $x,y$ in $X$, there is an $n$-path in $X$ between these two points. The equivalence classes related to this relation are the $n$-connected components of $X$. Note that, if $X$ is finite, the infinite connected component of $\overline{X}$ (complement of $X$) is called the background, the other connected components of $\overline{X}$ are called the cavities. The set composed of all the $n$-connected components of $X$ is denoted $C_{n\beta}(x)$. Moreover, if we use an $n$-connectivity for $X$ we have to use another $m$-connectivity for $X$, i.e., the $6$-connectivity for $X$ is associated to the $18$- or the $26$-connectivity for $\overline{X}$ (and vice versa). This is necessary for avoiding connectivity paradoxes [4].

4. Filtered Euclidean skeleton

The first step of a hole filling algorithm consists in the generation of filtered skeleton originally proposed by Couprie et al. [10]. In this section we present only the overall survey of the algorithm. For more details the interested reader is requested to refer to [10]. The filtered skeleton algorithm is based on four, main, well-defined, discrete mathematics notions: exact square Euclidean distance, medial axis, discrete bisector function, and Euclidean skeleton.

Let us denote by $E$ the discrete space $\mathbb{Z}^3$. A point $x \in E$ is defined by $(x_1, x_2, x_3)$ with $x_i \in \mathbb{Z}$. Let $x \in E$, we denote by $d^2(x,y)$ the square of Euclidean distance between $x$ and $y$, that is, $d^2(x,y) = (x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2$. Let $Y \subseteq E$, we denote by $d^2(x,Y)$ the square of the Euclidean distance between $x$ and the set $Y$, that is, $d^2(x,Y) = \min\{d^2(x,y); y \in Y\}$. Let $X \subseteq E$ (the “object”), we denote by $D^2(X)$ the map from $E$ to $\mathbb{R}$ which associates, to each point $x$ of $X$, the value $D^2(x,X)$, the map $D^2(X)$ is called the (squared Euclidean) distance map of $X$. The squared Euclidean distance transform (SEDT) of an object in $E$ may be calculated by separable technique [24,25] which is linear both in space and time complexity.
Let $x \in E, r \in \mathbb{N}_+$, we denote by $B_r(x)$ the ball of (squared) radius $r$ centred on $x$, defined by $B_r(x) = \{y \in E, d^2(x, y) < r\}$. A ball $B_r(x) \subseteq X \subseteq E$ is maximal for $X$ if it is not included in any other ball included in $X$ (see Fig. 5). Notice that, for any point $x$ in $X$, the value $D^2_r(x)$ is precisely the radius of the maximal ball included in $X$ and centred on $x$.

The medial axis of $X$, denoted by $MA(X)$, is the set of the centres of all the maximal balls for $X$ (see Fig. 5). An efficient algorithm has been proposed to extract the exact Euclidean medial axis of a shape, from an exact squared Euclidean distance map and pre-computed look-up tables [26,27].

The thickness of an object $X$ in point $x \in MA(X)$ is defined as a radius of the maximal ball centred in $x$.

Let us recall the notion of bisector function. Let $E$ be either $\mathbb{R}^3$ or $\mathbb{Z}^3$, let $S$ be a nonempty subset of $E$, and let $x \in E$. The projection of $x$ on $S$, denoted by $\Pi_S(x)$, is the set of points $y$ of $S$ which are at minimal distance from $x$; more precisely, $\Pi_S(x) = \{y \in S, \forall z \in S, d(y, x) \leq d(z, x)\}$. The bisector angle of a point $x$ in $X$ can be defined,
in the continuous framework, as the maximal unsigned angle formed by \( x \) (as the vertex) and any two points in the projection of \( x \) on \( X \) [28,29]. In particular, if \( \#P_{\langle x \rangle} = 1 \), then the bisector angle of \( x \) is zero (see Fig. 5). The bisector function of \( X \) is the function which associates to each point \( x \) of \( X \), its bisector angle in \( X \). The definition has been also adapted to the discrete case in [29,30,31]. In the presented approach the authors utilise the notions of discrete bisector angle and discrete bisector function introduced in [10]. Let us present the definitions for 2D lattice of points for simplicity. The extended projection of \( x \in X \) on \( X \), denoted by \( P_{\langle x \rangle} \), is the union of sets \( P_{\langle x \rangle} \) for all \( y \) from \( N(x) \) such that \( d^2(y,x) \leq d^2(x,X) \). The (discrete) bisector angle of \( x \) in \( X \), denoted by \( \theta_x \), is the maximal unsigned angle between the vectors \( \overrightarrow{x,y} \) and \( \overrightarrow{x,z} \), for all \( y,z \in P_{\langle x \rangle} \) (see Fig. 6). In particular, if \( \#P_{\langle x \rangle} = 1 \), then \( \theta_x = 0 \). The (discrete) bisector function of \( X \), denoted by \( \theta_X \), is the function which associates to each point \( x \) of \( X \), its discrete bisector angle in \( X \). In [31], Attali and Montanvert have shown that an efficient filtering of a skeleton can be performed by removing two kinds of medial axis points: those which correspond to maximal balls of small radius, and those which have a small bisector angle. It is worth emphasising that we do not use filtering by the radius because there is a ball which is maximal for \( r \) of small radius, and those which have a small bisector angle. It is worth emphasising that the bisector function \( \theta_X \) is extensively studied by Couprie et al. in [10]. It is worth emphasising that the bisector function \( \theta_X \) needs only to be computed on points of the medial axis.

Based on procedures and notions defined earlier the parameterised, filtered Euclidean skeleton procedure [10], denoted by FES, can be presented as follows:

**FES (Input \( X, T_b, \) Output \( Z \))**

1. \( D_X^2 \leftarrow \text{SEDT}(X) \)
2. \( M \leftarrow \text{MedialAxis}(X,D_X^2) \)
3. \( \theta_X \leftarrow \text{DiscreteBisector}(X,M) \)
4. \( Y \leftarrow \{ x \in M : \theta(x) \geq T_b \} \)
5. \( Y' \leftarrow \{ x \in M : N(x) \cap M \neq \emptyset \} \)
6. \( Z \leftarrow \text{EuclideanSkeleton}(X,D_X^2,Y') \)

In the above procedure the third parameter of \text{EuclideanSkeleton} algorithm represents a set which constrains the generated skeleton \( Z \). The exact squared Euclidean distance transform (SEDT) may be computed in linear time, both in 2D and 3D, see [24,25]. It only has to be computed once and is used by procedures \text{MedialAxis} and \text{EuclideanSkeleton}. Procedure \text{MedialAxis} can be efficiently implemented in 2D and 3D thanks to the method of Coeurjolly [32] or that of Rémy and Thiel [27]. The computation cost of the \text{DiscreteBisector} procedure which calculates the Voronoi diagram, extended projection function and bisector function has been extensively studied by Couprie et al. in [10]. It is worth emphasising that the bisector function \( \theta_X \) needs only to be computed on points of the medial axis.

Fig. 7 presents an example of FES result when applied to a 2D image. Fig. 7(d) presents Euclidean skeleton of the input object without any filtration. Note that the skeleton is too thick in some areas and there are also small unwanted branches. Fig. 7(c) shows the main drawback of discrete bisector function from the skeleton pruning point of view. The ending points of branches which are usually border points of the input object have very high discrete bisector angle (\( \theta \)), very often maximal value \( \theta = \pi \). For example the end points of three branches in the Fig. 7(c), e.g. upper left, bottom left and bottom right areas, have \( \theta = \pi \). It prevents from effective filtration of medial axis by thresholding the discrete
5. Hole closing algorithm

The second step of hole filling algorithm consists in application of a modified version of the hole closing algorithm proposed by Aktouf et al. [9]. The original version of the algorithm is linear in time and space complexity and may be presented as follows: First it computes a bounding box \( Y \) which has no cavities and no holes and which contains the input object \( X \). Then it iteratively deletes points of \( Y \times X \) which are border and not 2D isthmuses (see Section 3). If a border point \( p \) of \( Y \times X \) is also a 2D isthmus it can be deleted only if its distance \( d(p, X) \) is greater than a predefined parameter holesize. The last condition results in an effect that only holes of size less or equal to the parameter holesize are closed. The deletion process is ordered by a priority function which is defined as Euclidean distance from \( X \). The algorithm repeats this procedure until stability. The pseudocode of the hole closing algorithm (HCA) can be presented as follows:

\[
\text{HCA}(\text{Input } X, \text{holesize}, \text{Output } Y)
\]

Generate a bounding box \( Y \) which contains \( X \)
Repeat until no point to delete:
- Select a point \( p \) of \( Y \times X \) such that: \( T_m(p, Y) = 1 \)
- or such that: \( T_m(p, Y) = 2 \) and \( d(p, X) > \text{holesize} \)
- which is at the greatest distance from \( X \)
- \( Y := Y \setminus p \)
Result: \( Y \)

An example of the algorithm result when applied to a torus is presented in Fig. 2(d).

The first problem connected with hole closing algorithm is that it closes not only holes in an input object but also cavities (bounded connected components of complementary of an object). From the hole filling point of view it is important to fill only holes, not cavities. To ensure that hole closing algorithm will only deal with holes and not cavities, we begin with a standard cavity filling procedure. The first step consists in generation of a 3D, 1 voxel thick, frame around the input image \( X \) and assign the value 0 to each voxel of the frame except the voxel of all coordinates equal to 1 which is set to the value 1. In this way we obtain image \( X_0 \). Then we perform geodesic dilation of \( X \) under \( X_0 \) and inverse the result.

The second, more significant, drawback of hole closing algorithm, from the hole filling point of view, is that the shape of a hole closing patch may be significantly influenced by irrelevant branches which are close to the hole. Such a situation is presented in Fig. 8. An input object, which can be treated as a result of filtered Euclidean skeleton algorithm (see Section 4) is presented in Fig. 2(d). There is one big hole in the middle of the object and a thin branch above. Fig. 8(b) presents the result of hole closing algorithm. The hole closing patch, which is represented with red colour, goes up to the branch, so it does not correspond to the “geometry of the hole”, which leads to wrong hole filling. In this case we expect that the hole closing patch is flat as the object around the hole is flat. To overcome this problem we propose a 4-step modification of hole closing algorithm. The first step of the modification consists in application of original hole closing algorithm. As a result we obtain an object with the hole closed but the hole closing path is not correspond to the “geometry of the hole”, which leads to wrong hole filling. In this case we expect that the hole closing algorithm will only deal with holes and not cavities. From the hole filling point of view it is important to fill only holes, not cavities.
points from the input object until stability. In this way it deletes all branches of an input object. The last, fourth step consists in application of hole closing algorithm on the hole contour. As the hole contour does not contain any branch, the hole closing patch is not influenced by any branch and matches the geometry of the corresponding hole (see Fig. 8(f)).

In some extreme cases, where a skeleton of a crack contains branches with thickness greater than one voxel, the procedure presented above fails. In such a case it is better to apply several iterations of erosion which preserve topology of an input object, followed by the same number of iterations of a geodesic dilation over the set of eroded voxels.

The modified hole closing algorithm, denoted by HCA+, can be computed in quasi-linear time as UHS has quasi-linear time complexity and all other steps have linear time complexity [9,33]. HCA+ procedure can be presented in the following pseudocode:

\[
\text{HCA}^+ (\text{Input } X, \text{Output } Z) \\
01. \quad X_{cf} \leftarrow \text{CavitiesFilling}(X) \\
02. \quad Y \leftarrow \text{HCA} (X_{cf}) \\
03. \quad Y_{dil} \leftarrow \text{GeoDilat}(Y,X) \\
04. \quad C \leftarrow Y_{dil} \cap X \\
05. \quad Z \leftarrow \text{HCA}(C)
\]

Procedure \text{GeoDilat}(Y,X) performs one iteration of geodesic dilation of an object Y over object X and returns dilated Y.

The comparison between HCA and HCA+ applied to a volumetric portion of the crack which represents the branch problem is presented in Fig. 9.

6. Hole filling algorithm

In this section we propose the original hole filling algorithm, based on notions and procedures presented earlier:

\[
\text{HFA} (\text{Input } X, \theta, \text{Output } Z) \\
01. \quad S \leftarrow \text{FES}(X,\theta) \\
02. \quad P \leftarrow \text{HCA}^+ (S) \\
03. \quad P' \leftarrow \text{MeanFilter}(S,P) \\
04. \quad B \leftarrow \text{DilationByBalls}(P') \\
05. \quad Z \leftarrow B \cap X
\]

\text{MeanFilter} is a procedure which realises propagation of an object thickness represented by values of its filtered skeletal voxels, into hole closing patches. The algorithm, in each iteration, calculates a new value for each voxel, from a hole closing patch, as an average value of voxels from its neighbourhood which belong either to the

**Fig. 8.** Example result of each step of HCA+. (a) An isosurface of an input object. (b) The result of HCA. The hole closing patch is represented with red colour. Notice that the patch goes up to the branch situated over the hole. (c) The result of geodesic dilation of the patch over the input object. The intersection of dilated patch and input object, called hole contour, is visualised with grey colour. (d) Visualisation of the hole contour, where one can see all its details. (e) The result of ultimate homotopic skeletonisation algorithm applied to the hole contour. Note that, the branch has been deleted and topology of the hole contour has been preserved. (f) Result of HCA applied to hole contour superimposed to the input object. Note that, the hole closing patch (red colour) is not influenced by the branch. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Fig. 9.** Exemplar results of HCA+ and HCA where hole closing patches are represented with red colour. (a) A fragment of isosurface of a crack with a hole inside and a branch above the hole. (b) The result of HCA. (c) Enlarged view of the most interesting fragment of the image (a). It is clearly seen that the hole closing patch goes up to the branch. (d) The result of HCA+. (e) Zoomed view of the same area as in (c) to show that the branch does have no more influence on the geometry of the patch. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
hole closing patch or to the filtered skeleton. The algorithm stops when no significant changes occur during an iteration.

Fig. 10 presents an exemplar result of the mean filter procedure. The filtered skeleton of a crack object is presented in Fig. 10(a). Note that, voxel values represented by different colours are small, which means that the corresponding crack is very thin at this location. Fig. 10(b) shows a rendering of hole closing patches superimposed on the skeleton. Notice that, the hole closing patches exactly close all holes in the skeleton, but their voxel values do not represent the thickness of the crack. Fig. 10(c) shows that the propagation of crack thickness into the hole closing patches has been carried out properly.

The last procedure: \textit{DilationByBalls} for each voxel \(x\) of its input image generates a ball, centred in \(x\), of radius equal to the value of voxel \(x\).

Finally we obtain the hole filling algorithm which has the following properties:

- It is based on well-defined mathematical notions and exact algorithms like: medial axis, bisector function, exact Euclidean distance, Euclidean skeleton.
- It is easy to use: only needs one parameter to be tuned (bisector threshold).
- It is efficient: since most of its steps are optimal in time and space complexity.

According to Couprie et al. [10] it is possible to implement efficient \textit{FilteredSkeleton} procedure (see also Section 4). \textit{ModifiedHoleClosing} can be computed in quasi-linear time and linear space complexity. \textit{MeanFilter} is not optimal, from computational cost point of view. Probably the problem of interpolation of object thickness inside hole closing patches could be solved more efficiently with, for example, a minimal surface approach [34]. On the other hand, the mean filter is very intuitive and easy to implement. Moreover, it is applied to hole closing patches, which are small compared to the size of an input image. Therefore, the algorithm does not have significant influence on computational cost of the whole hole filling algorithm. The last procedure: \textit{DilationByBalls} takes linear time and space.

7. Results

Qualitative examples of the results of hole filling algorithm are presented in Figs. 11–13. It can be observed that in the case of the

![Fig. 10. Visualisation of an exemplar result of the mean filter procedure applied to a filtered skeleton of a crack propagation with closed holes. The colour bar represents different voxel values. (a) Filtered skeleton of a crack object for bisector threshold equal to 2.8. (b) Filtered skeleton of a crack object with superimposed isosurface of corresponding hole closing patches. (c) The result of the mean filter algorithm added to the filtered skeleton. It can be seen that values of skeletal voxels which are adjacent to holes have been properly propagated inside the corresponding hole closing patches. For example the mean value of voxels from the largest hole closing patch is equal to 2.89 and is similar to the mean value of voxels from the filtered skeleton of the crack which were used as propagation seeds: 3.04. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)](image)

![Fig. 11. (a) Visualisation of a torus with non uniform thickness. (b) Result of hole filling algorithm applied to the torus. Hole filling volume is represented with red colour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)](image)
non uniform torus (Fig. 11) and the chain with different elements size and thickness (Fig. 12), all holes are closed and the thickness of the corresponding hole filling volumes match the thickness of these objects. In the case of a crack path (Fig. 13) all holes are closed but it seems that the thickness of the hole filling volume does not correspond, everywhere, to the thickness of the crack. In this application, measurement of the thickness of the hole is important, since it is a signature of the work undergone by the ligament to resist the crack growth. Note that, Fig. 13(d) represents a magnified part of Fig. 13(b) where one can see that the thickness of the hole filling volume does not correspond to the thickness of the crack. Thorough analysis of such areas reveals that the thickness of the hole filling volume is correct and the visual mismatch is the result of discretisation, which does matter in case of thin objects. The phenomenon may be explained based on Fig. 14 which shows results of consecutive steps of hole filling.

Fig. 12. Visualisation of a chain, whose links have different thicknesses. Hole filling volume is represented with red colour. (a) An input object. (b) Result of hole filling algorithm applied to the chain (left side view of the object). (c) Result of hole filling algorithm applied to the chain (right side view of the object). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 13. Visualisation of a result of hole filling algorithm, applied to a crack path inside a material (material is represented by the background), with bisector threshold equal to 2.7. (a) Crack. Note that, there are some holes inside the crack to be filled. (b) The crack with holes filled. Hole filling volume is represented in red colour. (c) Upside down view of the crack with holes filled. (d) Zoomed part of the image. (b) It appears in the visualisation that the hole filling volume does not exactly correspond to the thickness of the crack. The region of interest is encircled. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
algorithm applied to one 2D slice of a 3D image. Input grid of voxels is presented in Fig. 14(a). Object’s voxels are visualised in grey colour, white voxels represent complementary of the object. Note that, there is a fragment of a crack between two components of the object. One can see from Fig. 14(f) a difference between the hole filling volume thickness and thickness of the object, but in fact the thickness of the hole filling volume is 2 voxels which is equal to the thickness of the object. In some cases, hole filling may not match with what would be perceived from visual inspection. It is therefore a problem which is linked with how the data can be interpreted, but is not linked with errors in the presented method.

Quantitative analysis of results of the hole filling algorithm are presented in Figs. 15 and 16. The results were obtained for volumetric portion of the crack of size 370 × 380 × 290 voxels. The HFA procedure detected and filled 226 holes in the image. Fig. 15 presents, the scattergram of the mean thickness of the hole filling volume against the mean thickness of the skeletal voxels of the crack adjacent to the hole filling volume. The linear trend of real data is slightly larger than the theoretical one. What is more, the larger the thickness of the crack the larger the difference between compared thicknesses. The mean and the standard deviation of average thicknesses of hole filling volumes are equal to 2.29 and 1.36 respectively. The mean value of the absolute difference between the crack and the corresponding hole filling thicknesses is equal to 0.11 for a standard deviation of 0.24. The dilation by balls procedure implies that mean filter values are rounded to the closest integer. This operation implies, in the present case study, that the difference greater or equal to 1 occurred 36 times which is around 16% of total number of hole filling volumes. The difference equal to 2 occurred only once for the hole filling volume mean of thickness equal to 8. There is no difference larger than 2 as it can be depicted in Fig. 15. Moreover, note that most hole filling volumes have mean thickness equal to 1 or 2 but there is a significant number of volumes (around 30%) with mean thickness larger than 3. Therefore, it can be concluded that the crack near a hole is usually thin (1–2 voxel thick) but there is a significant number of holes for which the corresponding crack thickness near the hole is large enough to justify the application of the crack thickness restoration inside the hole.

Fig. 16(a) presents the mean thickness of all hole filling components in the input image as a function of bisector threshold used in the filtered Euclidean skeleton algorithm. Fig. 16(b) presents the total volume of all hole filling components in the input image as a function of the bisector threshold. The figure presents the evolution between 2 and 3 because it has been verified that values of the bisector threshold smaller than 2 result in very weak skeleton pruning. One can see that both evolutions have ascending trend. One can also deduce from these figures that the total volume grows with the mean thickness. However, it is hazardous to judge from these evolutions which is the best bisector threshold. Nevertheless, they show that filtering of a skeleton by bisector threshold is stable since small changes of $T_b$ result in relatively small changes of the average thickness and total volume of hole filling components. If $T_b$ is too small, the skeleton of an input object is not enough pruned, hence noisy voxels from the surface of the object can potentially form cusp-like shapes that may result in a too thin hole filling volume. On the other hand, if the parameter is too large then an input object skeleton is over-pruned, consequently the corresponding hole filling volume is usually too large and partly spread over the input object. Therefore few tries of hole filling algorithm (starting from $T_b=2$ and gradually increasing it) followed by visual inspection of results are usually needed to set the proper bisector threshold for an input object.

8. Conclusions

In the paper the authors have presented a flexible, efficient, algorithm of hole filling for volumetric images, based on well defined mathematical notions. The algorithm has been tested on artificially constructed images and on images of a crack inside a material, for which it is an intended application. The quantitative analysis of the results in terms of comparison between average hole thickness and corresponding average thickness of surrounding crack portion followed by thorough inspection of some regions of particular interest confirmed that the thickness of generated hole filling volumes correspond to the thickness of input objects. Moreover, the algorithm has only one parameter—bisector threshold which is easy to tune by performing several tries for an input image.

The study of related works draws to a conclusion that there is only one significant approach which suppresses holes in volumetric objects [9]. But the algorithm rather closes holes than fills them as it uses one voxel thick objects for reparation. Therefore according to our knowledge the presented approach is the first algorithm of hole filling for volumetric images.
Fig. 15. Comparison of mean thickness of hole filling component with mean thickness in voxels of a crack from which the thickness was propagated by the use of mean filter, for each component. The plot has been prepared for typical 3D crack propagation image of size $370 \times 380 \times 290$ voxels with 226 holes. The black dash line represents ideal trend where mean thicknesses of hole filling components are equal to mean thicknesses of corresponding crack voxels. The grey line represent trend of presented data which is slightly different from the ideal one.

Fig. 16. Analysis of bisector threshold influence on hole filling algorithm results for a typical crack propagation 3D image of size $370 \times 380 \times 290$ voxels with 226 holes. (a) Mean thickness of all hole filling components as a function of bisector threshold. (b) Total volume of all hole filling components as a function of bisector threshold.
Work is now in progress to characterise the shape of the holes using region-based shape representation technique. The full set of data that characterise the shape, size, thickness and orientation of the holes can be compared with the data extracted from diffraction computed tomography [DCT] [35]. The technique, performed prior to CT scans on the same sample, provides valuable information about the crystallographic properties of the material, including grain boundaries and bridge ligaments [3]. A unique comparison is under way in collaboration with the University of Manchester that provides both data from CT and DCT. This is partly only possible thanks to the hole filling algorithm described in this paper.

Acknowledgements

The work is funded by the European Community's Sixth Framework Programme—Marie Curie Transfer of Knowledge Action (DENIDIA, Contract no. MTKD-CT-2006-039546). The work reflects any use that may be made of the information contained therein.

The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST), Eurographics Association, Aire-la-Ville, Switzerland, 2005, article no. 23.

The work is funded by the European Community’s Sixth Framework Programme: Technology Platform on Quality of Life (DENIDIA, Contract no. MTKD-CT-2006-039546). The work reflects any use that may be made of the information contained therein.

References


About the Author—MARCIN JANASZEWSKI has been graduated from the Technical University of Łódź, Poland, Faculty of Technical Physics, Applied Mathematics and Computer Science in 1989. He received his Ph.D. degree from the University of Science and Technology, Krakow, Poland in 2003. He has been an employee of the Computer Engineering Department, Technical University of Łódź on a position of assistant professor since 2005. His current research interests include image analysis and pattern recognition.

About the Author—MICHEL COUPIE received his Ingénieur’s degree from the École Supérieure d’Ingénieurs en Électronique et Électrotechnique, Paris, France, in 1985 and the Ph.D. degree from the Pierre & Marie Curie University, Paris, France, in 1988. He has been working in ESIEE since 1988. He has been a professor since 2005. His current research interests include image analysis and discrete mathematics.

About the Author—LAURENT BABOUT received his M.Sc. and Ph.D. degrees in Materials Science and Engineering from the Institut National des Sciences Appliquées, Lyon, France, in 1998 and 2002, respectively. Currently, he is an assistant professor at the Computer Engineering Department of the Technical University of Łódź, Poland. His main research interests include 3D image processing and analysis for materials science applications.