Making 3D Binary Digital Images Well-Composed

Marcelo Siqueira[a], Longin Jan Latecki[b] and Jean Gallier[a]

[a]University of Pennsylvania, Philadelphia PA 19104, USA;
[b]Temple University, Philadelphia PA 19122, USA

ABSTRACT

A 3D binary digital image is said to be *well-composed* if and only if the set of points in the faces shared by the voxels of foreground and background points of the image is a 2D manifold. Well-composed images enjoy important topological and geometric properties; in particular, there is only one type of connected component in any well-composed image, as 6-, 14-, 18-, and 26-connected components are equal. This implies that several algorithms used in computer vision, computer graphics, and image processing become simpler. For example, thinning algorithms do not suffer from the irreducible thickness problem if the image is well-composed, and the extraction of isosurfaces from well-composed images using the Marching Cubes (MC) algorithm or some of its enhanced variations can be simplified, as only six out of the fourteen canonical cases of cube-isosurface intersection can occur. In this paper, we introduce a new randomized algorithm for making 3D binary digital images that are not well-composed into well-composed ones. We also analyze the complexity and convergence of our algorithm, and present experimental evidence of its effectiveness when faced with practical medical imaging data.

Keywords: Well-composed images, digital topology, randomized algorithms

1. INTRODUCTION

A special class of 3D digital images, called 3D well-composed images, has been defined in Ref 1. A well-composed image enjoys very useful topological and geometric properties, such as the fact that there is only one connectedness relation on points of the image. This implies that several basic algorithms in computer vision, computer graphics, and image processing become simpler when dealing with well-composed images. For instance, thinning algorithms do not suffer from the irreducible thickness problem if the image is well-composed.[2] Thinning well-composed gray-scale images leads not only to theoretically better understood algorithms but also to practically relevant thinning and segmentation algorithms.[3] The property of well-composedness is used as an important tool for proving theoretical results in digital topology.[4] More recently, we showed in Ref. 5 that the extraction of isosurfaces from well-composed images using the Marching Cubes (MC) algorithm[6] or some of its enhanced variations[7,8] can be simplified. On the other hand, if an image lacks the property of being well-composed, then the digitization process that gave rise to it is not topology-preserving, as the results in Ref. 9 show that if the resolution of the digitization process is fine enough to ensure preservation of topology, then the image must be well-composed. This fact has motivated the development of algorithms for "repairing" images that are not well-composed.[4,10]

The idea behind a repairing algorithm is to change the color assigned to some points of the image, so that the resulting image is well-composed. There is no guarantee that the resulting well-composed image is the same as the one that would be obtained by a topology-preserving digitization process. However, if the repairing algorithm changes the color assigned to a few points of the image only, then the image and its well-composed counterpart will be very similar, which may be entirely satisfactory for several image-based applications that can benefit from using well-composed images.

Here, we introduce a new repairing algorithm for making 3D *binary* digital images well-composed. Our algorithm changes the color assigned to background points of the input image only, so that they become foreground

Further author information:
Marcelo Siqueira: E-mail: marcelos@seas.upenn.edu, Telephone: +1 215 898 8745
Longin Jan Latecki: E-mail: latecki@temple.edu, Telephone: +1 215 204 5781
Jean Gallier: E-mail: jean@cis.upenn.edu, Telephone: +1 215 898 4405

points in the resulting well-composed image. The resulting image has the same size as the input one. To our best knowledge, this is the first repairing algorithm for 3D binary digital images with this property. Rosenfeld, Kong and Nakamura[4] introduced an image operator, called *simple deformation*, that can be used to making 2D binary digital images well-composed. This operator can be extended to deal with 3D images in a straightforward manner,[11] but the resulting well-composed images are 9 and 27 times bigger than the input one in the 2D and 3D cases, respectively.

The remaining of this paper is organized in the following way: Section 2 introduces basic concepts of digital topology that are needed for the description of our algorithm; Section 3 formally defines 3D well-composed images and presents some of their most important properties; Section 4 describes our algorithm for making 3D binary digital images well-composed; Section 5 presents the computational complexity of our algorithm and provides a probabilistic argument for its effectiveness; Section 6 reports on the use of our algorithm on several medical digital images; and Section 7 summarizes our results and discusses future work.

2. PRELIMINARIES

This section introduces some basic concepts from "digital topology", which is the field that studies the topological properties of digital images. The material is this section can be found in a book by Herman[12] and in a paper by Latecki.[1]

Let X be any set of points in $\mathbb{R}^3$. Then, the *Voronoi Neighborhood* $\mathcal{V}_X(p)$ in $\mathbb{R}^3$ of any element p of X is defined as

$$\mathcal{V}_X(p) = \{x \in \mathbb{R}^3 \mid d(x,p) \leq d(x,r), \forall r \in X\},$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance function between two points in $\mathbb{R}^3$. In other words, the Voronoi neighborhood $\mathcal{V}_X(p)$ of p in X consists of all those points in $\mathbb{R}^3$ which are at least as close to p as to any other point r of X.

As usual in digital topology, we interpret $\mathbb{Z}^3$ as the set of points with integer coordinates in $\mathbb{R}^3$, and we identify each point p of $\mathbb{Z}^3$ with its Voronoi neighborhood $\mathcal{V}_{\mathbb{Z}^3}(p)$ in $\mathbb{Z}^3$. Note that $\mathcal{V}_{\mathbb{Z}^3}(p)$ is the unit upright cube in $\mathbb{R}^3$ centered at p and whose faces are parallel to the coordinate planes. Since this correspondence between points in $\mathbb{Z}^3$ and their Voronoi neighborhoods in $\mathbb{R}^3$ plays an important role in this work, we define it formally as a function $CA : \mathbb{Z}^3 \to \mathcal{P}(\mathbb{R}^3)$ such that, for every $p \in \mathbb{Z}^3$, $CA(p) = \mathcal{V}_{\mathbb{Z}^3}(p)$, where $\mathcal{P}(\mathbb{R}^3)$ denotes the power set of $\mathbb{R}^3$. We refer to $CA(p)$ as the *continuous analog* or *voxel* of p. We can extend the definition of continuous analog of a point to a set of points as follows: The *continuous analog* $CA(X)$ *of a set* $X \subset \mathbb{Z}^3$ is defined as $CA(X) = \bigcup\{CA(p) \mid p \in X\}$. In this case, CA can be viewed as a function $CA : \mathcal{P}(\mathbb{Z}^3) \to \mathcal{P}(\mathbb{R}^3)$, where $\mathcal{P}(\mathbb{Z}^3)$ is the power set of $\mathbb{Z}^3$. Figure 1 illustrates the continuous analog $CA(X)$ of a set $X = \{a,b,c,d\}$ consisting of four points in $\mathbb{Z}^3$.

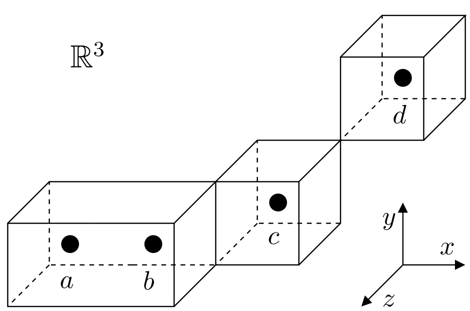

Figure 1. Continuous analog of the set $X = \{a,b,c,d\} \subset \mathbb{Z}^3$.

Let A be any set and ρ be a binary relation on A, i.e., ρ is a subset of $A^2 = A \times A$, the set of all ordered pairs of elements of A. If $(p,q) \in \rho$ then we say that p *is* ρ*-adjacent to* q. Furthermore, if ρ is a *symmetric* relation, i.e., if for any $p,q \in A$, we have that $(p,q) \in \rho$ if and only if $(q,p) \in \rho$, then we also say that p and q

are ρ-adjacent. For $A = \mathbb{Z}^3$, we define three symmetric binary relations on $\mathbb{Z}^3$, α_3, δ_3 and ω_3, as follows: For any two points $p = (p_1, p_2, p_3)$ and $q = (q_1, q_2, q_3)$ of $\mathbb{Z}^3$,

$$(p, q) \in \alpha_3 \text{ if and only if } p \neq q \text{ and, for } 1 \leq i \leq 3,\ |p_i - q_i| \leq 1,$$

$$(p, q) \in \delta_3 \text{ if and only if } (p, q) \in \alpha_3 \text{ and } \sum_{i=1}^{3} |p_i - q_i| \leq 2,$$

and

$$(p, q) \in \omega_3 \text{ if and only if } \sum_{i=1}^{3} |p_i - q_i| = 1.$$

Note that $\omega_3 \subseteq \delta_3 \subseteq \alpha_3$. For instance, points a and b of $\mathbb{Z}^3$ shown in Fig. 1 are ω_3-, δ_3-, and α_3-adjacent, points b and c are δ_3- and α_3-adjacent but not ω_3-adjacent, and points c and d are α_3-adjacent but not δ_3- nor ω_3-adjacent. The symmetric binary relations ω_3, δ_3 and α_3 are also denoted by the numbers 6, 18, and 26, respectively. For any two distinct points p and q in $\mathbb{Z}^3$, if $(p, q) \in \omega_3$ then we say that $CA(p)$ and $CA(q)$ are *face-adjacent*; if $(p, q) \notin \omega_3$ and $(p, q) \in \delta_3$ then we say that $CA(p)$ and $CA(q)$ are *edge-adjacent*; and if $(p, q) \notin \delta_3$ and $(p, q) \in \alpha_3$ then we say that $CA(p)$ and $CA(q)$ are *corner-adjacent*. So, for the points a, b, c, and d of $\mathbb{Z}^3$ in Fig. 1, we have that $CA(a)$ and $CA(b)$ are face-adjacent, $CA(b)$ and $CA(c)$ are edge-adjacent, and $CA(c)$ and $CA(d)$ are corner-adjacent.

Let A be any set. For any p and q in A, the sequence $\langle x^{(0)}, \ldots, x^{(k)} \rangle$ of elements of A is said to be a *ρ-path in A connecting p to q* if $x^{(0)} = p$, $x^{(k)} = q$ and, for $1 \leq i \leq k$, $x^{(i-1)}$ is ρ-adjacent to $x^{(i)}$. We call k the *length* of this path. In particular, there are ρ-paths of length zero, e.g., $\langle x \rangle$. We refer to them as *trivial paths*. If there is a ρ-path in A connecting p to q then we say that *p is ρ-connected in X to q*. For instance, if $X = \{a, b, c, d\} \subset \mathbb{Z}^3$ is the set of points in Fig. 1, then points a and b are ω_3-, δ_3- and α_3-connected in X, points a and c are δ_3- and α_3-connected in X, but not ω_3-connected, and points a and d are α_3-connected in X, but not ω_3- nor δ_3-connected in X. We say that a subset B of A is a *ρ-connected subset* if, for any p and q in B, we have that p is ρ-connected in B to q. It is easy to see that ρ-connectedness in A is a reflexive and transitive relation on A. If it is also symmetric, then it is an equivalence relation on A, which implies that it partitions A into *ρ-components*, which are nonempty ρ-connected subsets of A that are not proper subsets of any other ρ-connected subset of A. Note that if ρ is a symmetric relation then ρ-connectedness in A is also a symmetric relation on A, and therefore an equivalence relation on A.

A *digital space* is a pair (A, ρ), where A is an arbitrary nonempty set and ρ is a symmetric binary relation on A such that A is ρ-connected. For our purposes, the set A is always a subset of $\mathbb{Z}^3$ and the symmetric binary relation ρ is either ω_3, δ_3, or α_3. The elements of A are called *spels* (short for spatial elements) and elements of ρ are called *surfels* (short for surface elements). Any nonempty subset of ρ is called a *digital surface* in (A, ρ). Although ρ is a symmetric binary relation, the fact that a surfel (p, q) is in the digital surface does not imply that (q, p) must be. Actually, we will restrict our attention to *antisymmetric digital surfaces* S in (A, ρ), meaning that if $(p, q) \in S$ then $(q, p) \notin S$. Note that an antisymmetric digital surface is also an antireflexive relation: For every spel p, we have that $(p, p) \notin S$.

Let (A, ρ) be a digital space, and let M and N be subsets of A. Then, the *digital boundary in (A, ρ) between M and N* is defined as $bd_\rho(M, N) = \{(p, q) \in A \mid (p, q) \in \rho,\ p \in M, \text{ and } q \in N\}$. Note that, if $bd_\rho(M, N)$ is not empty, then $bd_\rho(M, N)$ is a digital surface in (A, ρ). Furthermore, if $bd_\rho(M, N)$ is nonempty and M and N are disjoint sets, then $bd_\rho(M, N)$ is clearly an antisymmetric surface. For instance, consider the digital space $(\mathbb{Z}^3, \alpha_3)$, and let X be the singleton set $\{a\} \subseteq \mathbb{Z}^3$ such that $a = (0, 0, 0)$. Then, the boundary $bd_{\omega_3}(X, X^c)$ in $(\mathbb{Z}^3, \alpha_3)$ between X and X^c, where $X^c = \mathbb{Z}^3 \setminus X$, is the antisymmetric digital surface consisting of the surfels (a, b), (a, c), (a, d), (a, e), (a, f), and (a, g), where b, c, d, e, f and g are points of $\mathbb{Z}^3$ with $b = (-1, 0, 0)$, $c = (1, 0, 0)$, $d = (0, -1, 0)$, $e = (0, 1, 0)$, $f = (0, 0, -1)$, and $g = (0, 0, 1)$.

We can also extend the definition of continuous analog to apply to sets of surfels in $(\mathbb{Z}^3, \rho)$ as follows: Let $CA : \mathbb{Z}^3 \times \mathbb{Z}^3 \to \mathcal{P}(\mathbb{R}^3)$ be a function such that $CA((p, q)) = CA(p) \cap CA(q)$, for $p, q \in \mathbb{Z}^3$, and let $CA : \mathcal{P}(\mathbb{Z}^3 \times \mathbb{Z}^3) \to \mathcal{P}(\mathbb{R}^3)$ be a function such that $CA(B) = \bigcup\{CA(b) \mid b \in B\}$, where $B \subseteq \mathbb{Z}^3 \times \mathbb{Z}^3$ and $\mathcal{P}(\mathbb{Z}^3 \times \mathbb{Z}^3)$ is the power set of $\mathbb{Z}^3 \times \mathbb{Z}^3$. For instance, if B is the digital boundary $bd_{\omega_3}(X, X^c)$ in the previous example, then $CA(B)$ is exactly the topological boundary of $CA(a)$ in $\mathbb{R}^3$, i.e., the set of points in the faces of the voxel $CA(a)$. Here, we are only interested in digital boundaries $bd_{\omega_3}(M,N)$ in $(\mathbb{Z}^3,\alpha_3)$ between a set M and its complement set $N = M^c$ when either M or M^c is a nonempty *digital set* of $\mathbb{Z}^3$, i.e., a nonempty and finite subset of $\mathbb{Z}^3$.

By definition, the continuous analog $CA(bd_{\omega_3}(M,M^c))$ of the digital boundary $bd_{\omega_3}(M,M^c)$ in $(\mathbb{Z}^3,\omega_3)$ between M and M^c is the set of points of the faces of the voxels in $CA(M)$ that are shared by a voxel in $CA(M)$ and a voxel in $CA(M^c)$, or equivalently, not in $CA(M)$. We denote $CA(bd_{\omega_3}(M,M^c))$ by $bdCA(M)$. Note that $bdCA(M)$ is just the topological boundary of $CA(M)$ in $\mathbb{R}^3$. Since, for any two points p and q of $\mathbb{Z}^3$, we have that $(p,q) \in bd_{\omega_3}(M,M^c)$ if and only if $(q,p) \in bd_{\omega_3}(M^c,M)$, it follows that $bd_{\omega_3}(M,M^c) \neq bd_{\omega_3}(M^c,M)$, but $bdCA(M) = bdCA(M^c)$.

A *3D (gray-scale) digital image* $\mathcal{I} : D \to V$ is a function from a subset D of points in $\mathbb{Z}^3$ to a subset V of $\mathbb{R}$. We refer to D as *grid* and to the elements in V as *colors*. A *3D binary digital image* is a 3D digital image $\mathcal{I} : D \to V$ in which the set of colors V is the binary set $\{0,1\}$. We shall denote a binary image $\mathcal{I}$ by the pair (D,X), where D is the grid of $\mathcal{I}$ and X is the set $\{p \in D \mid \mathcal{I}(p) = 1\}$. The sets X and X^c, where $X^c = D \setminus X = \{p \in D \mid \mathcal{I}(p) = 0\}$ is the complement set of X with respect to D, are commonly referred to as *foreground* and *background* points of (D,X), respectively. Since every background point is assigned the color 0 and every foreground point is assigned the color 1, we sometimes denote X by X_1 and X^c by X_0. A binary image (D,X) is typically obtained from a gray-scale digital image $\mathcal{I}' : D \to V'$ as follows: We choose a real number $\alpha \in \mathbb{R}$ and then, for every $p \in D$, we let $\mathcal{I}(p) = 0$ if $\mathcal{I}'(p) \leq \alpha$ and $\mathcal{I}(p) = 1$ otherwise; that is, we define $X = \{p \in D \mid \mathcal{I}'(p) > \alpha\}$ and $X^c = \{p \in D \mid \mathcal{I}'(p) \leq \alpha\}$. This method of generating (D,X) from $\mathcal{I}$ is commonly referred to as “thresholding”.

3. WELL-COMPOSEDNESS

Let $(\mathbb{Z}^3,X)$ be a 3D binary digital image, and assume that either X or its complement $X^c = \mathbb{Z}^3 \setminus X$ is a digital set. Recall that a subset S of $\mathbb{R}^3$ is a 2D manifold (surface) if each point in S has an open neighborhood homeomorphic to the unit open disk in $\mathbb{R}^2$.[13] Now, consider the following definition of a 3D well-composed, binary digital image in Ref. 1:

Definition 3.1. *A 3D binary digital image $(\mathbb{Z}^3,X)$ is* well-composed *if $bdCA(X)$ is a 2D manifold.*

From the above definition, it is not easy to think of any property that the points in X or X^c must satisfy in order to have that $(\mathbb{Z}^3,X)$ is well-composed. However, well-composedness is indeed equivalent to two simple and local conditions involving voxels of $CA(X)$ and $CA(X^c)$ that contain a face in $bdCA(X)$. Let Y be a any subset of four points of $\mathbb{Z}^3$. We say that Y is an instance of the *critical configuration (C1) in* $(\mathbb{Z}^3,X)$ if the voxels of the four points of Y share an edge, two of them are in $CA(X)$ and the other two are in $CA(X^c)$, and the two voxels in $CA(X)$ (resp. $CA(X^c)$) are edge-adjacent. Figure 2(a) illustrates the voxels of the points of Y when Y is an instance of the critical configuration (C1). Now, let Y be a any subset of eight points of $\mathbb{Z}^3$. We say that Y is an instance of the *critical configuration (C2) in* $(\mathbb{Z}^3,X)$ if the voxels of the eight points of Y share a vertex, two of them are in $CA(X)$ (resp. $CA(X^c)$) and the other six are in $CA(X^c)$ (resp. $CA(X)$), and the two voxels in $CA(X)$ (resp. $CA(X^c)$) are corner-adjacent. Figure 2(b) illustrates the voxels of the points of Y when Y is an instance of the critical configuration (C2).

The following theorem proved in Ref. 1 establishes an important equivalence between the well-composedness property of a 3D binary digital image and the nonexistence of instances of critical configurations (C1) and (C2) in the image:

Theorem 3.1. *A 3D binary digital image $(\mathbb{Z}^3,X)$ is well-composed if and only if the critical configurations (C1) and (C2) with respect to X do not occur in $\mathbb{Z}^3$.*

Note that it is a straightforward task to determine if a binary image is well-composed based on the results from Theorem 3.1, as we just have to verify if there is any instance of (C1) and (C2) in $(\mathbb{Z}^3,X)$. Note also that $(\mathbb{Z}^3,X)$ is well-composed if and only if $(\mathbb{Z}^3,X^c)$ is well-composed, as $bdCA(X) = bdCA(X^c)$. Theorem 3.1 also implies that there is only one kind of connectedness in well-composed images, i.e., if $(\mathbb{Z}^3,X)$ is well-composed then any α_3-connected component of X or X^c is also a δ_3-connected component, which in turn is also a ω_3-connected component. This is due to the fact that, if a δ_3-connected component of X is not ω_3-connected, then there must exist an instance of (C1) in $(\mathbb{Z}^3, X)$. Similarly, if a α_3-connected component of X is not ω_3-connected, then there must exist an instance of (C2) in $(\mathbb{Z}^3, X)$. Finally, it has also been shown in Ref. 1 that, if $(\mathbb{Z}^3, X)$ is well-composed, then every connected component of $bdCA(X)$ is a *simple closed surface* or *Jordan surface*, i.e., a connected 2D manifold in $\mathbb{R}^3$. Furthermore, each connected component of $bdCA(X)$ separates the points of a ω_3-component Y of $\mathbb{Z}^3$, such that $X \subseteq Y$, from the points of Y^c, which is also a ω_3-component. The digital boundary $bd_{\omega_3}(Y, Y^c)$ in $(\mathbb{Z}^3, \alpha_3)$ is called a Jordan (digital) surface by Herman,[12] as it captures the notion of a Jordan surface in $\mathbb{R}^3$.

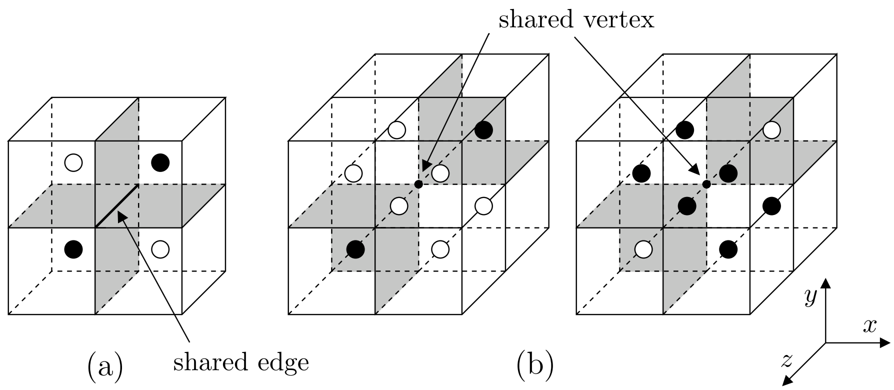

Figure 2. (a) Example of critical configuration (C1) in $(\mathbb{Z}^3, X)$. (b) Example of critical configuration (C2) in $(\mathbb{Z}^3, X)$. Points in X are represented by black circles, while points in X^c are represented by white circles.

4. THE ALGORITHM

We now describe our new algorithm for repairing 3D binary digital images. Our algorithm takes as input a 3D binary digital image (D, X) such that D is assumed to be a finite rectilinear grid of points of $\mathbb{Z}^3$, $D = [d_{11}, d_{12}] \times [d_{21}, d_{22}] \times [d_{31}, d_{32}]$, where $d_{i1}, d_{i2} \in \mathbb{Z}$ and $d_{i1} < d_{i2}$, for $1 \leq i \leq 3$. The output is a 3D well-composed, binary digital image (D, X') such that $X = X'$ if (D, X) is already well-composed, and $X \subset X'$ otherwise. In other words, if (D, X) is not well-composed, then the foreground X'_1 of (D, X') is a proper superset of the foreground X_1 of (D, X), or equivalently, the background X'_0 of (D, X') is a proper subset of the background X_0 of (D, X). The set X' is computed by finding a subset P of X_0 such that the image (D, X'), with $X' = X \cup P$, is well-composed. So, the set P can be viewed as the subset of the set X_0 of background points of (D, X) whose assigned colors are changed from 0 to 1 to produce (D, X'). Observe that such a set P always exists, as (D, X') is well-composed if we let $P = X_0$. However, we ideally would like to find a smallest $P \subseteq X_0$ such that (D, X') is well-composed.

Since D is a finite set, the background X_0 of (D, X) is also finite. Hence, there is a very simple algorithm for finding a smallest P: Enumerate and test all subsets P of X_0 in increasing order of cardinality until a well-composed image $(D, X' = X \cup P)$ is found. Recall that we can determine if $(D, X' = X \cup P)$ is well-composed by checking the existence of an instance of (C1) or (C2) in $(\mathbb{Z}^3, X')$. But, because X_0 has $2^{|X_0|}$ subsets, where X_0 is the cardinality of X_0, and X_0 has typically one or two million points in practical applications, this algorithm is impractical. By realizing that critical configurations are *local* configurations of points of D, we devised a better alternative to this naïve solution. Our algorithm is not guaranteed to find a smallest P such that $(D, X' = X \cup P)$ is well-composed, but our experiments in Section 6 show evidences that it is very effective on practical data, and Section 5 provides a formal proof for its effectiveness.

Our algorithm starts by letting $P = \emptyset$, and then it iteratively inserts points from $X_0 \setminus P$ into P one at a time. Every point inserted into P eliminates at least one instance of a critical configuration in $(D, X \cup P)$. However, the insertion of a point may also give rise to *new* critical configurations in D, which will trigger the insertion of at least one more point from $X_0 \setminus P$ into P. Before we give details of our repairing algorithm, we describe which points from $X_0 \setminus P$ the algorithm chooses to insert into P in order to eliminate *one* instance of (C1) or (C2) from $(D, X \cup P)$.

First, consider an instance Y of (C1) in $(D, X \cup P)$, and refer to Fig. 2(a). The set Y has four points, two of them, say a and b, are in $X_0 \setminus P$ and two of them are in $X \cup P$. We notice that the insertion of either a or b into P eliminates Y from $(D, X \cup P)$. On the other hand, points a and b are the only ones from $X_0 \setminus P$ that can be inserted into P in order to eliminate Y from $(D, X \cup P)$. Now, consider an instance Y of (C2) in $(D, X \cup P)$, and refer to Fig. 2(b). The set Y has eight points, two of them are in $X_0 \setminus P$ (resp. $X \cup P$) and the other six are in $X \cup P$ (resp. $X_0 \setminus P$). Consider the case in which Y has exactly two points in $X_0 \setminus P$, say a and b. We notice that the insertion of either a or b into P eliminates Y from $(D, X \cup P)$. Furthermore, points a and b are the only ones from $X_0 \setminus P$ that can be inserted into P in order to eliminate Y from $(D, X \cup P)$. Finally, consider the case in which the critical configuration Y has six points in $X_0 \setminus P$. We first notice that the insertion of any of these points into P eliminates Y from $(D, X \cup P)$. However, this insertion always gives rise to an instance of (C1) in $(D, X \cup P)$. This situation is illustrated in Fig. 3. On the other hand, only the insertion of one of the six points in $Y \cap (X_0 \setminus P)$ can eliminate Y from $(D, X \cup P)$.

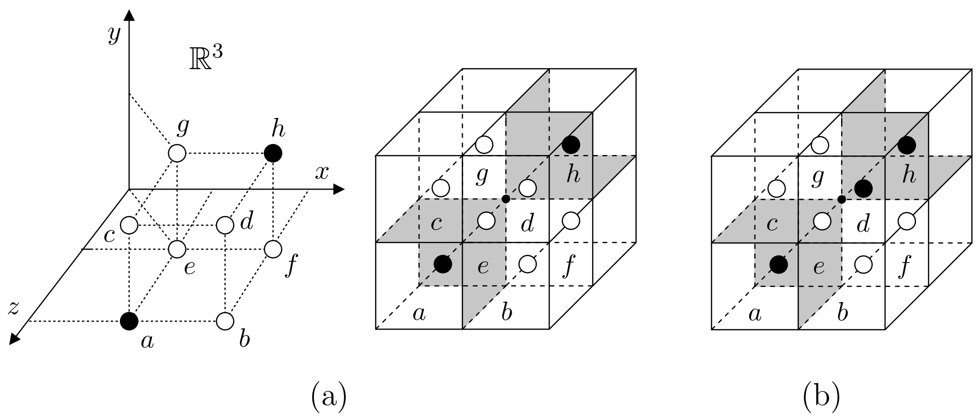

Figure 3. (a) $Y = \{a, b, c, d, e, f, g, h\}$ is an instance of (C2) in $(D, X \cup P)$. The six points in $Y \cap (X_0 \setminus P)$ are $\{b, c, d, e, f, g\}$. (b) We can eliminate Y from $(D, X \cup P)$ by inserting d into P. However, this elimination gives rise to an instance of (C1) in $(D, X \cup P)$, where P is now a set containing the point d.

Our algorithm computes the set $P \subseteq X_0$, such that $(D, X' = X \cup P)$ is well-composed, in two steps. First, the algorithm loops over all points in D to find all instances of (C1) and (C2) in (D, X). Next, the algorithm eliminates all critical configurations found in the first step and the ones that may eventually arise during this elimination process. We now describe the first step. Assume that P is initially empty and recall that $D = [d_{11}, d_{12}] \times [d_{21}, d_{22}] \times [d_{31}, d_{32}]$, where $d_{i1}, d_{i2} \in \mathbb{Z}$ and $d_{i1} < d_{i2}$, for $1 \leq i \leq 3$. So, for each $j \in \{d_{11}, \ldots, d_{12}\}$, $k \in \{d_{21}, \ldots, d_{22}\}$, and $l \in \{d_{31}, \ldots, d_{32}\}$, we consider the point $r \in D$ with integer coordinates $r = (j, k, l)$, and we define the sets $\mathcal{N}_x(r) = \{(j, y, z) \mid y \in \{k, k+1\} \text{ and } z \in \{l, l+1\}\}$, $\mathcal{N}_y(r) = \{(x, k, z) \mid x = \{j, j+1\} \text{ and } z \in \{l, l+1\}\}$, $\mathcal{N}_z(r) = \{(x, y, l) \mid x = \{j, j+1\} \text{ and } y \in \{k, k+1\}\}$, and $\mathcal{N}(r) = [j, j+1] \times [k, k+1] \times [l, l+1]$ of points of $\mathbb{Z}^3$. Note that each instance of (C1) in $(D, X \cup P)$ is one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$, for some $r \in D$. Likewise, each instance of (C2) in $(D, X \cup P)$ is the set $\mathcal{N}(r)$, for some $r \in D$. So, in order to find all instances of (C1) and (C2) in (D, X), the algorithm considers the sets $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$, for every $r \in D$. If any of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration, the algorithm inserts r into a list Q, which is empty in the beginning of the first step.

If Q is empty in the end of the first step then the input image (D, X) is already well-composed, and the algorithm outputs (D, X) and terminates. Otherwise, the second step takes place. The algorithm removes one point r from Q at a time, and then eliminates *all* instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$, and the instance of (C2) in $\mathcal{N}(r)$, if any. By examining Fig. 2, we can see that if $\mathcal{N}(r)$ is an instance of (C2) in $(D, X \cup P)$, then none of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$ and $\mathcal{N}_z(r)$ is an instance of (C1) in $(D, X \cup P)$. Conversely, if at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$ and $\mathcal{N}_z(r)$ is an instance of (C1) in $(D, X \cup P)$, then $\mathcal{N}(r)$ is not an instance of (C2) in $(D, X \cup P)$. So, for each

$r \in Q$, there are three mutually exclusive cases: None of the sets $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration in $(D, X \cup P)$, or at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) in $(D, X \cup P)$, or $\mathcal{N}(r)$ is an instance of (C2) in $(D, X \cup P)$.

Consider the case in which at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) in $(D, X \cup P)$, i.e., assume that the set $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains one, two or three instances of (C1) in $(D, X \cup P)$. We saw before that an instance Y of (C1) in $(D, X \cup P)$ can be eliminated from $(D, X \cup P)$ if and only if one of the two points of $Y \cap (X_0 \setminus P)$ is inserted into P. So, if $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly one instance Y of (C1) in $(D, X \cup P)$, then our algorithm chooses one out of two points to insert into P in order to eliminate Y from $(D, X \cup P)$. If $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly two instances of (C1) in $(D, X \cup P)$, say Y_1 and Y_2, then note that the set $Y_1 \cup Y_2$ has exactly three points of $X_0 \setminus P$, not four, as any two instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ share two points, one of which is a point of $X_0 \setminus P$ and the other is a point of $X \cup P$. Figure 4 illustrates all possible cases in which the set $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ has exactly two instances of (C1) in $(D, X \cup P)$. Note that, for each of the six cases in Fig. 4, our repairing algorithm eliminates both Y_1 and Y_2 from $(D, X \cup P)$ by inserting either the common point of Y_1 and Y_2 in $X_0 \setminus P$ or the other two points of $(Y_1 \cup Y_2) \cap (X_0 \setminus P)$ into P. Finally, if $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains three instances of (C1) in $(D, X \cup P)$, then each of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) and $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ has either three or four points of $X_0 \setminus P$, as illustrated by Fig. 5. If $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly three points of $X_0 \setminus P$, then our algorithm inserts two of these three points into P to eliminate all instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$. Otherwise, our algorithm inserts either the common point of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ in $X_0 \setminus P$ or the other three points of $(\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)) \cap (X_0 \setminus P)$ into P. In either case, all instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ are eliminated from $(D, X \cup P)$.

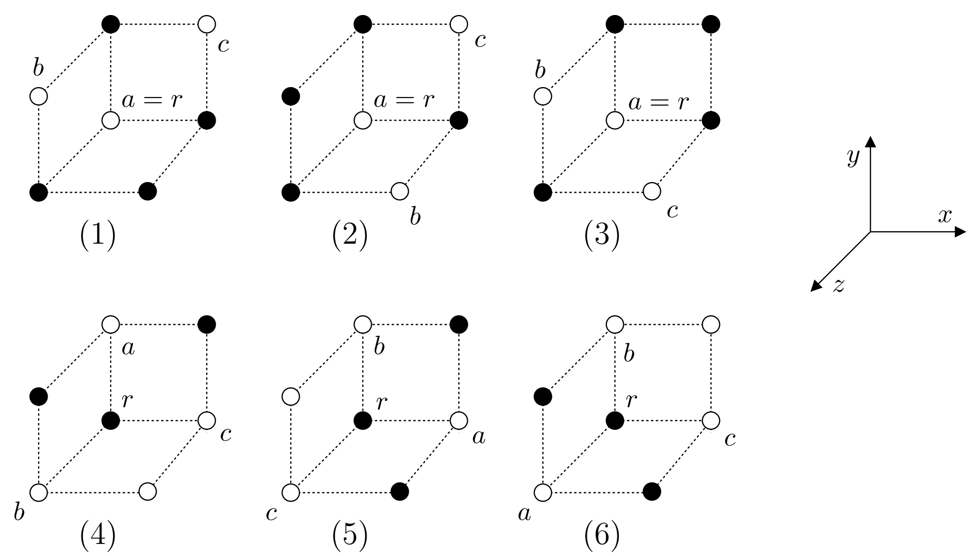

Figure 4. All possible cases in which the set $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly two instances of (C1) in $(D, X \cup P)$. Points in $X \cup P$ are represented by black circles, while points in $X_0 \setminus P$ are represented by white circles. Our repairing algorithm inserts either point a or points b and c into P to eliminate the two instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ from $(D, X \cup P)$.

Now, consider the case in which $\mathcal{N}(r)$ is an instance of (C2) in $(D, X \cup P)$. As shown in Fig. 2(b), the set $\mathcal{N}(r)$ contains either two or six points of $X_0 \setminus P$. In either case, our repairing algorithm inserts only one of these points into P to eliminate the instance of (C2) in $\mathcal{N}(r)$ from $(D, X \cup P)$. Recall that, if the set $\mathcal{N}(r)$ contains six points of $X_0 \setminus P$, then the insertion of any of these points into P always gives rise to a new critical configuration in $(D, X \cup P)$.

Our repairing algorithm chooses one or more points to insert into P to eliminate all instances of (C1) in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ or the instance of (C2) in $\mathcal{N}(r)$ according to three rules. These rules are mutually exclusive. Before we describe the rules, we define three sets of subsets of points of $X_0 \setminus$ used in our description of the rules: $B(r)$, $B_1(r)$, and $B_2(r)$. We define $B(r)$ as the set of all subset of points of $X_0 \setminus P$ that our algorithm can insert into P in order to eliminate all instances of (C1) or (C2) in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$. Then, if $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly one instance Y of (C1) in $(D, X \cup P)$, then $B(r) = \{\{a\}, \{b\}\}$, where a and b are the two points of $Y \cap (X_0 \setminus P)$. If $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains exactly two instances of (C1) in $(D, X \cup P)$, say Y_1 and Y_2, then $B(r) = \{\{a\}, \{b, c\}\}$, where a is the common point of Y_1 and Y_2 and $X_0 \setminus P$, and b and c are the two other points of $X_0 \setminus P$ in $Y_1 \cup Y_2$. If $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains three instances of (C1) in $(D, X \cup P)$, then we have two situations: If the common point a of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is in $X_0 \setminus P$, then $B(r) = \{\{a\}, \{b, c, d\}\}$, where b, c, and d are the other points of $(\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)) \cap (X_0 \setminus P)$. Otherwise, $B(r) = \{\{b, c\}, \{b, d\}, \{c, d\}\}$, where b, c, and d are the points of the set $(\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)) \cap (X_0 \setminus P)$. Finally, if $\mathcal{N}(r)$ contains an instance of (C2) in $(D, X \cup P)$, then we also have two situations: If $\mathcal{N}(r)$ has exactly two points a and b of $X_0 \setminus P$, then $B(r) = \{\{a\}, \{b\}\}$. Otherwise, $B(r) = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}\}$, where a, b, c, d, e, and f are the six points of $\mathcal{N}(r) \cap (X_0 \setminus P)$. We also define the partition $B_1(r)$ and $B_2(r)$ of $B(r)$ such that $S \in B_1(r)$ if and only if $S \in B(r)$ and the insertion of all elements of S into P does not give rise to any critical configuration in $(D, X \cup P)$, and $S \in B_2(r)$ if and only if $S \in B(r)$ and the insertion of all elements of S into P gives rise to at least one critical configuration in $(D, X \cup P)$. Now, for each $r \in D$, if one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration, our repairing algorithm chooses a set S from $B(r)$ according to the following three rules:

(1) If $B_1(r)$ is a unit set, then let S be $B_1(r)$ and insert the elements of S into P.

(2) If $B_1(r)$ has two or more elements, then select a set S from $B_1(r)$ at random and insert the elements of S into P.

(3) If $B_1(r)$ is the empty set, or equivalently, if $B_2(r) = B(r)$, then select a set S from $B_2(r)$ at random and insert the elements of S into P. Furthermore, insert into list Q all points $p \in D$ such that at least one of the sets $\mathcal{N}_x(p)$, $\mathcal{N}_y(p)$, $\mathcal{N}_z(p)$, and $\mathcal{N}(p)$ is a critical configuration created by inserting the points of S into P.

Note that any point $p \in D$ inserted into Q by rule (3) must be contained in $\mathcal{N}(r)$, as $S \subset \mathcal{N}(r)$. So, the sets $\mathcal{N}_x(p)$, $\mathcal{N}_y(p)$, $\mathcal{N}_z(p)$, and $\mathcal{N}(p)$ are all contained in the grid $[k-1, k+2] \times [j-1, j+2] \times [l-1, l+2] \subset \mathbb{Z}^3$, where (j, k, l) are the integer coordinates of point r. So, our algorithm can find all points $p \in D$ to insert into Q by looking for new critical configurations created by the insertion of the elements of S into P inside the grid $[k-1, k+2] \times [j-1, j+2] \times [l-1, l+2]$.

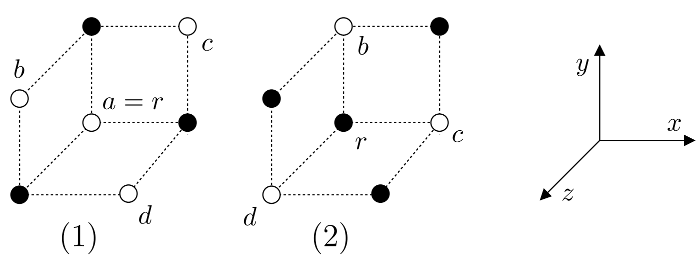

Figure 5. The two cases in which the set $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains three instances of (C1) in $(D, X \cup P)$. Points in $X \cup P$ are represented by black circles, while points in $X_0 \setminus P$ are represented by white circles. In case (1), our algorithm inserts either a or b and c into P to eliminate the three instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ from $(D, X \cup P)$. In case (2), either b and c, or b and d, or c and d are inserted into P.

Since critical configurations may share points in $X_0 \setminus P$, the insertion of the points of S into P may actually eliminate more than the critical configurations found in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$. So, by the time the algorithm picks the next point r from Q, it is possible that none of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration, even though at least one of them was. Otherwise, point r would not have been inserted into Q. In our experiments with medical digital images (see Section 6), we noticed that this situation occurred very often.

Recall that, in the beginning of the second step, the list Q contains the point $r \in D$ if and only if at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) or $\mathcal{N}(r)$ is an instance of (C2) in the input image (D, X). Each iteration of the second step removes one point r from Q, and eliminates all instances of (C1) in $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ and the instance of (C2) in $\mathcal{N}(r)$, if any. This elimination process may give rise to new critical configurations in $(D, X \cup P)$. Each instance of a new critical configuration is a set $\mathcal{N}_x(p)$, $\mathcal{N}_y(p)$, $\mathcal{N}_z(p)$, or $\mathcal{N}(p)$, for some $p \in D$. Rule (3) inserts p into Q whenever one of $\mathcal{N}_x(p)$, $\mathcal{N}_y(p)$ and $\mathcal{N}_z(p)$ is an instance of (C1) or $\mathcal{N}(p)$ is an instance of (C2) in $(D, X \cup P)$ after the insertion of the elements of S into P. So, at any given time, the set $\bigcup_{r \in Q}\{\mathcal{N}_x(r), \mathcal{N}_y(r), \mathcal{N}_z(r), \mathcal{N}(r)\}$ is a superset of the set of critical configurations in $(D, X \cup P)$. Hence, if Q is empty, the binary image $(D, X' = X \cup P)$ must be well-composed. Our algorithm ends the second step when Q becomes empty.

We now must show that Q eventually becomes empty. Note that this termination condition implies the correctness of our algorithm, as $(D, X' = X \cup P)$ is well-composed if Q is empty. In fact, the list Q eventually becomes empty, and this fact follows from two simple observations. First, the set D is a finite set, which implies that the set X_0 of background points of (D, X) is also finite. Second, the algorithm inserts at least one point from $X_0 \setminus P$ into P in each iteration of the second step, and it never removes points from P. Since there are finitely many points in X_0, our algorithm cannot insert points into P indefinitely. So, after a finite number of iterations, no more points are inserted into Q. Since the algorithm removes one point from Q in each iteration, the list Q will eventually become empty, and therefore our algorithm always terminates and produces a well-composed image. In principle, the set P obtained by our algorithm may be the entire set X_0 of background points. However, as we shall see in the next section, it is very unlikely that our algorithm will obtain a set P such that $P = X_0$.

5. COMPLEXITY, EFFECTIVENESS, AND OPTIMALITY

The first step of our repairing algorithm enumerates the sets $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$, for each $r \in D$, and determines if they are instances of a critical configuration in D with respect to X. So, the time complexity of the first step of our algorithm is $\mathcal{O}(|D|)$, where $|D|$ is the cardinality of D. The second step is also a loop whose number of iterations is the number n of points inserted into Q during the entire execution of the algorithm. Since each point r inserted into Q corresponds to at least one instance of a critical configuration in $(D, X \cup P)$, the number n is bounded by the number of critical configurations in $(D, X \cup P)$ during the entire execution of our algorithm, which in turn is no larger than $|D|$. So, the time complexity of the second step is also $\mathcal{O}(|D|)$, which implies that our algorithm is linear in the number of points of D.

Our algorithm is not guaranteed to obtain a smallest set P such that $(D, X' \cup P)$ is well-composed. Besides, the set P computed by our algorithm can in principle be much larger than the smallest possible size. Since P is a subset of X_0 and any instance of a critical configuration must contain at least two points of $X_0 \setminus P$, the size of the set P computed by *any* repairing algorithm that only adds points from $X_0 \setminus P$ to P is at most $|X_0| - 1$, where $|X_0|$ is the cardinality of X_0. Since the points inserted into P by our algorithm in each iteration of the second step eliminates at least one critical configuration of $(D, X \cup P)$, the size of the set P computed by our algorithm is also bounded above by $m + t$, where m is the number of critical configurations in (D, X) and t is the number of new critical configurations generated by our algorithm in order to produce the resulting well-composed image $(D, X' = X \cup P)$. So, we can have an idea of how effective our repairing algorithm is by providing an upper bound for the size of P in terms of m and t. On deriving this upper bound, we view m and t as an intrinsic feature of the input image and an intrinsic feature of our algorithm, respectively, and therefore we express the number t of new critical configurations in terms of m. Since our algorithm is randomized, we actually derive an expression for the *expected value* of t in terms of m.

From the description of our algorithm in Section 4, a new critical configuration is created if and only if rule (3) is applied to eliminate an existent critical configuration. We also know that any critical configuration in (D, X) or any new critical configuration created by our repairing algorithm is one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$, for some $r \in D$. Since $\mathcal{N}_x(r_1) \neq \mathcal{N}_x(r_2)$, $\mathcal{N}_y(r_1) \neq \mathcal{N}_y(r_2)$, $\mathcal{N}_z(r_1) \neq \mathcal{N}_z(r_2)$, and $\mathcal{N}(r_1) \neq \mathcal{N}(r_2)$, for any two distinct points r_1 and r_2, we can express t as $\sum_{r \in D} C(r)$, where $C(r)$ is the number of new critical configurations resulting from the elimination of *all* instances of (C1) in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, or the instance of (C2) in $\mathcal{N}(r)$, if any. If we think of $C(r)$ as a random variable, then the expected value $E[t]$ of t is $E[t] = E[\sum_{r \in D} C(r)] = \sum_{r \in D} E[C(r)]$, where $E[C(r)]$ is the expected value of $C(r)$. By definition, the expected value $E[C(r)]$ of $C(r)$ is given by $E[C(r)] = \sum_{i=1}^{K} i \cdot P[C(r) = i]$, where $P[C(r) = i]$ is the probability that $C(r)$ will have value i, and K is the largest value of $C(r)$. So, if we can find $P[C(r) = i]$, for $1 \leq i \leq K$, we can derive an expression for $E[C(r)]$ and $E[t]$.

Suppose that rule (3) is used to eliminate an instance of (C1) in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, or the instance of (C2) in $\mathcal{N}(r)$, for some $r \in D$. Let S be the set of points in $B(r)$ selected by rule (3) and whose elements are inserted into P. We know that every point $q \in S$ is a point of $\mathcal{N}(r)$, and that every new critical configuration created by the insertion of q into P is inside the set $G = [j-1, j+2] \times [k-1, k+2] \times [l-1, l+2]$ of points of $\mathbb{Z}^3$, where (j, k, l) are the integer coordinates of point r. These observations imply that we can compute $P[C(r) = i]$ by restricting our attention to the set G, which contains exactly 64 points. Let A be the set of all distinct binary images (G, X). Since each point in G is assigned a color from the binary set $\{0, 1\}$, the cardinality $|A|$ is 2^{64}. We are interested in the subset A' of A such that $(G, X) \in A'$ if and only if $(G, X) \in A$ and at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) in (G, X) or $\mathcal{N}(r)$ is an instance of (C2) in (G, X). There are exactly 84 ways of assigning a color from the set $\{0, 1\}$ to the points in $[j, j+1] \times [k, k+1] \times [l, l+1]$ such that at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, and $\mathcal{N}_z(r)$ is an instance of (C1) in (G, X) or $\mathcal{N}(r)$ is an instance of (C2) in (G, X). This implies that the cardinality $|A'|$ of the set A' is $|A'| = 2^{56} \cdot 84$. If we assume that each of the $2^{56} \cdot 84$ binary images (G, X) of A' are equally likely, then we can define the probability $P(X)$ of (G, X) as $P(X) = 1/|A| = 1/(2^{56} \cdot 84)$.

Let (G, X) be any binary image of A'. Since $(G, X) \in A'$, either the set $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ contains one, two or three instances of (C1) or the set $\mathcal{N}(r)$ is an instance of (C2). Recall that our repairing algorithm eliminates *all* critical configurations of (G, X) in either $\mathcal{N}_x(r) \cup \mathcal{N}_y(r) \cup \mathcal{N}_z(r)$ or $\mathcal{N}(r)$ by selecting a set S of $B(r)$ and inserting all elements of S into P. Recall also that rule (3) is used to select S if and only if, for any $S \in B(r)$, the insertion of the elements of S into P gives rise to at least one critical configuration in $(D, X \cup P)$. So, if we assume that each $S \in B(r)$ is equally likely to be selected by rule (3), then we can define the probability $P(S)$ that S will be chosen from B by rule (3) as $P(S) = 0$ if $B_2(r) \neq B(r)$ and $P(S) = 1/|B_2(r)|$ otherwise, where $|B_2(r)|$ is the cardinality of the set $B_2(r)$. Since the set $B(r)$ is not empty whenever one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration in $(D, X \cup P)$, the probability $P(S)$ is well-defined. Now, we can define the probability that the insertion of the elements of S into P will give rise to exactly i critical configurations by $\sum_{S \in B(r)} P(S) \cdot N(S, X, i)]$, where $N(S, X, i)$ is 1 if the insertion of the elements of S into P gives rise to exactly i critical configurations, and 0 otherwise.

Since any set $S \in B(r)$ contains at most 3 elements, and instances of (C1) and (C2) are mutually exclusive in the same set $\mathcal{N}(p)$, for any $p \in D$, and each point q in S can give rise to at most 12 instances of (C1) and 8 instances of (C2), we have that the largest value K of $C(r)$ is no larger than 36. Furthermore, since the set G has only 64 points, we can easily build a computer program to obtain $N(S, X, i)$ for each binary image (G, X) in A', for each set S in $B(r)$, and for each i, with $1 \leq i \leq K = 36$. Now, note that the probability $P[C(r) = i]$ that, for a given $r \in D$, the random variable $C(r)$ will have value i is precisely the sum of the probabilities $P(X)[\sum_{S \in B(X)} P(S) \cdot N(s, X, i)]$ that the elimination of all instances of (C1) in $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, or the instance of (C2) in $\mathcal{N}(r)$, if any, will give rise to exactly i new critical configurations given that at least one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration. Thus, we can express the probability $P[C(r) = i]$ that $C(r)$ will have value i as

$$P[C(r) = i] = \sum_{(G,X) \in A} P(X)[\sum_{S \in B(r)} P(S) \cdot N(s, X, i)].$$

We built a computer program to compute $N(S, X, i)^*$, and then we computed $P[C(r) = i]$, for each i with $1 \leq i \leq 36$. Using these values, we have that $E[C(r)]$ of $C(r)$ is 0.28529. The following theorem uses the fact that $E[C(r)] = \sum_{i=1}^{36} i \cdot P[C(r) = i] = 0.28529$ to derive an upper bound for the expected value $E[t] = \sum_{r \in D} E[C(r)]$ of t in terms of m:

*http://www.seas.upenn.edu/~marcelos/probab.cpp

THEOREM 5.1. *Let (D, X) be a 3D binary digital image. If there are m instances of critical configurations in (D, X), then the expected number of new critical configurations created by the repairing algorithm in Section 4 on input (D, X) is bounded above by $m/2$.*

Proof. Let R be the set of points $r \in D$ such that one of $\mathcal{N}_x(r)$, $\mathcal{N}_y(r)$, $\mathcal{N}_z(r)$, and $\mathcal{N}(r)$ is an instance of a critical configuration in (D, X). Clearly, the number of points in R is less than or equal to m. For any $r \in D$, we have that $E[C(r)] = 0.28529$. So, since $C(r) = 0$ for $r \in D \backslash R$, if there are m critical configurations in (D, X), then the expected number of critical configurations resulting from the elimination of m existent critical configurations in (D, X) is $E[\sum_{r \in D} C(r)] = E[\sum_{r \in R} C(r) + \sum_{r \in D \backslash R} C(r)] = E[\sum_{r \in R} C(r)] = \sum_{r \in R} E[C(r)] = 0.28529 \cdot |R| \leq 0.28529 \cdot m$, where $|R|$ is the cardinality of R. Since the algorithm iterates until all critical configurations are eliminated from the binary image $(D, X \cup P)$, where P is the set of background points converted into foreground points at any given time, we have that the expected number $E[t] = \sum_{r \in D} E[C(r)]$ of new critical configurations created during the entire execution of the algorithm is bound above by $\sum_{k=1}^{\infty} (0.28529)^k \cdot m < \sum_{k=1}^{\infty} (1/3)^k \cdot m = m/2$. $\square$

The result from Theorem 5.1 provides a theoretical basis for explaining the good performance of our repairing algorithm in the experiment described in Section 6. Our repairing algorithm is also relatively simple, as it is the case with most randomized algorithms.[14] However, in principle, the size of the set P obtained by our algorithm can still be much larger than the smallest possible size. We are currently investigating the existence of a deterministic, polynomial-time repairing algorithm that finds a smallest set P such that $(D, X \cup P)$ is well-composed. We have not yet ruled out the possibility that the *decision* problem equivalent to our optimization problem of finding such a smallest P is **NP**-complete. By the equivalent decision problem, we mean the following problem: Given any 3D binary digital image (D, X) and a positive integer k, can we decide, in polynomial time in $|D|$, whether or not there is a subset P of X_0 such that $(D, X \cup P)$ is well-composed and $|P| \leq k$? Note that we are considering the case in which points may be inserted into P but never removed from P, as it is the case with our repairing algorithm.

6. EXPERIMENTAL RESULTS

As we mentioned in the introduction, if the digitization process that gives rise to a given 3D binary digital image is topology-preserving then the image must be well-composed. So, if an image is not well-composed then the digitization process that produced the image is not topology-preserving. The purpose of any repairing algorithm is to "restore" the well-composedness property of the input image (D, X) whenever (D, X) is not well-composed. However, without additional information about the topology that (D, X) would have if it had been generated by a topology-preserving process, the repairing algorithm generates a well-composed image that is at best a good guess for the "correct" well-composed image. From the point of view of image-based applications, a good guess is more likely to be a well-composed image that does not differ too much from (D, X). So, in this context, if (D, X') is the well-composed image, then the size of the set $(X - (X \cap X')) \cup (X' - (X \cap X'))$ is the main indication of the usefulness and effectiveness of a repairing algorithm. Note that the size of $(X - (X \cap X')) \cup (X' - (X \cap X'))$ is precisely the number of identical points in both images that are not assigned the same color. Since $X \subseteq X'$ in the image generated by our repairing algorithm, the set $(X - (X \cap X')) \cup (X' - (X \cap X'))$ is exactly the set P such that $X' = X \cup P$.

We have shown an upper bound for the expected value of the number t of new critical configurations generated by our repairing algorithm on an image with m critical configurations (see Section 5). Since the size of the set P generated by our repairing algorithm is bounded above by $m + t$, this bound provides in theory an idea of the effectiveness of our algorithm. Here, we report on an experiment that applies our algorithm to several medical digital images encountered in practice. Our goal is to also show empirical evidences of the effectiveness of our repairing algorithm. We used six magnetic resonance (MR) images in our experiment. All of them are binary images with $129^3 = 2146689$ points obtained from resampling and thresholding segmented grey-scale images. Three of them were obtained from the segmentation of a normal brain image into white matter, grey matter, and cerebrospinal fluid (CSF). The normal brain image was produced by an on-line 3D MR image simulator,[15] which can be found in `www.bic.mni.mcgill.ca/brainweb`. Two other images are real lung images corresponding to the inspiration and expiration stages of lung motion [†]. The last image is a male thorax image from the dataset of

[†]We are very grateful to Tessa Sundaram for providing us with the segmented lung images used in this experiment.

the Visible Human Project in `www.nlm.nih.gov/research/visible`. This image was segmented using a fuzzy segmentation method.[16]

We ran our repairing algorithm on each image 10 times. In each execution of the algorithm, we collected the size of the set P obtained by the algorithm and the number of critical configurations created by our algorithm in order to generate $(D, X' = X \cup P)$. Finally, we computed the average size of P and the average number of new critical configurations for each image. Table 1 shows the results of our experiment. Note that, for each image, the average size of P is at most $0,32\%$ of the size of the image. Furthermore, in all cases, the average size of P is smaller than the total number of critical configurations, which implies that the optimal size of P is also smaller than the number of critical configurations, and therefore the average size of P and its optimal size are close. Note also that the average number of critical configurations is no larger than $0.2 \cdot m$, where m is the number of critical configurations in the input image. The results in Table 1 shows that our algorithm generated well-composed images that are very similar to the input ones. Hence, this algorithm can be very useful as a preprocessing step of algorithms that benefit from dealing with well-composed images, such as isosurface extraction and thinning algorithms.

Table 1. Results from the application of the repairing algorithm in Section 4 to six distinct MR images with 129^3 points each. The first column identifies the images; the second and third columns contain the number of instances of the critical configuration (C1) and critical configuration (C2) in the input image, respectively; the fourth column shows the average size of the set P; and the fifth column shows the average number of new critical configurations generated by our algorithm in order to compute P. The average size of P and the average number of critical configurations were computed by executing our repairing algorithm on each image 10 times.

Image	**# C. Conf. 1)**	**# C. Conf. 2)**	**Avg. |P|**	**Avg. # New C. C.**
Brain (White Matter)	2538	418	1833.5	249.9
Brain (Grey Matter)	9688	1316	6834.2	2115.9
Brain (CSF)	8574	676	5303.3	787.1
Lung (Inspiration)	1908	128	1213.7	288.4
Lung (Expiration)	3284	237	2068.7	483.2
Thorax	1962	84	1123.7	143.8

7. CONCLUSION

We presented a new repairing algorithm for making 3D binary digital images well-composed. Our algorithm is randomized and "repairs" the input image by removing points from the background and inserting them into the foreground. Each point removed from the background and inserted into the foreground eliminates at least one critical configuration of the image, but it may also give rise to one or more critical configurations. However, our algorithm cannot create new critical configurations indefinitely, as we showed that it always terminates and produces a well-composed image. We also showed that the expected number of new critical configurations is bounded above by a fraction of the number of critical configurations in the input image. Furthermore, experiments with medical imaging data showed that the number of new critical configurations generated by our algorithm is far below the given theoretical bound.

We plan to incorporate some improvements to the random procedure used by our algorithm to select points from the background to eliminate critical configurations. Recall that our algorithm chooses points from a very small subset of the background based on whether or not these points create at least one new critical configuration. It would be interesting to evaluate the effect of consecutive choices of points by "looking ahead" the number of new critical configurations created by the algorithm. Then, we could choose a "chain" of insertions that lead to less new critical configurations. Certainly, the number of such chains is exponential in the number of points of the image. So, we must evaluate the trade off between runtime and the size of P. We are also investigating the existence of a deterministic, polynomial-time algorithm that generates a well-composed image by removing a smallest possible subset of background points. Such an algorithm, if it exists, would be very useful from both theoretical and practical point of views.

REFERENCES

1. L. Latecki, "3d well-composed pictures," *Graphical Models and Image Processing* **59**(3), pp. 164–172, 1997.
2. L. Latecki, U. Eckhardt, and A. Rosenfeld, "Well-composed sets," *Computer Vision and Image Understanding* **61**(1), pp. 70–83, 1995.
3. J. Marchadier, D. Arques, and S. Michelin, "Thinning grayscale well-composed images," *Pattern Recognition Letters* **25**(5), pp. 581–590, 2004.
4. A. Rosenfeld, T. Kong, and A. Nakamura, "Topology-preserving deformations of two-valued digital pictures," *Graphical Models and Image Processing* **60**(1), pp. 24–34, 1998.
5. M. Siqueira, L. Latecki, and J. Gallier, "Making 3d binary digital images well-composed," Tech. Rep. MS-CIS-04-22, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA, 2004.
6. W. Lorensen and H. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, **21**, pp. 163–169, 1987.
7. R. Shekhar, E. Fayyad, R. Yagel, and J. Cornhill, "Octree-based decimation of marching cubes surfaces," in *Proceedings of the IEEE Visualization '96*, pp. 335–342, 1996.
8. T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Transactions on Graphics* **21**(3), pp. 339–346, 2002.
9. A. Gross and L. Latecki, "Digitizations preserving topological and differential geometric properties," *Computer Vision and Image Understanding* **62**(3), pp. 370–381, 1995.
10. L. Latecki, *Discrete Representation of Spatial Objects in Computer Vision*, Kluwer Academic Publishers, 1998.
11. T. Y. Kong, "Topology-preserving deletion of 1's from 2-, 3- and 4-dimensional binary images," in *Proceedings of the DCGI'97*, *Lecture Notes in Computer Science* **1347**, pp. 3–18, Springer-Verlag, 1997.
12. G. Herman, *Geometry of Digital Spaces*, Springer-Verlag, 1998.
13. E. Bloch, *A First Course in Geometric Topology and Differential Geometry*, Birkhäuser, 1997.
14. R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
15. D. Collins, A. Zijdenbos, V. Kollokian, J. Sled, N. Kabani, C. Holmes, and A. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Transactions on Medical Imaging* **17**(3), pp. 463–468, 1998.
16. G. Herman and B. Carvalho, "Multiseeded segmentation using fuzzy connectedness," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(5), pp. 460–474, 2001.